Project 4: A Bayesian model for hurricane trajectories.

Jack Yan, Jiayi Shen, Siquan Wang, Jin Ge

5/12/2020

Introduction

Predicting hurricane trajectories is an important task in meteorology. It is helpful to model the wind speed of a hurricane in a few hours, provided the current wind speed, latitude, longitude and other parameters of the hurricane. In this project, we aimed to estimate the distribution of parameters in a wind speed prediction model by bayesian approach. Starting with a pre-specified prior distribution, conditional posterior distributions of the parameters were derived. Gibbs sampler was then used to sample from the posterior distributions. This enabled us to estimate the posterior means and 95% credibility intervals of the parameters.

Hurrican Data

hurrican356.csv collected the track data of 356 hurricanes in the North Atlantic area since 1989. For all the storms, their location (longitude & latitude) and maximum wind speed were recorded every 6 hours. The data includes the following variables

1. **ID**: ID of the hurricans
2. **Season**: In which **year** the hurricane occurred
3. **Month**: In which **month** the hurricane occurred
4. **Nature**: Nature of the hurricane

- ET: Extra Tropical
- DS: Disturbance
- NR: Not Rated
- SS: Sub Tropical
- TS: Tropical Storm

5. **time**: dates and time of the record

6. **Latitude** and **Longitude**: The location of a hurricane check point
7. **Wind.kt** Maximum wind speed (in Knot) at each check point

In this project we are interested in studying 356 hurricanes in the North Atlantic area since 1989, especially about how they are formed and what predictors will be of high prediction power of the hurricane properties. We have several measuremnt on each hurrianc such as the time when it was formed, the nature of the hurricane , the location of a hurricane check point and the maximum wind speed (in Knot) at each check point. To increase the flexibility of our model, we planned to use Bayesian approaching for modeling the outcome of interest, which is the wind speed. By assuming relatively simple structure of priors, we could use Gibbs sampling in our model computation and the result will be pretty robust.

Method

Let t be time (in hours) since a hurricane began, and for each hurrican i, we denote $Y_i(t)$ to be the wind speed at time t. The following Baysian model was suggested.

$$Y_{i,j}(t+6) = \mu_{i,j}(t) + \rho_j Y_{i,j}(t) + \epsilon_{i,j}(t)$$

where $\mu_{i,j}(t)$ is the funtional mean, and the errors $(\epsilon_{i,1}(t), \epsilon_{i,2}(t), \epsilon_{i,3}(t))$ follows a multivariate normal distributions with mean zero and covariance matrix Σ, independent across t. We further assume that the mean functions $\mu_{i,j}(t)$ can be written as

$$\mu_{i,j}(t) = \beta_{0,j} + x_{i,1}(t)\beta_{1,j} + x_{i,2}\beta_{2,j} + x_{i,3}\beta_{3,j} + \sum_{k=1}^{3} \beta_{3+k,j}\Delta_{i,k}(t-6)$$

where $x_{i,1}(t)$, ranging from 0 to 365, is the day of year at time t, $x_{i,2}$ is the calenda year of the hurrican, and $x_{i,3}$ is the type of hurrican, and

$$\Delta_{i,k}(t-6) = Y_{i,k}(t) - Y_{i,k}(t-6), k = 1, 2, 3$$

are the change of latitude, longitude, and wind speed between $t-6$ and t.

Prior distributions

We assume the following prior distributions

For $\boldsymbol{\beta} = (\beta_{k,j})_{k=0,...,6,j=1,2,3}$, we assume $\pi(\boldsymbol{\beta})$ is jointly normal with mean 0 and variance $diag(1, p)$.

We assume that $\pi(\rho_j)$ follows a trucated normal $N_{[0,1]}(0.5, 1/5)$

$\pi(\sigma^2)$ follows a $Wishart(3, diag(0.1, 3))$

Likelihood

The log-likelihood of $Y(t+6)$ is

$$l(Y(t+6)|\mathbf{X}, \boldsymbol{\beta}, \rho, Y(t)) = -n \log\left(\sigma\sqrt{2\pi}\right) - \sum_{i=1}^{n} \frac{1}{2\sigma^2}\left(Y_i(t+6) - \mathbf{X}^T\boldsymbol{\beta} - \rho Y_i(t)\right)^2$$

Posterior of β

Since each β_k is mutually-independent distributed, we can look at their posterior distribution individually. Note that $k = 0, 1, 2, ..., 10$ because there are five categories for x_{i3}. \

