Forecasting the 2019 Flu Season Using Bayesian Spatial Modeling

Aaron Moose, Katie Jurewicz, and Maddy St. Ville

December 14, 2018

Abstract

This paper forecasts the 2019 monthly flu season for each state in the United States, with the exception of Alaska, Florida, and Hawaii. The monthly proportion of patients diagnosed with "influenza-like-illnesses" for each state was treated as the response and was modeled using a Bayesian spatial Conditional Autoregressive (CAR) model. Markov Chain Monte Carlo sampling techniques were utilized to generate a sample used to predict the monthly number of flu cases, and modeling techniques were verified by performing several model diagnostics procedures.

1 Introduction

Influenza - the flu - is a contagious respiratory disease that comes as a result of influenza viruses. The two common types of influenza viruses, Types A and B, are the main virus strains that cause yearly seasonal flu epidemics [1]. These flu epidemics result in up to 5 million severe cases and up to 500,000 deaths globally each year [2]. Although there are cases of the flu year round in the United States, pandemics are most common during the fall and winter. The number of cases tends to increase in October, with peaks in the winter months [1].

The ultimate goal of this paper is to forecast the monthly proportion of flu cases for each state in order to predict peak flu activity. As flu epidemics typically see a fall/winter seasonality in the United States, studies have shown that the transmission of influenza viruses is dependent on the climate [3]. Hence, it is expected that neighboring states, with similar monthly climate conditions, will have similar peak flu months; i.e., there is most likely a spatial correlation in the monthly results. Therefore, a Conditional Autoregressive (CAR) model was fit to model the spatial effects. For covariate/predictor information, the model includes monthly climate information, demographic information, and flu history for each state. All covariate information has monthly observations from 2010-2018. The model was fit using MCMC methods.

2 Methodology

2.1 The Model

After removing data for Alaska, Florida, and Hawaii, we are taking observations on 47 spatial units. To capture the relationship between the observations and areal units, we fit a CAR model given by

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{G}\boldsymbol{b} + \boldsymbol{\epsilon}.$$

Define $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m)$ such that $\boldsymbol{y}_j$ is a vector representing flu cases for the j^{th} state from October 2010 to September 2018, where y_{ijk} denotes the proportion of flu cases in the i^{th} state during the j^{th} month and k^{th} year.
Let $\boldsymbol{X}$ and $\boldsymbol{\beta}$ denote the design matrix and regression coefficients, respectively, associated with the fixed effects. Further, $\boldsymbol{G}$ denotes the design matrix associated with spatial random effects, and $\boldsymbol{b}$ is the corresponding spatial random effects vector with b_i representing the spatial random effect for the i^{th} state. Finally, we assume that $\boldsymbol{y}$ follows a normal distribution; i.e.,

$$\boldsymbol{y} \sim N(\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{G}\boldsymbol{b}, \sigma^2\boldsymbol{I}).$$

The following prior distributions were assigned:

$$\begin{aligned}
\boldsymbol{\beta} &\sim N(0, R) \\
\boldsymbol{b} &\sim \text{CAR}(\tau^2, \rho) \implies \boldsymbol{b} \sim N(0, \tau^2(D - \rho W)^{-1}) \\
\sigma^{-2} &\sim \text{Gamma}(a_\sigma, b_\sigma) \\
\tau^{-2} &\sim \text{Gamma}(a_\tau, b_\tau)
\end{aligned}$$

Here, W represents the 47 × 47 adjacency matrix, and D is a 47 × 47 diagonal matrix whose i^{th} diagonal entry is the number of neighboring states the i^{th} state has. We assume the correlation parameter, ρ, to be fixed.

The full conditional distributions were derived and are provided below where n is the length of the vector $\mathbf{y}$ and m is then number of states, 47.

$$
\begin{aligned}
p(\boldsymbol{\beta}|\sigma^2, \boldsymbol{y}, \boldsymbol{b}) &\sim N\left((\boldsymbol{X}^T\boldsymbol{X} + \sigma^2 R^{-1})^{-1}\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{G}\boldsymbol{b}), \sigma^2(\boldsymbol{X}^T\boldsymbol{X} + \sigma^2 R^{-1})^{-1}\right) \\
p(\boldsymbol{b}|\sigma^2, \tau^2, \boldsymbol{y}, \boldsymbol{\beta}) &\sim N\left(\left(\boldsymbol{G}^T\boldsymbol{G} + \sigma^2\frac{(D-\rho W)}{\tau^2}\right)^{-1}\boldsymbol{G}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}), \sigma^2\left(\boldsymbol{G}^T\boldsymbol{G} + \sigma^2\frac{(D-\rho W)}{\tau^2}\right)^{-1}\right) \\
p(\sigma^{-2}|\boldsymbol{y}, \boldsymbol{\beta}, \boldsymbol{b}) &\sim \text{Gamma}\left(a_\sigma + \frac{n}{2}, b_\sigma + \frac{1}{2}(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{G}\boldsymbol{b})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} - \boldsymbol{G}\boldsymbol{b})\right) \\
p(\tau^{-2}|\boldsymbol{b}) &\sim \text{Gamma}\left(a_\tau + \frac{m}{2}, b_\tau + \frac{1}{2}\boldsymbol{b}^T(D-\rho W)\boldsymbol{b}\right)
\end{aligned}
$$

