Take-Home Exam

Dustin Leatherman

November 3, 2020

Background

The data used in this analysis are simulated measurements of the Normalized Difference Vegation Index (NVDI). This is commonly used indicate the "greeness" of a given area. Specifically, it is most often measured from satellites to monitor desertification. This fake dataset is generated from three fake satellites, each with their own pros and cons.

$Y_1 : 365 \times 6$ matrix of unbiased, but noisy measurements θ_{tj} where $j = 1, ..., 6$ and $t = 1, ..., 365$. Y_1 is missing 80% of the time. The data can be considered Missing Completely At Random.

Y_2: A potentially biased (and noisy) measurement. This satellite only provides a measurement of θ_{1t}.

Y_3: A potentially biased (and noisy) measurement. It estimates over the entire spatial domain of θ. i.e. $\frac{1}{6}\sum_{j=1}^{6}\theta_{tj}$

The data has been generated from the following models:

$$\theta_1 \sim N(\mu_1, \Sigma_1),\ \theta_t|\theta_{t-1} \sim N(\mu_2 + \rho\theta_{t-1}, \Sigma_2)$$

where μ_1, μ_2 are mean vectors, $\rho \in (0, 1)$ which controls temporal dependence, and Σ_1, Σ_2 are 6×6 covariance matrices.

Statistical Model

Since the data were derived from a Normal distribution, it is appropriate to use a Normal distribution for it's likelihood functions.

$Y_1 : Y_1 \sim N(\theta_{tj}, \tau_{Y_1})$

The expected value of Y_1 should be the unbiased value in the Y1 dataset with some variance. The Y_1 likelihood is the most straightforward among the three likelihoods.

$\tau_{Y_1} \sim InvWishart(k = 7, R)$. K and R are hyper-parameters for the Wishart Distribution. They were chosen to be k = 7 and R a 5 x 5 a diagonal matrix with values of 0.1. The assumption for this prior is that a measurement for a given pixel is independent of other pixels. A small variance was chosen in order to make this uninformative.

θ_{t1} is a Normal Prior with mean $\mu_1 \sim N(0, 1000)$ and precision $\tau \sim InvGamma(0.1, 0.1)$. This is derived from the initial distribution of θ_{tj}. Since there is correlation between the NVDI measurements for a given day and its surrounding days, that needs to be taken into account. Thus $\theta_{t,2-6} \sim N(\mu_2 + \rho\theta_{t-1,j})$ where $\rho \in (0, 1)$. This can be represented with the uninformative Beta Prior, $\rho \sim Beta(1, 1)$

$Y_2 : Y_2 \sim N(\theta_{t1} + \gamma_{Y_2}, \tau_{Y_2})$

Y_2 is a biased estimate of θ_1 with some unknown variance. This is similar to Y_1 but with the addition of an unknown bias γ_{Y_2}, which is represented as an uninformative Normal Prior. τ_{Y_2} is an uninformative Inverse Gamma Prior $\tau_{Y_2} \sim InvGamma(0.1, 0.1)$

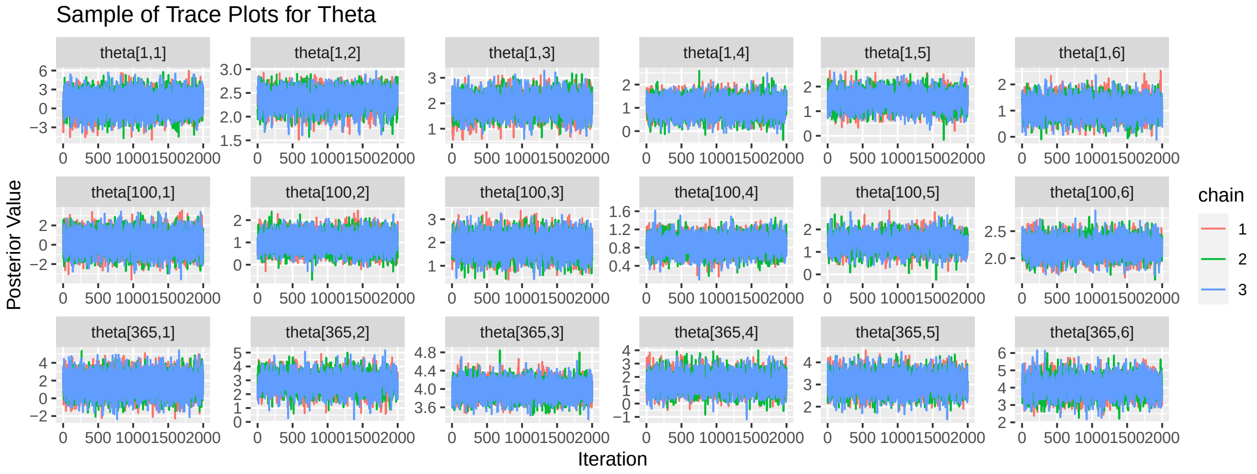
$Y_3 : Y_3 \sim N(\gamma_{Y_3} + \frac{1}{6}\sum_{j=1}^{6} \theta_{tj}, \tau_{Y_3})$

Y_3 is represented as an average across of all NVDI measurements plus some unknown bias γ_{Y_3}. Priors γ_{Y_3} and τ_{Y_3} are modeled similarly to the other datasets.