The prior of β_k has the log-likelihood function:

$$\log \pi(\beta_k) = -\log(\sqrt{2\pi}) - \frac{1}{2}\beta_k^2$$

The posterior of β_k has the log-likelihood function:

$$\begin{aligned}
\log \pi(\beta_k|Y(t+6), \mathbf{X}, \boldsymbol{\beta}_{-k}, \rho, Y(t)) &= \log \pi(\beta_k) + l(Y(t+6)|\mathbf{X}, \boldsymbol{\beta}, \rho, Y(t)) \\
&\propto \text{const} - \frac{1}{2}\beta_k^2 - \sum_{i=1}^{n} \frac{1}{2\sigma^2}\left[\beta_k^2 x_{ik}^2 - 2\beta_k x_{ik}\left(Y_i(t+6) - \mathbf{X}_{-k}^T\boldsymbol{\beta}_{-k} - \rho Y_i(t)\right)\right] \\
&= \text{const} - \left[\beta_k^2\Big\{\sum_{i=1}^{n} \frac{1}{2\sigma^2}(x_{ik}^2 + \frac{\sigma^2}{n})\Big\} - 2\beta_k \frac{1}{2\sigma^2}\Big\{\sum_{i=1}^{n} x_{ik}\left[Y_i(t+6) - \mathbf{X}_{-k}^T\boldsymbol{\beta}_{-k} - \rho Y_i(t)\right]\Big\}\right] \\
&= \text{const} - \frac{1}{2}\Big\{\sum_{i=1}^{n} \frac{1}{\sigma^2}(x_{ik}^2 + \frac{\sigma^2}{n})\Big\}\Big\{\beta_k - \frac{\sum x_{ik}[Y_i(t+6) - \mathbf{X}_{-k}^T\boldsymbol{\beta}_{-k} - \rho Y_i(t)]}{\sum(x_{ik}^2 + \frac{\sigma^2}{n})}\Big\}^2
\end{aligned}$$

Thus, the posterior of β_k follows a normal distribution with

$$\begin{aligned}
\mu_k &= \frac{\sum x_{ik}[Y_i(t+6) - \mathbf{X}_{-k}^T\boldsymbol{\beta}_{-k} - \rho Y_i(t)]}{\sum(x_{ik}^2 + \frac{\sigma^2}{n})} \\
\sigma_k^2 &= \Big\{\sum_{i=1}^{n} \frac{1}{\sigma^2}(x_{ik}^2 + \frac{\sigma^2}{n})\Big\}^{-1}
\end{aligned}$$

Posterior of ρ

The prior of ρ has the log-likelihood function:

$$\log \pi(\rho) = -\log(\sqrt{\frac{2\pi}{5}}) - \frac{25}{2}(\rho - \frac{1}{2})^2$$

The posterior of ρ is proportional to

$$\begin{aligned}
&\text{const} - \frac{25}{2}(\rho - \frac{1}{2})^2 - \sum_{i=1}^{n} \frac{1}{2\sigma^2}\Big(Y_i(t+6) - \mathbf{X}^T\boldsymbol{\beta} - \rho Y_i(t)\Big)^2 \\
&= \text{const} - \frac{n}{2\sigma^2}\Big(\frac{25\sigma^2}{n}\rho^2 - \frac{25\sigma^2}{4n}\rho\Big) - \sum_{i=1}^{n} \frac{1}{2\sigma^2}\Big(\rho^2 Y_i(t)^2 - 2\rho Y_i(t)[Y_i(t+6) - \mathbf{X}^T\boldsymbol{\beta}]\Big) \\
&= \text{const} - (\frac{25}{2} + \frac{1}{2\sigma^2}\sum Y_i(t)^2)\rho^2 + \frac{1}{\sigma^2}\rho \sum \Big\{\Big(Y_i(t)[Y_i(t+6) - \mathbf{X}^T\boldsymbol{\beta}]\Big) + \frac{25\sigma^2}{8n}\Big\} \\
&= \text{const} - \frac{1}{2}\Big\{25 + \frac{1}{\sigma^2}\sum Y_i(t)^2\Big\}\Big\{\rho - \frac{\frac{1}{\sigma^2}\sum\Big\{\Big(Y_i(t)[Y_i(t+6) - \mathbf{X}^T\boldsymbol{\beta}]\Big) + \frac{25\sigma^2}{8n}\Big\}}{25 + \frac{1}{\sigma^2}\sum Y_i(t)^2}\Big\}^2
\end{aligned}$$