2.2 Fitting the Model

The above model was fit via a full block Gibbs Sampling Algorithm. The algorithm ran for 50,000 iterations under the following procedure.

1. Provide initial values for the parameters, $(\beta^{(0)}, b^{(0)}, (\sigma^{-2})^{(0)}, (\tau^{-2})^{(0)})$.
2. For $g = 1, \ldots, 10000$, generate from the following sequence of full conditional distributions:
   (a) Sample $\beta^{(g)} \sim p\left(\boldsymbol{\beta}|(\sigma^{-2})^{(g-1)}, b^{(g-1)}, y\right)$
   (b) Sample $b^{(g)} \sim p\left(\boldsymbol{b}|\beta^{(g)}, (\sigma^{-2})^{(g-1)}, (\tau^{-2})^{(g-1)}, y\right)$
   (c) Sample $(\sigma^{-2})^{(g)} \sim p\left(\sigma^{-2}|\beta^{(g)}, b^{(g)}, y\right)$
   (d) Sample $(\tau^{-2})^{(g)} \sim p\left(\tau^{-2}|b^{(g)}\right)$
3. Reinitialize parameter values to $(\beta^{(g)}, b^{(g)}, (\sigma^{-2})^{(g)}, (\tau^{-2})^{(g)})$.

The following hyperparameters and initial values were assigned:

$$
\begin{aligned}
R &= 100\boldsymbol{I} \\
\boldsymbol{b} &= 0 \\
\rho &= 0.9 \\
\sigma^2 &= 2 \\
a_\sigma, b_\sigma &= 0 \\
\tau^2 &= 2 \\
a_\tau, b_\tau &= 0.
\end{aligned}
$$

2.3 The Data

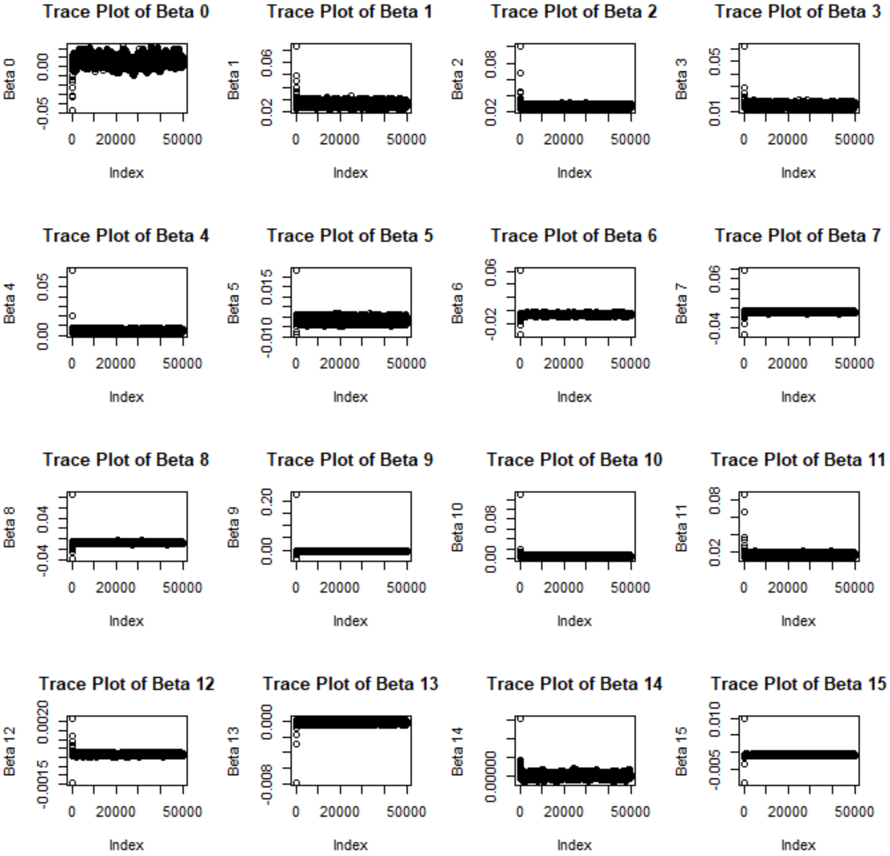
We utilized previous flu history from 2010-2018 to fit the model. There are more than 2900 healthcare providers nationally that provide data to the Centers for Disease Control and Prevention (CDC) of the total number of patients seen as well as the total number of patients diagnosed with an ILI each week. With this, the CDC organizes and releases ILINet surveillance data to the public each week [4]. ILI refers to influenza-like illness and is defined to be a temperature of 100 degrees Fahrenheit or greater as well as a cough or sore throat with no other known cause except for influenza [4]. The total number of ILI patients divided by the total number of patients provides an estimate of the total population that experience an ILI. This value is not adjusted to the population of the state and is therefore an unweighted ILI estimate [5]. The ILINet also provides a weighted ILI percentage which is weighted on the basis of state population [5]. The weeks stated in the ILINet dataset are based on the surveillance weeks as defined by the morbidity and mortality weekly reports [6].

