Baysian Modeling of Hurrican trajectories

Group11: Sibei Liu, Xue Jin, Yuchen Qi, Xinru Wang

05/14/2020

Introduction

North Atlantic hurricanes claim a large toll in terms of fatalities and economic damage every year. Therefore, efforts to abtain deeper understanding of their physical mechanisms such as wind speed which is of top priority for the development and tracking is very important for both researchers and societal and economic repercussions.

Wind speed for a specific time point is related to the day of year at this time point, the calenda year of the hurricane, type of hurrican, change of latitude, longitude, wind speed, and wind speed before this time point. Using this relationship, we designed a MCMC algorithm to build a simplified Bayesian model for predicting wind speed, and then use these posteri means to track the remaining 20% test hurricanes.

Objectives

To predict future spped of each hurricane, our project maimly focused on using MCMC algorithm to abtain each coefficients in the simplified Bayesian model and then calculate mean square errors using test hurricanes.

Dataset

The dataset contains information 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

We tidied our data and filtered the data point which is not the integer multiple of six, and randomly assign 80% of hurricanes to be our training data, remaining is our test data.

Method

Form the prespecified imformation about suggested Baysian model:

$$Y_i(t+6) = \mu_i(t) + \rho Y_i(t) + \epsilon_i(t)$$

$$\mu_i(t) = \beta_0 + x_{i,1}(t)\beta_1 + x_{i,2}\beta_2 +]x_{i,3}\beta_3 + \sum_{k=1}^{3} \beta_{3+k}\Delta_{i,k}(t-6)$$

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

For $\epsilon_i(t)$ ~$Normal(0, \sigma^2)$ $\pi(\boldsymbol{\beta})$ ~ $MVN(\mathbf{0},\ diag(1,7))$, $\pi(\rho)$ follows a trucated normal $N_{[0,1]}(0.5, 1/5)$ $\pi((\sigma^2)^{-1})$ follows a inverse-gamma $(0.001, 0.001)$

Among those information, we find: Yi is a linear combination of μ_i and $\rho Y_i(t)$ and $\epsilon_i(t)$. While μ_i and $\rho Y_i(t)$ is a constant. So $Y_i(t+6)$ follow normal distribution with $\mu_{new}(t)$ and variance σ^2:

$$Y_i(t+6) \sim Normal(\mu_{new}(t), \sigma^2)$$

$$\mu_{new}(t) = \mu_i(t) + \rho Y_i(t)$$

To exclude the time series influence on Y_i we choose to use

$$\epsilon_i = Y_i(t+6) - \mu_{new} \sim Normal(0, \sigma^2)$$

The poesteria distribution

$$f(\epsilon_i|\boldsymbol{\beta}, \rho, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{(Y_i(t+6) - \mu_{new} - 0)^2}{2\sigma^2})$$

Poesterian distribution:

$$\pi(\boldsymbol{\beta}, \rho, \sigma^2|\epsilon_i) \propto \prod_{i=1}^{n}\prod_{k=1}^{m} f(\epsilon_i|\boldsymbol{\beta}, \rho, \sigma^2) * \pi(\boldsymbol{\beta}) * \pi(\rho) * \pi((\sigma^2)^{-1})$$

The i denote the ith hurricane, n is the total number of hurricane. The k denote the kth speed among this hurricane, m is the total measured speeds among this hurricane.