Thus, the posterior of ρ follows a normal distribution with

$$\begin{aligned}
\mu_\rho &= \frac{\frac{1}{\sigma^2}\sum\Big\{\Big(Y_i(t)[Y_i(t+6) - \mathbf{X}^T\boldsymbol{\beta}]\Big) + \frac{25\sigma^2}{8n}\Big\}}{25 + \frac{1}{\sigma^2}\sum Y_i(t)^2} \\
&= \frac{\sum\Big(Y_i(t)[Y_i(t+6) - \mathbf{X}^T\boldsymbol{\beta}]\Big) + \frac{25\sigma^2}{8}}{25\sigma^2 + \sum Y_i(t)^2} \\
\sigma_\rho^2 &= \Big\{25 + \frac{1}{\sigma^2}\sum Y_i(t)^2\Big\}^{-1}
\end{aligned}$$

Posterior of σ^2

The prior of σ^2 has the log-likelihood function:

$$\begin{aligned}
\log \pi(\sigma^2) &= \text{const} - (\alpha + 1)\log\frac{1}{\sigma^2} + \frac{-\alpha'}{\sigma^2} \\
&= \text{const} - 2(\alpha + 1)\log(\sigma) - \alpha'\sigma^{-2}
\end{aligned}$$

The posterior of σ^2 is proportional to

$$\begin{aligned}
&\text{const} - 2(\alpha+1)\log(\sigma) - \alpha'\sigma^{-2} - n\log\Big(\sigma\sqrt{2\pi}\Big) - \sum_{i=1}^{n}\frac{1}{2\sigma^2}\Big(Y_i(t+6) - \mathbf{X}^T\boldsymbol{\beta} - \rho Y_i(t)\Big)^2 \\
&= \text{const} - \Big(n + 2(\alpha+1)\Big)\log(\sigma) - \sigma^{-2}\Big\{\alpha' + \sum_{i=1}^{n}\frac{1}{2}\Big(Y_i(t+6) - \mathbf{X}^T\boldsymbol{\beta} - \rho Y_i(t)\Big)^2\Big\}
\end{aligned}$$

where $\alpha = \alpha' = 0.001$.

Thus, the posterior of ρ follows an inverse-gamma distribution with

$$\begin{aligned}
\alpha_{post} &= n + 2\alpha + 1 \\
\alpha'_{post} &= \alpha' + \sum_{i=1}^{n}\frac{1}{2}\Big(Y_i(t+6) - \mathbf{X}^T\boldsymbol{\beta} - \rho Y_i(t)\Big)^2
\end{aligned}$$

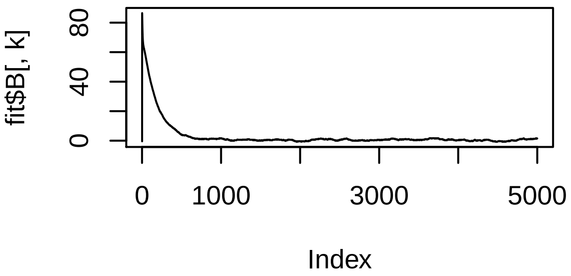
Gibbs sampling algorithm

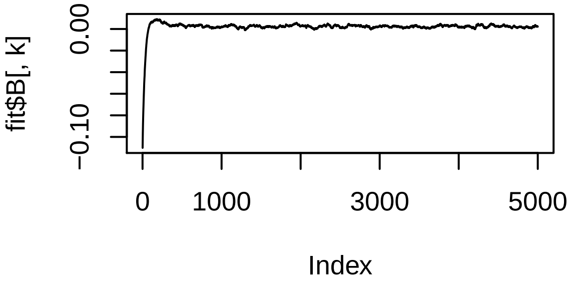
Denote $\theta = (\beta_0, \beta_1, ..., \beta_9, \rho, \sigma^2)$. We proceed as follows:
1. Begin with some initial values of θ^0.
2. Sample each component of the vector, θ, from the distribution of that component conditioned on all other components sampled so far. For example, for $k \geq 1$, Generate $\beta_0^{(k)}$ from $\pi(\beta_0|\beta_1^{(k-1)}, ..., \beta_9^{(k-1)}, \rho^{(k-1)}, \sigma^{2(k-1)}, Y, \mathbf{X})$. Then generate $\beta_1^{(k)}$ from $\pi(\beta_1|\beta_0^{(k)}, \beta_2^{(k-1)}, ..., \beta_9^{(k-1)}, \rho^{(k-1)}, \sigma^{2(k-1)}, Y, \mathbf{X})$.
3. Repeat the above step k times.