The unweighted ILI data was used as the response variable for the current analysis. The weighted ILI were unavailable, thus unweighted were used. The ILI data was originally in a weekly format; thus, we transformed the data based on the morbidity and mortality weekly reports calendar. Depending on if January 1st falls in the beginning or end of the week, ”week 1” data could fall into either January or December, allowing for there to be a maximum of 53 weeks in a year [7].

The full model included the following predictor variables: temperature, precipitation, population density, and unemployment as well as an intercept and indicator variables for each month, with October as the baseline. Temperature and precipitation data points refer to the averages for each month for every state [8]. Temperature was measured in degrees Fahrenheit and precipitation was measured in inches. Population density was calculated based off of population and land area metrics for each state [9]. The United States Census Bureau [9] provided population estimates by year and land area in square miles for each state; thus, population density was calculated as the number of people in the state per square mile. Lastly, unemployment data refers to the unemployment rate for each month in every state [10]. Data from years 2010-2018 framed the current analysis. All data were on the state and monthly levels. Florida, DC, Alaska, and Hawaii were removed from all data sets and, when applicable, New York City data was combined with New York State data.

2.4 Model Diagnostics

Trace plots and effective sample size were utilized in performing model diagnostics. The effective sample size ranged from 139 to 46,842. Further, the following figure provides the trace plots for the components of $\boldsymbol{\beta}$, of which we can see that each component converged.



Trace Plot of Beta 0; Trace Plot of Beta 1; Trace Plot of Beta 2; Trace Plot of Beta 3; Trace Plot of Beta 4; Trace Plot of Beta 5; Trace Plot of Beta 6; Trace Plot of Beta 7; Trace Plot of Beta 8; Trace Plot of Beta 9; Trace Plot of Beta 10; Trace Plot of Beta 11; Trace Plot of Beta 12; Trace Plot of Beta 13; Trace Plot of Beta 14; Trace Plot of Beta 15

The expected mean of all the β's are

β_0	2.821×10^{-3}	β_1	2.617×10^{-2}
β_2	2.637×10^{-2}	β_3	1.467×10^{-2}
β_4	4.373×10^{-3}	β_5	-1.545×10^{-2}
β_6	-5.548×10^{-3}	β_7	-7.784×10^{-3}
β_8	-7.299×10^{-3}	β_9	-3.981×10^{-1}
β_{10}	5.055×10^{-3}	β_{11}	1.617×10^{-2}
β_{12}	1.5333×10^{-4}	β_{13}	-1.324×10^{-4}
β_{14}	6.996×10^{-10}	β_{15}	-7.493×10^{-4}