JAGS

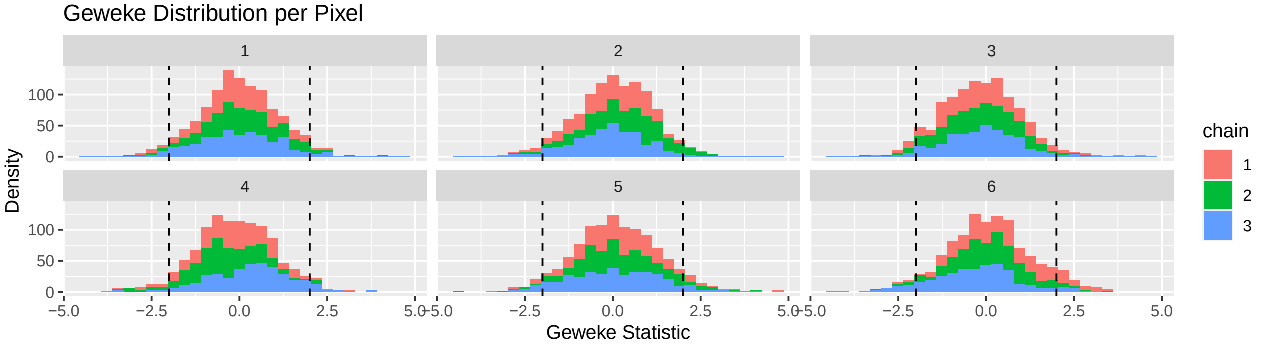
```r
model_string <- textConnection("model{
  # Likelihood
  for(i in 2:n){
    for(j in 1:6){
      Y1[i, j] ~ dnorm(theta[i, j], tauY1)
    }
    Y2[i] ~ dnorm(theta[i, 1] + gammaTheta2, tauY2)
    Y3[i] ~ dnorm(((theta[i, 1] + theta[i, 2] + theta[i, 3] + theta[i, 4] + theta[i, 5] + theta[i, 6])/6
    theta[i, 1] ~ dnorm(mu[1], tauTheta1)
    theta[i, 2:6] ~ dmnorm(mu[2:6] + rho * theta[i - 1, 2:6], sigmaTheta2)
  }


  # Set first row separately since there is serial-correlation in the model.
  for(j in 1:6){
    Y1[1, j] ~ dnorm(theta[1, j], tauY1)
  }
  Y2[1] ~ dnorm(theta[1, 1] + gammaTheta2, tauY2)
  Y3[1] ~ dnorm(((theta[1, 1] + theta[1, 2] + theta[1, 3] + theta[1, 4] + theta[1, 5] + theta[1, 6])/6)

  theta[1, 1] ~ dnorm(mu[1], tauTheta1)
  theta[1, 2:6] ~ dmnorm(mu[2:6], sigmaTheta2)

  # Priors
  for(z in 1:6) {
    mu[z] ~ dnorm(0, 0.001)
  }
  tauY3 ~ dgamma(0.1, 0.1)
  tauY2 ~ dgamma(0.1, 0.1)
  tauY1 ~ dgamma(0.1, 0.1)
  tauTheta1 ~ dgamma(0.1, 0.1)
  gammaTheta2 ~ dnorm(0,0.001)
  gammaTheta3 ~ dnorm(0,0.001)
  rho ~ dbeta(1,1)
  # k = 7 makes the expected value of sigma equal to R
  sigmaTheta2 ~ dwish(R[,], 7)
}")

model <- jags.model(model_string,data=data,n.chains=nChains, quiet = TRUE)
update(model,burn=nBurn,progress.bar="none")
model1.out <- coda.samples(model,variable.names=params,thin=nThin,n.iter=nIter, progress.bar = "none")
unioned.data <- unionJagsOutput(model1.out)
```

Convergence Diagnostics



Simulation of Days 1, 10, 100, and 365.

Pictured are Trace plots for a sample of days between 1 and 365. A sample of days are displayed due to the sheer number of parameters. The three chains appear to overlap decently well for the variables of interest which suggest that the parameters for the models have converged.



Histograms highlighting the convergence of the chains using the Geweke Statistic. It is ideal to be between the dotted lines.

Most of the Geweke Statistics fall under abs(2) though all thetas have some values that fall outside. This indicates that some runs of the parameters have not converged.

