Markel Sanz Ausin (msanzau@ncsu.edu)

Applied Bayesian Analysis, Exam 2

1. Introduction

We used Bayesian methods to build a model that uses sea surface temperature (SST) measurements from 10 different locations in the Atlantic ocean during 6 different months to estimate the number of tropical storms that hit the US coast during that year.

2. Methods

All our methods will consist of a Poisson regression, due to the fact that the number of tropical storms always needs to be a non-negative integer number. As the parameter for the Poisson regression needs to be positive, we will use a regression model that gets exponentiated, thus making it positive. This allows us to use different priors, including priors that can produce negative sample values, such as the normal distribution. The different methods employed will represent the regression in a different way, accounting for random effects sometimes, using different priors, and evaluating the performance of each of these models.

First, we must say we implemented some very simple models, such as calculating the average of all the observations across the different months for each location, and building a Poisson regression with those averages as new covariates. But these models were too simple and did not provide a very good fit, so we will not describe them any further.

The first model we implemented and fit the data relatively well, which we will refer to as **Model 1**, was a simple linear regression, where each of the 60 covariates is multiplied by its corresponding parameter for the month and its corresponding parameter for the location. This way, we have 16 parameters and 49 observations, which is reasonable. And this model incorporates the shared information, meaning that the 6 different observations for some location

will get multiplied by the same location parameter, but they will all be multiplied by different month parameters. This is summarized in Equation 1.

$$\log(\lambda_i) = \alpha + \Sigma\,(X_{kji} * B_k * B_j) \quad \text{and} \quad Y_i \sim \text{Poisson}(\lambda_i) \qquad (1)$$

Here, i indicates the observation number, k indicates the location and j indicates the month. The priors selected for this model were normal distributions, but they were not fixed, we allowed each of these normal distributions to have its own mean and variance.

The second model we implemented (**Model 2**) was a simple linear regression on the 60 different covariates, but also accounting for the random effects of the correlations found in each location. For this reason, Equation 2 shows how we added a vector of 10 parameters named theta ($\boldsymbol{\theta}$), which creates a single value per location that is then multiplied by each covariate.

$$\log(Y_i) = \alpha + \Sigma\,((X_{kji} * B_{kj}) \,\%*\%\, \boldsymbol{\theta}_k) \qquad (2)$$

Here, %*% indicates the matrix multiplication operation. The priors for this model were also normal, and critically, we did not fix the values for the prior, we let each value come from a different mean and prior, which we found to be really important for convergence. We also tried the double exponential prior, with the intent of building a simpler model where some parameters are irrelevant, but it did not show any improvement.

Finally, our third model (**Model 3**), used Stochastic Search Variable Selection (SSVS) in order to simplify the model and make several parameters be very close to zero, thus making them irrelevant in our model. This is done by making the beta parameters have a prior that consist of the multiplication of other two parameters: delta $\boldsymbol{\delta}$ (which comes from a normal distribution), and gamma γ (which comes from a bernoulli distribution). This can be seen in Equation 3. This model makes the gamma decide how relevant each covariate will be to the final model. When the posterior for gamma is centered around 0 and with a small variance, that covariate will likely be irrelevant and can be removed from the model.

$B_{ij} = \gamma_{ij} * \boldsymbol{\delta}_{ij}$ where $\gamma \sim$ Bernoulli(0.5) and $\boldsymbol{\delta} \sim$ N(0, sigma2) (3)

3. Computation

We used the software named JAGS, and we used R as an interface to access JAGS. This piece of software automatically estimates the posterior distribution using MCMC algorithms, such as Gibbs Sampling when the prior and likelihood are conjugate, and Metropolis Hastings when they are not. For all of our models, we used two different chains that can be used to evaluate whether convergence has been found or not, and we collected 20,000 samples (after 10,000 burning samples) from the posterior distribution, and a thinning factor of 10. Most of our models converged pretty well. We noticed that not fixing the mean and variance of the beta parameters in their prior helped a lot with convergence across several models. Overall, most or all of the parameters in the three chosen models converged, with Model 1 having the worst convergence values of the three models.

4. Model Comparisons

We used DIC and WAIC, as well as convergence diagnostics and model complexity in order to compare the different models. Model 1 shows that the convergence is not ideal. When looking at the diagnostics some parameters have converged and have a Gelman-Rubin diagnostic value that is close to 1, but others show a value of 1.08 or 1.09. The effective sample sizes for Model 1 are between 75 and 500 for most parameters, which is not large enough. When looking at the performance for this model, the DIC value shows a mean deviance of 292.4 with a penalty of 19.4. The fit for this model is not great but it is a pretty simple model so the penalty is not very large. The WAIC shows a value of 337.6 with a penalty of 34.6.

Model 2 has much better convergence, where all the 60 parameters and the 10 location parameters have converged. The DIC computation for Model 2 shows that the mean deviance is

250 with a penalty of 30.2. The WAIC shows a value of 281 with a penalty of 22. So both of these metrics agree on the fact that Model 2 is better than Model 1.

Finally, Model 3 has really good convergence values, so we can be confident that we have found a good solution with this model. It also shows a similar DIC value with a mean deviance of 254 with a penalty of 30. The WAIC value shows a value of 287 with penalty 24.

5. Results

We have decided to use Model 3 as our best model, since it can be simplified by removing some covariates, and it still provides a good fit for the data. Looking at the results and analyzing the location parameters, we observe that all locations except for location 2 have a positive correlation with the number of storms. Location 10 has the highest correlation, with 0.38 (this is the average of all the months for this location). Locations 6, 7, 8, and 9 also have a significant positive correlation. Regarding the months, our data shows that month 4 is the one with the highest impact, and after that month 2 is the next one. Months 1 and 3 have very little impact, as their parameters are close to zero.

6. Prediction

The prediction for the last data point (year 50), shows that the 95% interval for the posterior of Y is (3, 26), the median is 12, and the mean is 12.54. So this is the number of tropical storms we are expecting to see in the last year, given the observed sea surface temperatures. There is a 95% probability that it will be between 3 and 26.

A drawback of using a Poisson likelihood is that the mean is equal to the variance. This makes the 95% interval of this model be pretty large. I did not have time, but I would have used a Negative Binomial likelihood in order to be able to better control the variance, and get a model that produces more confident predictions.

CODE for **Model 3**:

```
params <- c("alpha", "beta", "Y_pred")
data <- list(Y=Y, X=X, n=n)
model_string = textConnection("model{
  # Likelihood
  for (i in 1:n) {
    Y[i] ~ dpois(mnY[i])
    for (row in 1:6) {
      for (col in 1:10) {
        res[row, col, i] <- X[row,col,i] * beta[row,col]
      }
    }
    log(mnY[i]) <- alpha + sum(res[,,i])
  }
  # PREDICT
  for (row in 1:6) {
    for (col in 1:10) {
      pred[row, col] <- X[row,col,50] * beta[row,col]
    }
  }
  log(mnY_pred) <- alpha + sum(pred)
  Y_pred ~ dpois(mnY_pred)
  # Priors
  for (row in 1:6) {
    for (col in 1:10) {
      beta[row, col] <- gamma[row,col]*delta[row,col]
      gamma[row, col] ~ dbern(0.5)
      delta[row, col] ~ dnorm(0, tau)
    }
  }
  tau ~ dgamma(0.1, 0.1)
  alpha ~ dnorm(0, 0.001)
  # WAIC
  for (i in 1:n) {
    like[i] <- dpois(Y[i], mnY[i])
  }
  }")
```