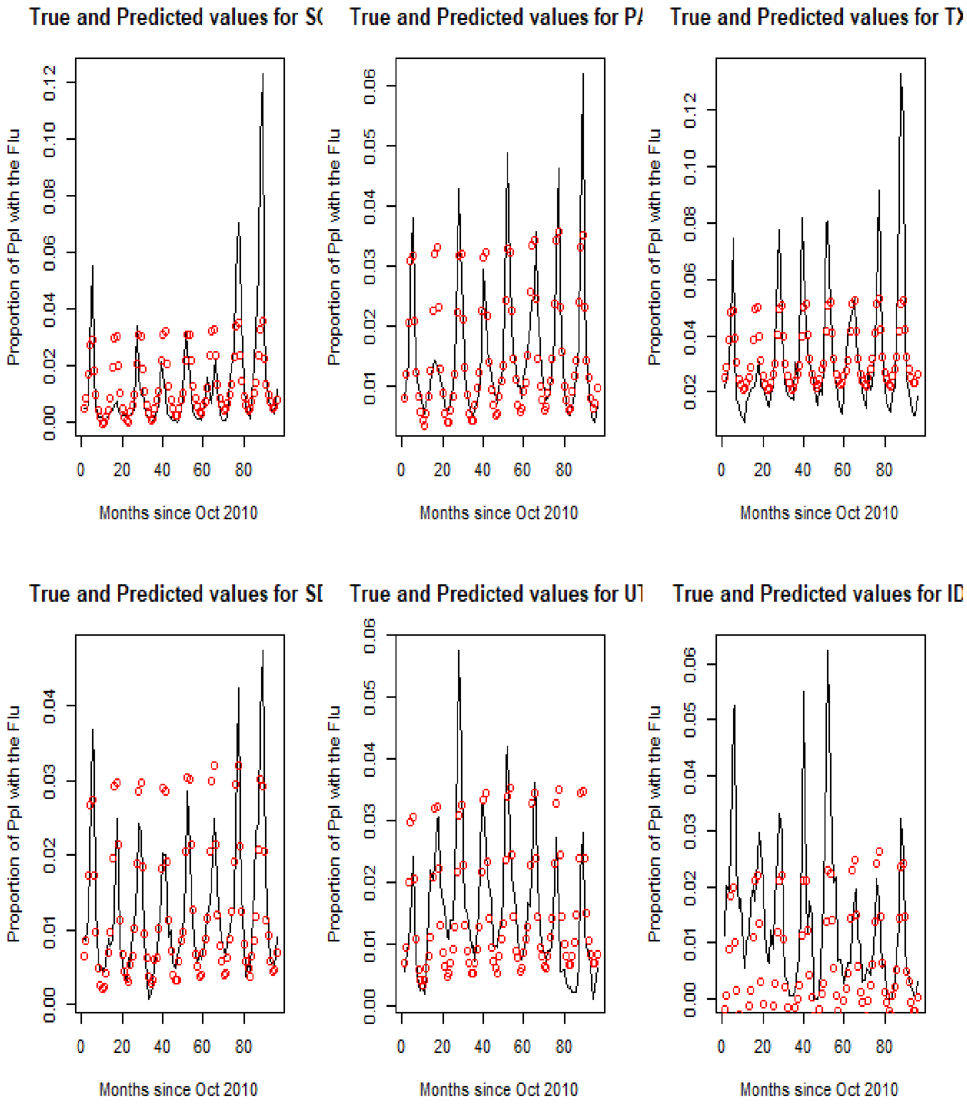
where β_0 to β_{11} are the coefficients for the months.

In addition to this, a 95% HPD interval was created to determine whether each β_i was significant.

β_0	$(-0.024, 0.014)$	β_1	$(0.024, 0.029)$
β_2	$(0.024, 0.0285)$	β_3	$(0.013, 0.016)$
β_4	$(0.003, 0.006)$	β_5	$(-0.003, -0.00009)$
β_6	$(-0.007, -0.004)$	β_7	$(-0.01, -0.006)$
β_8	$(-0.009, -0.005)$	β_9	$(-0.006, -0.002)$
β_{10}	$(0.003, 0.007)$	β_{11}	$(0.001, 0.002)$
β_{12}	$(0.00007, 0.0002)$	β_{13}	$(-0.0003, 0.00005)$
β_{14}	$-0.00001, 0.00001)$	β_{15}	$-0.0009, -0.0006)$