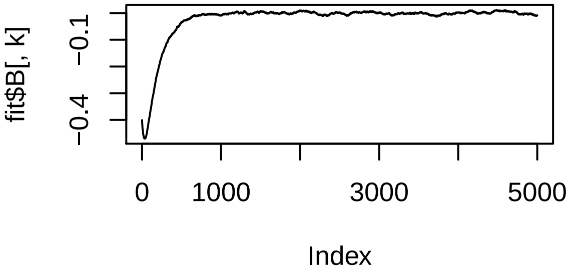
We will randomly select 80% hurricanes and applied the proposed Gibbs sampling algorithm to estiamte the posterior distributions of the model parameters. Then we will apply the model to track the remaining 20% hurricans, and evaluate model performance in terms of how well could predict and track these hurricanes.

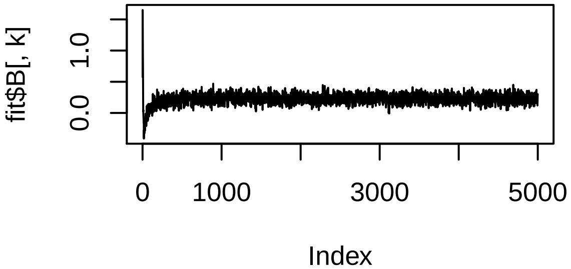
Results

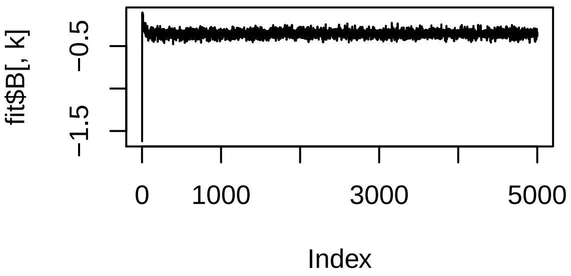
Parameter estimation of posterior distributions