Take the log form:

$$\pi'(\boldsymbol{\beta}, \rho, \sigma^2|\epsilon_i) \propto \sum_{i=1}^{n}\sum_{k=1}^{m} log(f(\epsilon_i|\boldsymbol{\beta}, \rho, \sigma^2)) + log(\pi(\boldsymbol{\beta})) + log(\pi(\rho)) + log(\pi((\sigma^2)^{-1}))$$

We use the random walk in Metropolis-Hasting algorithm to find the poeteria distribution.$\boldsymbol{\lambda}$ is the proposed value of next step of parameters when using random walk, whose search window is a is pre-defined. $\boldsymbol{\theta} = (\boldsymbol{\beta}, \rho, \sigma^2)$

The probability of accepting is :

$$\alpha(\lambda|\boldsymbol{\theta}) = min(1, \frac{\pi(\boldsymbol{\lambda})q(\boldsymbol{\theta}|\boldsymbol{\lambda})}{\pi(\boldsymbol{\theta})q(\boldsymbol{\lambda\theta})})$$

while the $q(\theta|\lambda) = q(\lambda|\theta)$ so

$$\alpha(\lambda|\boldsymbol{\theta}) = min(1, \frac{\pi(\boldsymbol{\lambda})}{\pi(\boldsymbol{\theta})}) = min(1, \pi'(\boldsymbol{\lambda}) - \pi'(\boldsymbol{\theta}))$$

Compare the $\alpha(\lambda|\boldsymbol{\theta})$ with the random drawed uniform(0,1) number,if $\alpha(\lambda|\boldsymbol{\theta})$ is larger, accept the proposed $\boldsymbol{\lambda}$ otherwise still accept $\boldsymbol{\theta}$

Results

The starting value is (-26,0,0.012,0.009,0,0.01,1,0.1,1) and search window a = c(0.1,0.0001,0.0001,0.01,0.001,0.01,0.001, 0.02,0.0003), which corresponds to β_0, β_1, β_2, β_3, β_4, β_5, β_6, σ, ρ,. After 20000 iteration, we get the following path plots:

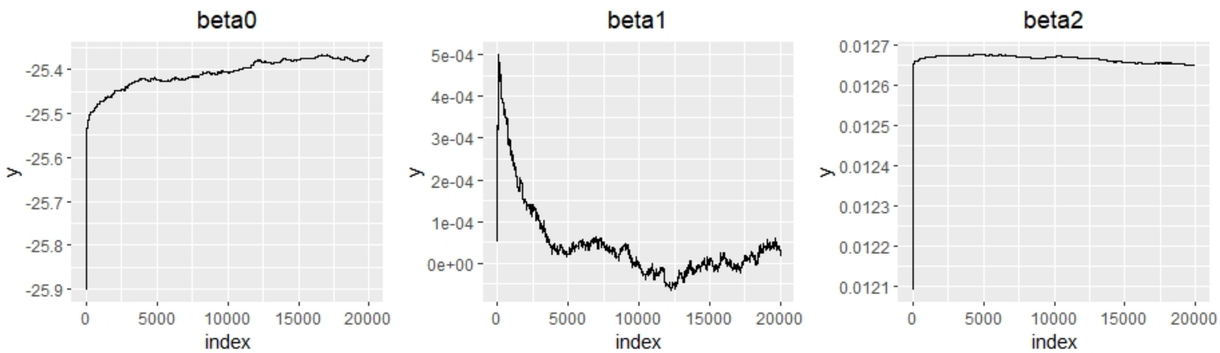

Figure 1. Path of β_0, β_1, β_2

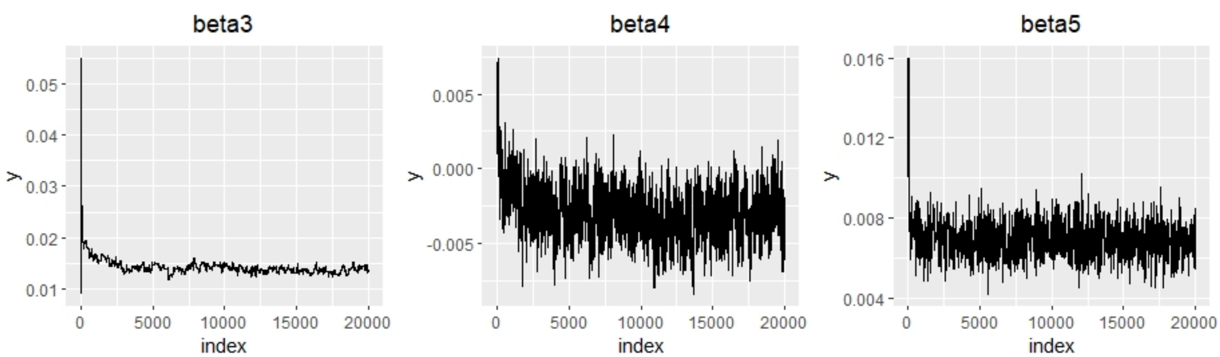

Figure 2. Path of β_3, β_4, β_5

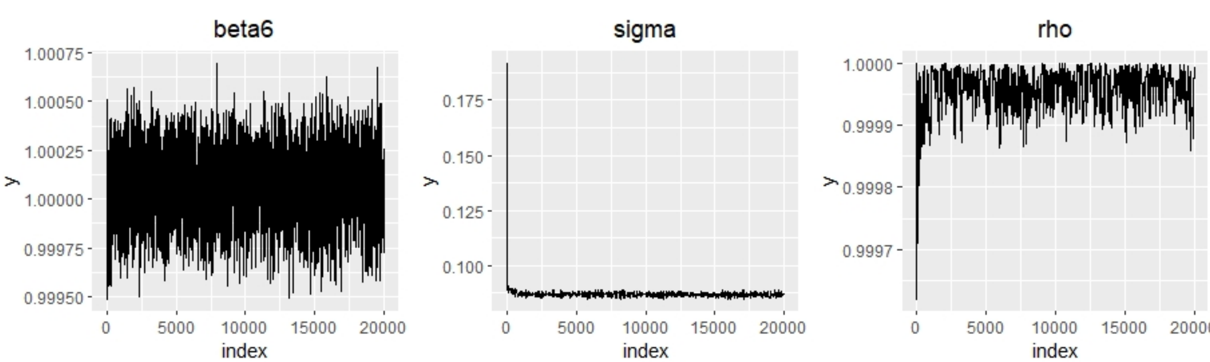

Figure 3. Path of β_6, σ, ρ

From above, we can tell that $\beta_3 - \beta_5$ and σ, ρ all almost enter into the stationary distribution after iteration reaches to 5000, while β_0 to β_2 don't enter the stable situation. When we use the 10000-20000 to calculate the poesterian mean to estimate the parameter and use 95% percentile to find the interval, the results is below. After using the mean as the parameters, the MSE of predicted spped and observed speed in test data is 22.74.

	mean	lower	upper
beta0	-25.3812000	-25.4044236	-25.3683768
beta1	-0.0000053	-0.0000541	0.0000473
beta2	0.0126588	0.0126501	0.0126717
beta3	0.0136956	0.0127268	0.0148076
beta4	-0.0033275	-0.0061918	-0.0003525
beta5	0.0068755	0.0054174	0.0083866
beta6	1.0000609	0.9997446	1.0003751
sigma	0.0867737	0.0855817	0.0881760
rho	0.9999641	0.9999037	0.9999993

Table 1. Mean and quantile of parameters

Discussion

A possible explanation for the chain fail to converge to the desired stationary distribution for β_0 to β_2 is that the starting point we choose is not good enough. This issue may be solved by starting at a point, like the mode, known to have reasonably high probability, but no such point is known. An alternative solution is to run the chain for many steps so it can converge and use the burn-in method. However this may need a large number of iterations which exceeds our computer capacity, and usually it should not need over 10,000 iterations to enter the stationary condition, so the inappropriate "moving length" a may account for the unsatisfactory plots of β_0 to β_2. We tried different margin, such as several numbers between 0.01 and 0.2 for β_0, but none of these numbers performed well. As it took a long time to generate a chain each time, we could not explore more chloices of a, which may provide a better solution.