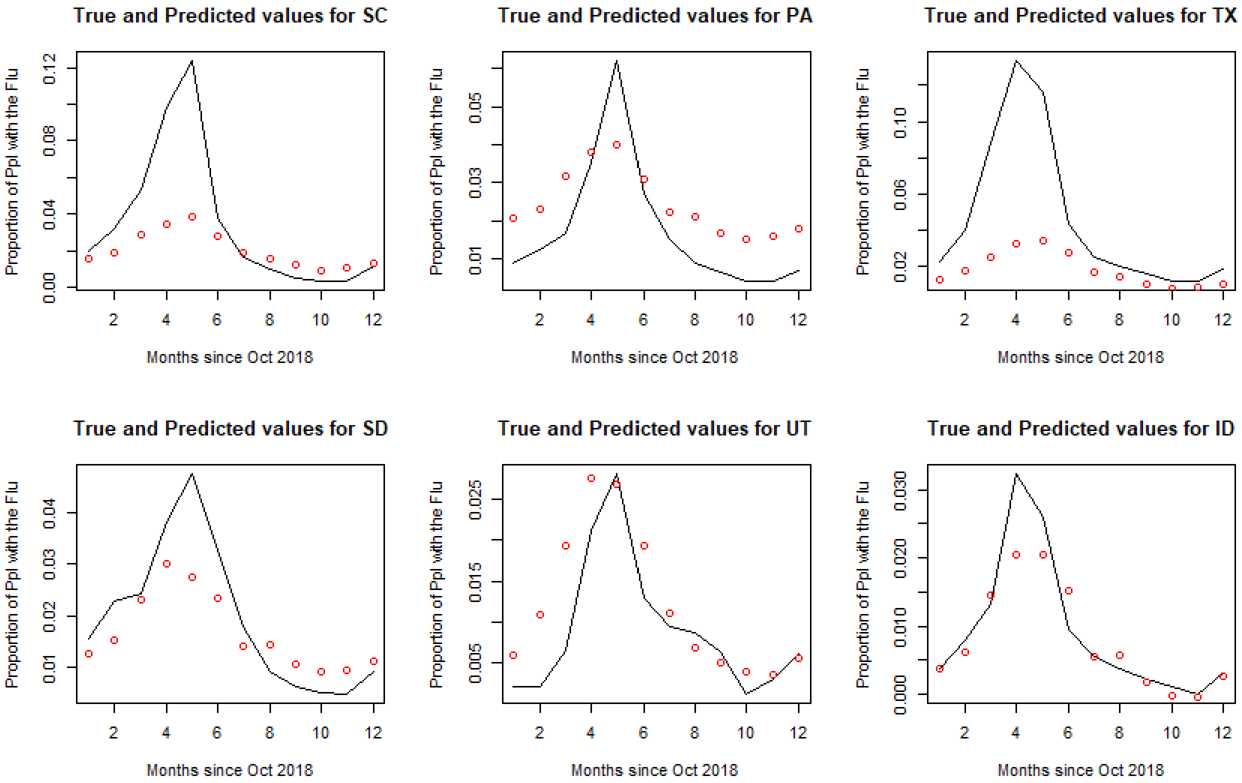
Based off of the HPD intervals, predictors 1,14, and 15 weren't significant corresponding to the baseline β_0, the precipitation, and population density of each state. Population density is understandable, but the insignificance of the precipitation is surprising.

The forecast is created using the sampled values of the parameters $\boldsymbol{\beta}$ and $\boldsymbol{b}$. Choosing six particular states from different regions of the United States (South Carolina, Pennsylvania, Texas, South Dakota, Utah, and Idaho), the true values (the line) were plotted along with the predicted values (the points).



3 Model Prediction

How well does this model do in predicting the flu outbreaks in the upcoming flu season? To test this, we split our data and responses into a training and testing set where our test set is the latest flu season, October 2017 to September 2018. We then built out model on the training set and tried to predict the proportion of flu cases in the test set. Looking at the same six states as before, the predicted and true proportion of flu cases are,



4 Conclusion

The model did an relatively good job at predicting the proportion of people with the flu. It was roughly split between over and under estimating the proportion of cases. To determine exactly how well they model did at predicting the proportion of flu cases, we'd need to do a goodness of fit test on all 47 states. Rather than determining how many individual cases might occur within a state, we decided to use the proportion since other models that the CDC uses have the weighted proportions as the response variables. The model did not quite predict the peaks well, but this could be because we are working with small values.

Overall, our model did fine with only four variables, and the HPD interval removes two of those. The CDC had data on the number of people in different (and also overlapping) age groups, but the headers of each column was abbreviated and had no intuitive explanation. Since young children and the elderly are more likely to get sick, if we had at least added the number of people in those categories into the model, we would expect the model to do better.