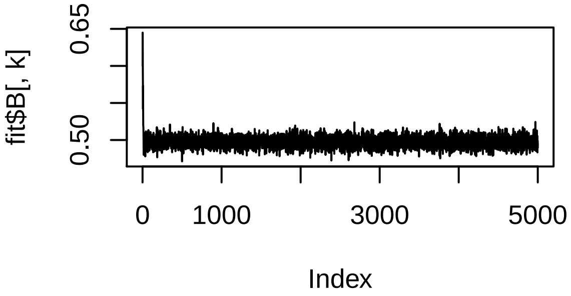


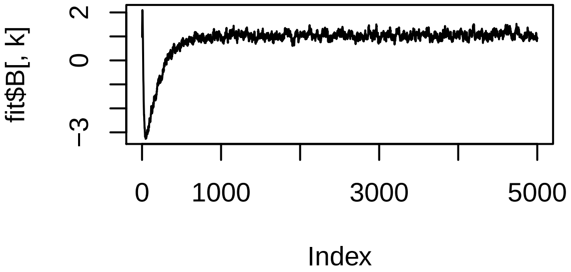


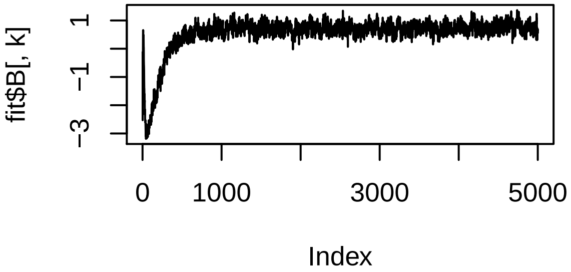


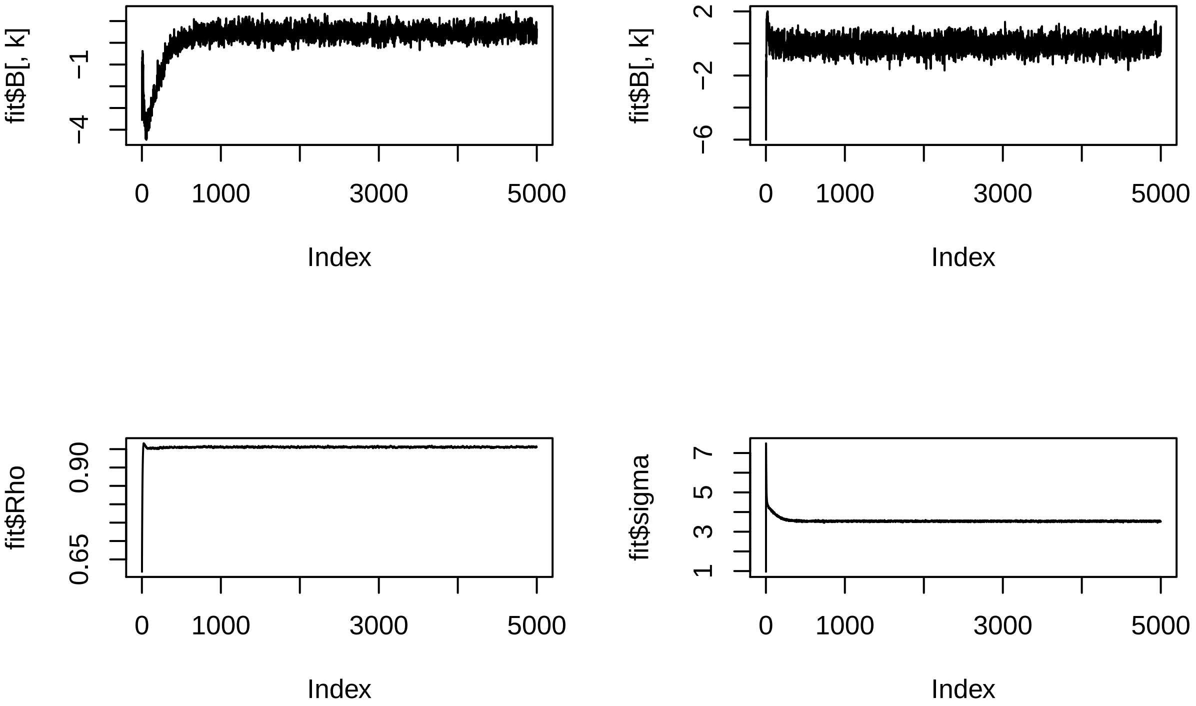




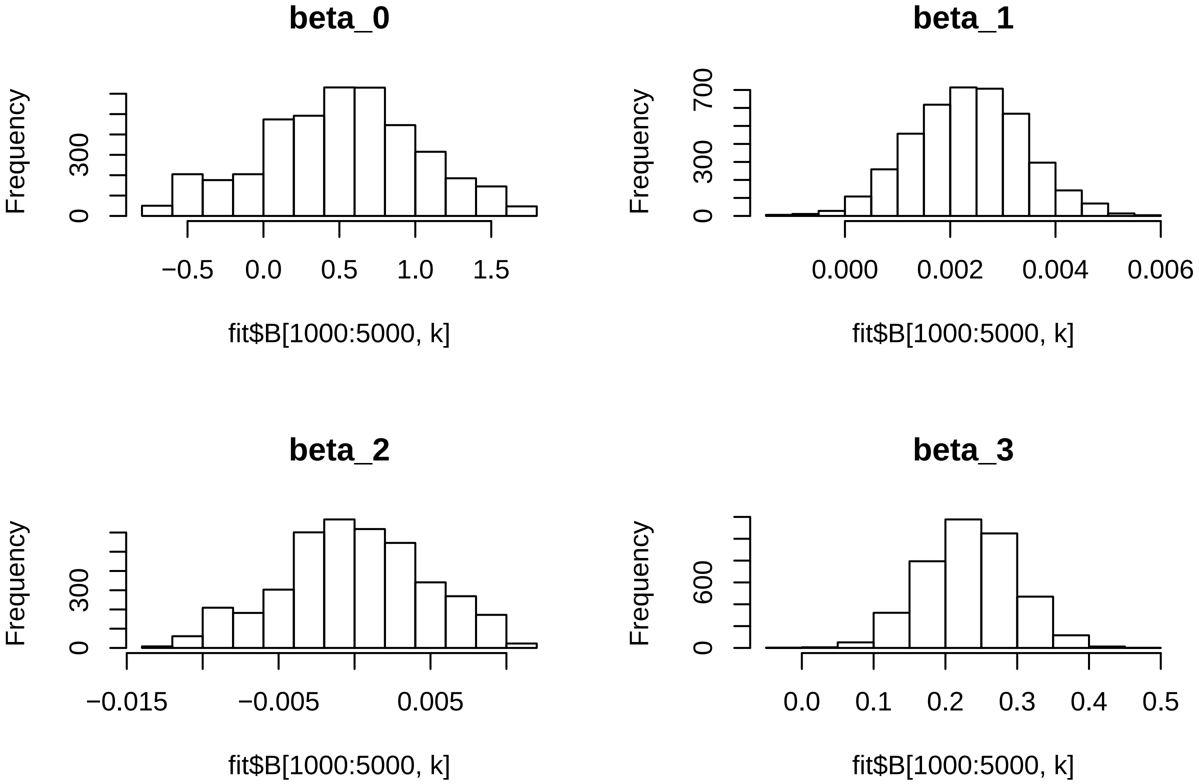