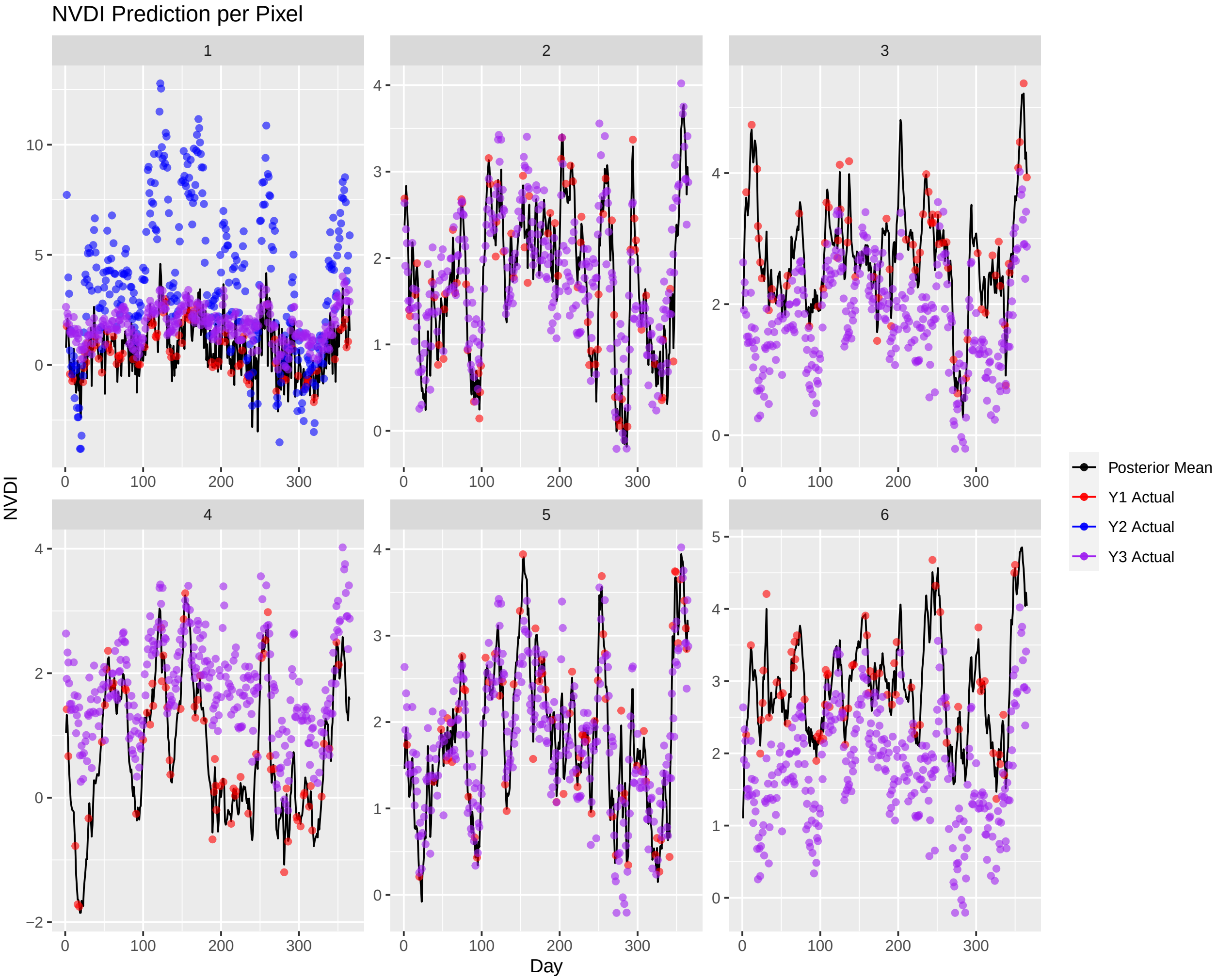
Table 1: Gelman-Rubin Statistic Quantiles to measure convergence of chains

quantile	Point est.	Upper C.I.
min	0.9995566	0.999585
0.25	1.0001667	1.000996
0.5	1.0006713	1.002686
0.75	1.0015320	1.005729
max	1.0097701	1.030181

The Gelman Statistics are close to 1 indicating that convergence has been reached.

Despite the magnitude in some of the Geweke statistics, it appears that convergence for this model has been attained based on the outcome of the Trace Plots, Gelman-Rubin Statistics, and Geweke Statistics..

Final Results



The Posterior mean accurately tracks the golden-image data (red) when available for all pixels. The Y3 Actuals (average-valued measurements) compares well to the posterior mean for Pixel 1, 2, and 5 but not so well for the others. The posterior mean does not follow the Y2 Actuals for pixel 1. Rather it seems closer to Pixel 2 or 4 in terms of shape. This model seems to accurately track the golden-image datapoints fairly closely and is bolstered in certain cases by the other datasets. Overall, I would consider this model to be fairly decent in predicting NVDI.