Another slight improvement we could do is add a Metropolis step to sample ρ rather than having it fixed. The reason we did not was because every time we needed to do an inversion of a matrix, it would take too long, and with 10000 iterations, it would take up too much time to run. But, since $\rho \in [-1, 1]$, we could have the proposal distribution as a reflected random walk if we wanted to do the Metropolis step.

```
library(maptools)
library(spdep)
library(maps)

###################################################################
###################################################################
#
# Neighborhood matrix for states
#
usa.state = map(database="state", fill=TRUE, plot=FALSE)
state.ID <- sapply(strsplit(usa.state$names, ":"), function(x) x[1])
usa.poly = map2SpatialPolygons(usa.state, IDs=state.ID)
usa.nb = poly2nb(usa.poly)
usa.adj.mat = nb2mat(usa.nb, style="B")

##Write the 0-1 adjacency matrix
W.raw = usa.adj.mat
#Remove D.C. and Florida
W=W.raw[c(1:7,10:49),c(1:7,10:49)]
colnames(W) <- rownames(W)
D = diag(colSums(W))


##########################################################
##########################################################
#
# Reading in our data
#

setwd("C:/Users/icamo/OneDrive/Documents/Clemson/Fall␣2018/MATH␣8820/Project
   ␣1")
library(coda)
library(tidyr)
library(MASS)

precip <- read.csv("Precipitation.csv", header= T, fileEncoding="UTF-8-BOM")
precip <- precip[precip$State != "FL", ]
precip <- precip[,1:14]


states <- droplevels(unique(precip$State))

temp <- read.csv("Temperature.csv", header= T, fileEncoding="UTF-8-BOM")
temp <- temp[temp$State != "FL", ]

unemp <- read.csv("Imputed_Unemployment_Data.csv", header = T, fileEncoding
   = "UTF-8-BOM")
colnames(unemp) <- c('Year', 'Month', as.character(states))

pop.land <- read.csv("PopulationLandArea.csv", header= T, fileEncoding="UTF
   -8-BOM")

pop.land <- pop.land[,c(5,7,8,9,10,11,12,13,14,15,58)]
pop.land <- pop.land[c(-2,-11),]
pop.land <- pop.land[pop.land$NAME != "Florida", ]

land.area <- pop.land[,11]
pop.density <- pop.land[,2:10]/land.area
pop.density <- cbind(pop.land$NAME, pop.density)

colnames(pop.density) <- c("State", "2010", "2011", "2012", "2013", "2014",
   "2015", "2016", "2017", "2018")

temp <- temp %>%
  gather('JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT
    ', 'NOV', 'DEC', key = "Month", value = "Deg.Far")
map = setNames(1:12, c('JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG
   ', 'SEP', 'OCT', 'NOV', 'DEC'))
temp[,3] <- map[unlist(temp[,3])]

precip <- precip %>%
  gather('JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL', 'AUG', 'SEP', 'OCT
    ', 'NOV', 'DEC', key = "Month", value = "Inch")
precip[,3] <- map[unlist(precip[,3])]

unemp <- unemp %>%
  gather(as.character(states), key = "State", value = "Unemp.Perc")
unemp <- unemp[with(unemp, order(State, Year, Month)),]

pop.density <- pop.density %>%
  gather('2010', '2011', '2012', '2013', '2014', '2015', '2016', '2017', '
    2018', key = "Year", value = "Ppl/SqMile")

pop.density$State <- states
pop.density$Year <- as.numeric(pop.density$Year)
pop.density <- pop.density[with(pop.density, order(State, Year)),]

# Cannot find average temperature or precipitation for September, so I'm
   making them
d <- na.omit(subset(temp, temp$Month == 9))
avg.temp <- aggregate(d[,4], list(d$State), mean)[,2]
var.temp <- diag(aggregate(d[,4], list(d$State), var)[,2])
create.Sept.temp <- round(mvrnorm(1,avg.temp, var.temp),1)
temp$Deg.Far[temp$Year == 2018 & temp$Month == 9] <- create.Sept.temp
temp <- na.omit(temp)

d <- na.omit(subset(precip, precip$Month == 9))
avg.precip <- aggregate(d[,4], list(d$State), mean)[,2]
var.precip <- diag(aggregate(d[,4], list(d$State), var)[,2])
create.Sept.precip <- round(mvrnorm(1,avg.precip, var.precip),1)
```

```
precip$Inch[precip$Year == 2018 & precip$Month == 9] <- create.Sept.precip
precip <- na.omit(precip)

# Removing months Jan-Sept for 2010 since we don't have flu data on those
   monts
temp <- temp[!(temp$Year == 2010 & temp$Month < 10),]
temp <- temp[with(temp, order(State,Year,Month)),]
precip <- precip[!(precip$Year == 2010 & precip$Month < 10),]
precip <- precip[with(precip, order(State,Year,Month)),]

# Repeat the population density values the required number of times so it
   has the same
# number of rows as temp and precip
pop.density <-data.frame(rep.int(unique(pop.density$State), times=rep(96,47)
   ), rep.int(pop.density$Year, times=rep(c(3, rep(12, 7), 9), 47)), rep.int(
   pop.density$`Ppl/SqMile`, times=rep(c(3, rep(12, 7), 9), 47)))
colnames(pop.density) <- c("State", "Year", 'Ppl.Sq.Mile')

flu <- read.csv("ILINet_monthly.csv", header = T)
flu[,7] <- flu$ILITotal/flu$TotalPatients
colnames(flu)[7] <- "Proportion"
flu$Proportion[is.na(flu$Proportion)] <- 0


# remove
flu$State = droplevels(flu$State)


# Changing the states into abbrevations
flu$State <- rep(states, each = 12*8)


Y <- flu$Proportion

# Making October (10th month) the baseline
indicators <- matrix(0, nrow = length(flu$Month), ncol = length(unique(flu$
   Month)))
indicators[cbind(seq_along(flu$Month), flu$Month)] <- 1
indicators <- indicators[,-10]
X <- cbind(1, indicators, temp$Deg.Far, precip$Inch, pop.density$Ppl.Sq.Mile
   , unemp$Unemp.Perc)

G <- matrix(0, nrow = length(flu$State), ncol = nlevels(flu$State))
G[cbind(seq_along(flu$State), flu$State)] <- 1



#
   ##########################################################################
```

```
# Necessary Packages

library(Matrix)
library(hierarchicalDS)
library(mvtnorm)
library(mnormt)
library(lattice)


#
  ######################################################################################

#
  ######################################################################################

#
  ######################################################################################