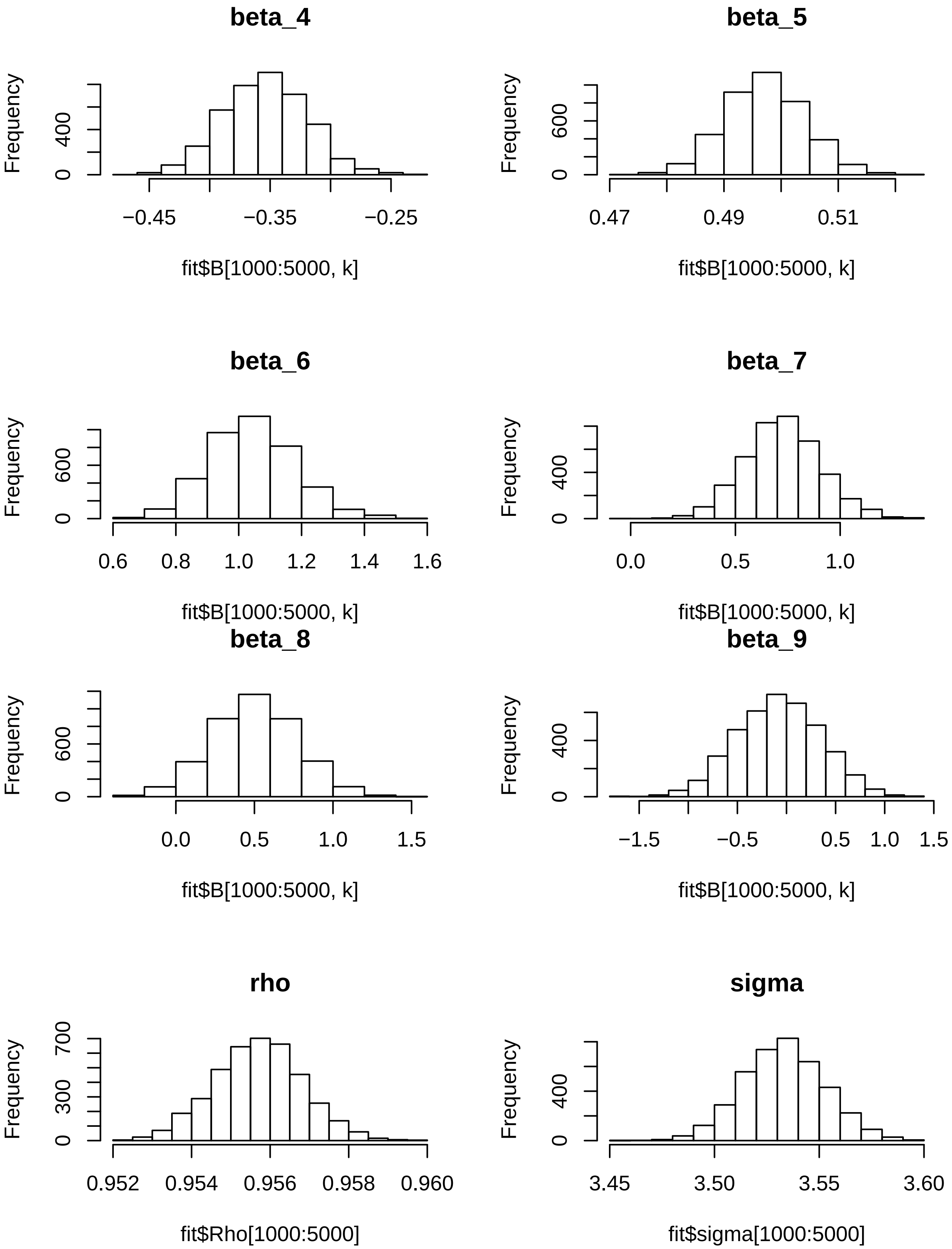






The plots above shows the length of burn-in period and stationary stage of each Markov chains. On average, burn-in period is about 500 runs.