#
  ######################################################################################

#
# Full block Gibbs sampler:
#
#


flu.mcmc<-function(Y,X,D,W,rho,b,sig2,tau2,as2,bs2,at2,bt2,N,G,verbose=FALSE
    ){

  V<-dim(D)[1]
  n <- length(Y)
  DW<-D-rho*W
  I<-sparseMatrix(1:V,1:V,x=1)
  Ri <- solve(R)

  b.mcmc<-matrix(-99,nrow=N,ncol=V)
  sig2.mcmc<-rep(-99,N)
  tau2.mcmc<-rep(-99,N)
  beta0.mcmc<-matrix(-99,nrow = N,ncol=dim(X)[2])


  for(g in 1:N){

    ##################################################
    # Sample the intercept

    XtXR <- solve(t(X)%*%X + sig2*Ri)
    Y.s<-Y-G%*%b
```

```
beta0<-mvrnorm(1, XtXR%*%t(X)%*%Y.s, sig2*XtXR)

##################################################
# sample spatial random effects

Yt<-t(G)%*%(Y-X%*%beta0)

Prec<- t(G)%*%G + sig2*DW/(tau2)
CH<-chol(Prec)
R1<-solve(CH,rnorm(V,0,sqrt(sig2)),sparse=TRUE)
R2<-solve(t(CH),Yt,sparse=TRUE)
R3<-solve(CH,R2,sparse=TRUE)
b<-as.vector(R1+R3)



###################################################
# Sample sig2

as2s<-as2+n/2
bs2s<-as.vector(bs2+ sum((Y-X%*%beta0-G%*%b)^2)/2)

samp<-rgamma(1,as2s,bs2s)

sig2<-1/samp

###################################################
# Sample tau2

at2s<-at2+V/2
bt2s<-as.vector(bt2+ t(b)%*%DW%*%b/(2))

samp<-rgamma(1,at2s,bt2s)

tau2<-1/samp

sig2.mcmc[g]<-sig2
tau2.mcmc[g]<-tau2
beta0.mcmc[g,]<-beta0
b.mcmc[g,]<-b

if(verbose==TRUE){
  print(g)

  if(g%%100==0){
    par(mfrow=c(2,1))
    plot(sig2.mcmc[1:g])
    plot(tau2.mcmc[1:g])
  }
```

```
      }
    }

    return(list("sig2"=sig2.mcmc,"tau2"=tau2.mcmc,"beta0"=beta0.mcmc,"b"=b.
       mcmc))
}




#
   ############################################################################################

#
   ############################################################################################

#
   ############################################################################################

#
   ############################################################################################

#
# Initialize parameters
#
#

# Number of Iterations
N = 10000

# Beta
R = diag(rep(100,ncol(X)))

# Spatial
rho = 0.9
b = rep(0,47)

# Sigma2
sig2 = 2
as2 = 0
bs2 = 0


# Tau2
tau2 = 2
at2 = 0
bt2 = 0
```

```
#
  #############################################################################

#
  #############################################################################

#
  #############################################################################

#
  #############################################################################

#
# Run the model
#
#
flu.gibbs <- flu.mcmc(Y,X,D,W,rho,b,sig2,tau2,as2,bs2,at2,bt2,N,G,verbose=
  FALSE)

# Trace plot of beta's
par(mfrow = c(4,4))
for(i in 1:ncol(flu.gibbs$beta0)){
  plot(flu.gibbs$beta0[,i], ylab = paste("Beta",as.character(i-1)), main =
     paste("Trace␣Plot␣of␣Beta",i-1))
}

# Means of beta'
colMeans(flu.gibbs$beta0)

#
  #############################################################################

#
  #############################################################################

#
  #############################################################################

#
  #############################################################################

#
# Plotting chosen states' predicted and true values, one from each region of
    US
#
#
par(mfrow = c(2,3))
ch.states = c('SC', 'PA', 'TX', 'SD', 'UT', 'ID')
for(i in 1:length(ch.states)){
  hold.x <- data.frame(flu$State, X)
  hold.x <- subset(hold.x, hold.x[,1] == as.character(ch.states[i]))
  hold.y <- subset(flu, flu$State == ch.states[i])
  plot(hold.y$Proportion, xlab = "Months since Oct 2010", ylab = "Proportion
     of Ppl with the Flu", main = paste("True and Predicted values for", as.
     character(ch.states[i])), type = 'l')
  points(1:96, as.matrix(hold.x[,-1])%*%colMeans(flu.gibbs$beta0) + colMeans
     (flu.gibbs$b)[which(states == as.character(ch.states[i]))], col = 'red')
}


HPDinterval(as.mcmc(flu.gibbs$beta0), cred = 0.95) # var 6,14,15 not
   significant
HPDinterval(as.mcmc(flu.gibbs$sig2), cred = 0.95)
HPDinterval(as.mcmc(flu.gibbs$tau2), cred = 0.95)



#
   ##########################################################################################
#
   ##########################################################################################
#
   ##########################################################################################
#
# Training/Testing
#

X.train <- data.frame(flu$Year, X)
X.train <- as.matrix(subset(X.train, X.train$flu.Year != 2018 & !( flu$Year
   == 2017 & flu$Month %in% c(10,11,12)))[,-1])
Y.train <- flu[flu$Year != 2018 & !( flu$Year == 2017 & flu$Month %in% c
   (10,11,12)),]$Proportion
G <- matrix(0, nrow = length(Y.train), ncol = nlevels(flu$State))
G[cbind(seq_along(Y.train), flu$State[1:length(Y.train)])] <- 1

flu.gibbs.train <- flu.mcmc(Y.train,X.train,D,W,rho,b,sig2,tau2,as2,bs2,at2,
   bt2,N,G,verbose=FALSE)

X.test <- data.frame(flu$Year, X)
X.test <- subset(X.test, X.test$flu.Year == 2018 | (flu$Year == 2017 & flu$
   Month %in% c(10,11,12)))[,-1]

par(mfrow = c(2,3))
ch.states = c('SC', 'PA', 'TX', 'SD', 'UT', 'ID')
for(i in 1:length(ch.states)){
  hold.x <- data.frame(rep.int(states,times = rep(12,47)), X.test)
  hold.x <- subset(hold.x, hold.x[,1] == as.character(ch.states[i]))
  hold.y <- subset(flu, flu$State == ch.states[i] & (flu$Year == 2018 | (flu
    $Year == 2017 & flu$Month %in% c(10,11,12))))
  plot(hold.y$Proportion, xlab = "Months since Oct 2018", ylab = "Proportion
     of Ppl with the Flu", main = paste("True and Predicted values for", as.
    character(ch.states[i])), type = 'l')
  points(1:12, as.matrix(hold.x[,-1])%*%colMeans(flu.gibbs.train$beta0) +
    colMeans(flu.gibbs.train$b)[which(states == as.character(ch.states[i]))
    ], col = 'red')
}
```