The histograms above show the distribution of parameter values after each chain enters stationary stage.

parameter	posterior.mean	CrI.low	CrI.high
(Intercept)	0.6485750	-0.5045427	36.3870108

parameter	posterior.mean	CrI.low	CrI.high
Yday	0.0023540	0.0000803	0.0063911
Year	-0.0013267	-0.3295719	0.0085877
DeltaLatitude	0.2346255	0.0747606	0.3566071
DeltaLongitude	-0.3539148	-0.4224030	-0.2818217
DeltaSpeed	0.4972567	0.4839647	0.5110733
NatureTS	1.0301557	-1.7909435	1.3176497
NatureET	0.7123291	-1.8519958	1.0847332
NatureSS	0.4867485	-2.5668894	1.0083927
NatureNR	-0.0839264	-0.9149674	0.7809101
Y(t)	0.9551738	0.9524984	0.9579976
sigma	3.5562687	3.4960875	3.8657191

The table above shows the estimated posterior mean of each parameter in the Bayesian model, with the associated 95% credibility intervals (CrI). According to this model, it seems that `DeltaSpeed`(change in wind speed) and `Y(t)` (the wind speed at current time point) are highly predictive of the wind speed at the next time point. More specifically, they are both positively associated with `Y(t+6)`, the wind speed after 6 hours. `Yday` (the day of a year at current time point) and `DeltaLatitude` (the change in latitude) also show significant association with the wind speed after 6 hours.

Prediction

We used the remaining 20% hurricans data to conduct predictions using our proposed model. The mean square error (MSE) is 26.8380143.

Discussion

Our Bayesian regression model and Gibbs sampling seem to work well on the hurricane dataset and the predictors we got had pretty high prediction powers. However, we could still improve our modeling precedure by considering the following extensions of this project: First, we might consider using other MCMC methods such as Hamiltonian Monte Carlo and compare its performance with Gibbs sampling. Secondly, we might use some more informative priors and see how the final outputs will change. Lastly, we might use more complicated modeling techniques such as hiearchical modeling to account to heteroscadesity among different subregions and always remember there is trade-off between model complexicity and computation efficacy/over-fitting.