References

[1] Centers for Disease Control and Prevention. *Seasonal Influenza (Flu)*. https://www.cdc.gov/flu/about/index.html. Updated 2018 Oct 26.

[2] World Health Organization. *Influenza (Seasonal)*. 2018.

[3] Lowen A, Steel J. *Roles of Humidity and Temperature in Shaping Influenza Seasonality*. https://jvi.asm.org/content/jvi/88/14/7692.full.pdf. 2014 July.

[4] Osthus D, Gattiker J, Priedhorsky R, Del Valle, S. *Dynamic Bayesian Influenza Forecasting in the United States with Hierarchical Discrepancy*. https://projecteuclid.org/download/pdfview_1/euclid.ba/1533866670. 2018.

[5] Centers for Disease Control and Prevention. *Overview of Influenza Surveillance in the United States*. https://www.cdc.gov/flu/weekly/overview.htm. Updated 2018 Oct 19.

[6] Biggerstaff M, Alper D, Dredze M, Fox S, Fung I, Hickmann K,.... *Results from the centers for disease control and prevention's predict the 2013-2014 Influenza Season Challenge*. https://doi.org/10.1186/s12879-016-1669-x.

[7] Centers for Disease Control and Prevention. *MMWR Weeks*. https://wwwn.cdc.gov/nndss/document/MMWR_Week_overview.pdf.

[8] Southern Regional Climate Center. *Monthly Summaries*. 2018.

[9] United States Census Bureau. *State Population Totals and Components of Change: 2010-2017*. https://www.census.gov/data/tables/2017/demo/popest/state-total.html.

[10] Bureau of Labor Statistics. *Local Area Unemployment Statistics*. https://www.bls.gov/lau.