Appendix

```r
library(ggplot2)
library(data.table)
library(tidyverse)
library(invgamma)

# Data Cleaning
dt = read.csv("hurrican356.csv")
dt <- as.data.table(dt)
dat.clean <- dt %>%
    mutate(time = as.character(strptime(time, format = "(%y-%m-%d %H:%M:%S)"))) %>%
    rename(Speed = Wind.kt) %>%
    arrange(ID, as.numeric(as.POSIXlt(time))) %>%
    select(c("ID", "time", "Nature", "Latitude", "Longitude", "Speed")) %>%
    mutate(Year = as.POSIXlt(time)$year) %>%
    mutate(Yday = as.POSIXlt(time)$yday) %>%
    group_by(ID) %>%
    mutate(Time = as.numeric(as.POSIXlt(time) - as.POSIXlt(first(time))) / 3600 / 6) %>%
    select(-"time") %>%
    mutate(SpeedPrev = lag(Speed)) %>%
    mutate(DeltaLatitude = Latitude - lag(Latitude)) %>%
    mutate(DeltaLongitude = Longitude - lag(Longitude)) %>%
    mutate(DeltaSpeed = Speed - lag(Speed)) %>%
    mutate(SpeedNext = lead(Speed)) %>%
    ungroup() %>%
    filter(!is.na(SpeedPrev) & !is.na(SpeedNext))

# Splitting into training (80%) and testing (20%)
ID.uniq <- unique(dat.clean$ID)
ID.training <- sample(ID.uniq, size = round(0.8 * length(ID.uniq)))
idx.training <- which(dat.clean$ID %in% ID.training)
idx.test <- which(!(dat.clean$ID %in% ID.training))

Y <- dat.clean %>% select(SpeedNext) %>% as.matrix()
Z <- dat.clean %>% select(Speed) %>% as.matrix()

Dummy <- function(dat, var) {
  x <- unlist(dat[, var], use.names = FALSE)
  dict.x = unique(x)
  dum <- outer(x, dict.x, "==") + 0
  colnames(dum) <- dict.x
  dum <- as.data.frame(dum)
  return(dum)
}

ref.Nature <- "DS"
dum.Nature <- Dummy(dat.clean, "Nature") %>% select(- ref.Nature)
colnames(dum.Nature) <- paste0("Nature", colnames(dum.Nature))

X <- dat.clean %>%
  select(c("Yday", "Year", "DeltaLatitude", "DeltaLongitude", "DeltaSpeed")) %>%
  bind_cols(dum.Nature)
X <- cbind(1, as.matrix(X))
colnames(X)[1] <- "(Intercept)"


Gibbs <- function(len.chain, Y, X, Z) {

  n <- nrow(X)
  q <- 1
  p <- 10

  mu <- 0.5
  sigma <- 1 / sqrt(5)

  invgamma_para1 = 1
  invgamma_para2 = 1
```

```
RCondPostB <- function(Rho, sigma, B) {

  for (k in 1:10){
    Res <- Y - X %*% B - Z * Rho
    mean_bk = sum(X[,k]*(Res + X[,k]*B[k])) / sum(X[,k]^2 + sigma^2/n)
    sigma_bk = sqrt(1/sum((X[,k]^2 + sigma^2/n)/sigma^2))
    B[k] = rnorm(1, mean = mean_bk, sd = sigma_bk)
  }
  return(B)
}

RCondPostRho <- function(B, sigma) {
  mean_Rho = (sum(Z*(Y -X %*% B)) + 25*sigma^2/8)/(25*sigma^2 + sum(Z^2))
  sigma_Rho = 1/(25+sum(Z^2)/sigma^2)
  Rho_new = rnorm(1, mean = mean_Rho, sd = sigma_Rho)
  while (Rho_new < 0 || Rho_new > 1){
    Rho_new = rnorm(1, mean = mean_Rho, sd = sigma_Rho)
  }
  return(Rho_new)
}

RCondPostSigma <- function(B, Rho) {
Res <- Y - X %*% B - Z * Rho
#Omega <- rWishart(1, n + d, chol2inv(chol(t(Res) %*% Res + Vinv)))[, , 1]
sigma_sq <- rinvgamma(1, n + 2*invgamma_para1 + 1, invgamma_para2 + 0.5*sum(Res^2))
while (is.infinite(sigma_sq) || sigma_sq == 0){
  sigma_sq <- rinvgamma(1, n + 2*invgamma_para1 + 1, invgamma_para2 + 0.5*sum(Res^2))
}
sigma_new = sqrt(sigma_sq)
return(sigma_new)
}

LogLik <- function(sigma, B, Rho) {
Res <- Y - X %*% B - Z * Rho
log.lik <- - n * log(sigma*sqrt(2*pi)) - (1 / 2) * (sigma^(-2))* sum(Res^2)
return(log.lik)
}

sigma <- rep(NA, len.chain)
B <- matrix(NA, nrow = len.chain, ncol = 10)
Rho <- rep(NA, len.chain)
log.lik <- rep(NA, len.chain)

sigma[1] <- sqrt(rinvgamma(1, invgamma_para1, invgamma_para2))
B[1, ] <- rnorm(p * q)
Rho[1] <- runif(1)
log.lik[1] <- LogLik(sigma[1], B[1, ], Rho[1])

for (k in 2 : len.chain) {
  print(k)
  B[k, ] <- RCondPostB(Rho[k-1], sigma[k-1], B[k-1, ])
  Rho[k] <- RCondPostRho(B[k, ], sigma[k-1])
```

```r
    sigma[k] <- RCondPostSigma(B[k, ], Rho[k])
    log.lik[k] <- LogLik(sigma[k], B[k, ], Rho[k])
  }
  return(list(sigma = sigma, B = B, Rho = Rho, log.lik = log.lik))
}

# Fitting
len.chain <- 5000
len.burnin <- 1000

ID.uniq <- unique(dat.clean$ID)
set.seed(5)
ID.training <- sample(ID.uniq, size = round(0.8 * length(ID.uniq)))
idx.training <- which(dat.clean$ID %in% ID.training)
idx.test <- which(!(dat.clean$ID %in% ID.training))

load("Gibbs.RData")

# Bayesian estimates
# B
colnames(fit$B) <- colnames(X)
B.postmean = colMeans(fit$B)
B.postmean
apply( fit$B , 2 , quantile , probs = c(0.025, 0.975) , na.rm = TRUE )

# Rho
Rho.postmean = mean(fit$Rho)
Rho.postmean
quantile(fit$Rho, probs = c(0.025, 0.975), na.rm = TRUE)

# sigma
sigma.postmean = mean(fit$sigma)
sigma.postmean
quantile(fit$sigma, probs = c(0.025, 0.975), na.rm = TRUE)

# plotting MC
par(mfrow = c(2, 2))
for (k in 1:10){
  plot(fit$B[,k], type = "l")
}
par(mfrow = c(1, 2))
plot(fit$Rho, type = "l")
plot(fit$sigma, type = "l")

# Predicting
Y.test <- Y[idx.test, ]
Y.test.pred <- X[idx.test, ] %*% B.postmean + Z[idx.test, ] * Rho.postmean
mean((Y.test.pred - Y.test)^2)
```