ST 540: Final Exam

Lee Pixton

May 10, 2021

Introduction

Simulation studies are often used in Bayesian statistics to study and compare frequentist methods. In these studies, multiple datasets are generated with known parameters and a Bayesian analysis is conducted on each dataset. The results of the algorithms can then be directly compared to the true known parameter values to determine how well the analysis functioned. In this analysis, we conduct a simulation study to investigate the performance of Bayesian non-linear regression. The simulated dataset investigates the measured pollen count across one year, and we will study the predictions of four separate beta values for multiple true parameters.

Bayesian Model

The model that will be used for the pollen count data will be $Y_t = \mu_t + \varepsilon_t$, where:

$$\mu_t = b_1 + b_2 * e^{(-\frac{(t-b_3)^2}{2b_4^2})}$$

The errors are normally distributed with mean 0 and variance σ^2 with correlation $Cor(\varepsilon_t, \varepsilon_{t+h}) = \rho^h$. The mean curve has four parameters, which we will be estimating in the analysis. b_1 is the baseline pollen mean, b_2 is the increase at peak pollen season, b_3 is the day of the year which contains the highest mean point and b_4 controls the width of the peak curve and must therefore be greater than 0.

The priors that are used for each b value will be uninformative:

$$b_1, b_2 \sim N(0, sd(Y)^2)$$

$$b_3 \sim U(0, 365)$$

$$b_4 \sim G(0.1, 0.1)$$

The prior for σ^2 will also be uninformative:

$$\sigma^2 \sim G(0.1, 0.1)$$

Computation

The full model described above was run using Markov Chain Monte Carlo (MCMC) sampling in the R programing language on each simulated dataset. The software JAGS was used and integrated into R with the library 'rjags', which was used to facilitate the MCMC. This MCMC was integrated into a user function included at the end of this report and repeated for each combination of n and rho_true.

Due to the nature of the model, we provide initial values for the MCMC sampling. The formulas used for these values are shown below.

$$b_1 = mean(Y)$$

$$b_2 = max(Y) - mean(Y)$$

$$b_3 = t, where\ max(Y)$$

$$b_4 = (t, (where\ max(Y) + 1)) - (t, (where\ max(Y) - 1))$$

b_1 takes on the mean value of the pollen count Y, b_2 takes the max(Y) - b_1, b_3 takes on the value of the "day" t where pollen count is highest, and b_4 takes the value of t one above b_3 and subtracts that from the value of t one below b_3 which estimates the width of the peak curve.

Two chains were run for each simulated dataset, including a discarded burn-in of 10,000. Each chain then ran for 20,000 iterations with no thinning. Different priors were tested in a sensitivity analysis, and the outcome from early models were not shown to be sensitive to the prior. Following the sampling, the trace plots were examined, and each indicated convergence for the different models. The effective sample sizes were large and the Gelman and Rubin's Convergence Diagnostic were about 1 for all models further pointing to convergence.

Data Generation

Data was simulated using provided code to create the curve described above. An example plot of the generated data is shown in figure 1. The general shape of the data is flat with an increase, peak, and decrease around the early springtime of the year. The true parameters of the data include n, the number of observations in the dataset, b_true, the true *b* values described previously, sig_true, the true value of sigma, and rho_true, the true value of rho. b_true and sig_true were held constant at values of *b* = (10, 50, 100, 10) for 1 to 4 respectively, and sig_true = 1. The parameters n and rho_true were adjusted for each separate simulation study. The values chosen for n were 50, 100, 200, and 365. The values for rho_true were 0.1, 0.25, 0.5, and 0.99. There were thus 16 separate simulations studies run with each combination of these two adjusted parameters.

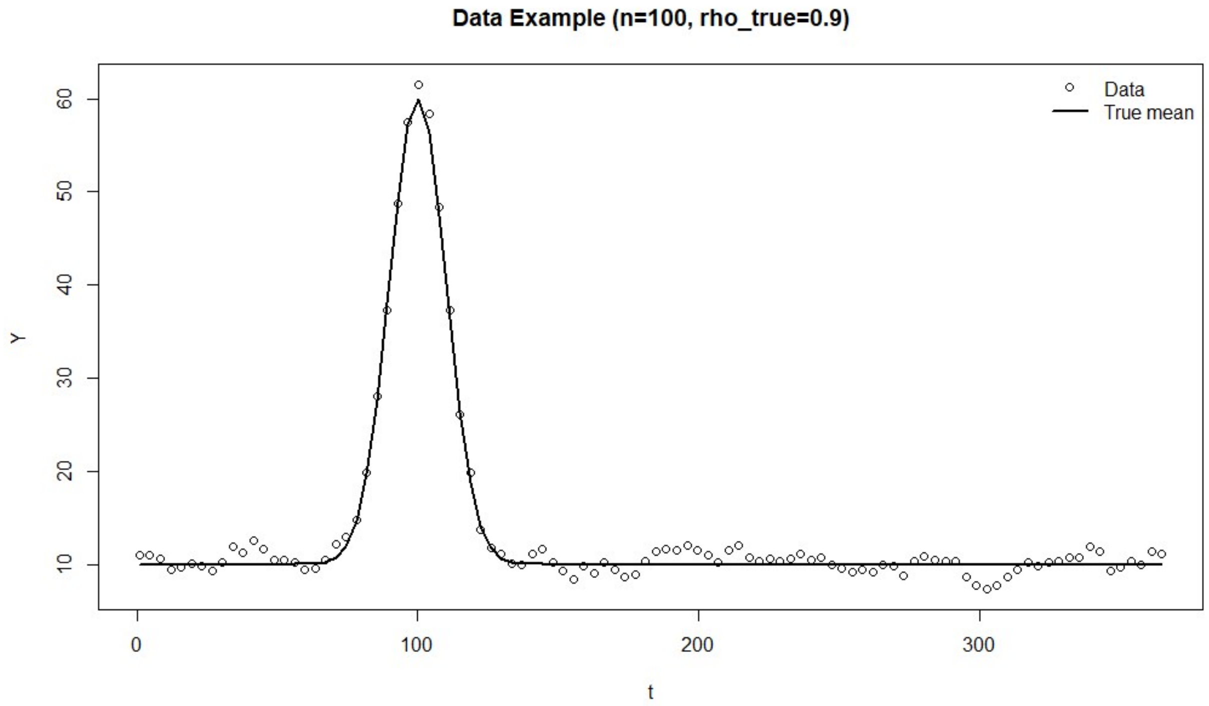


Figure 1: Simulated data example with n = 100 and rho_true = 0.9

Metrics

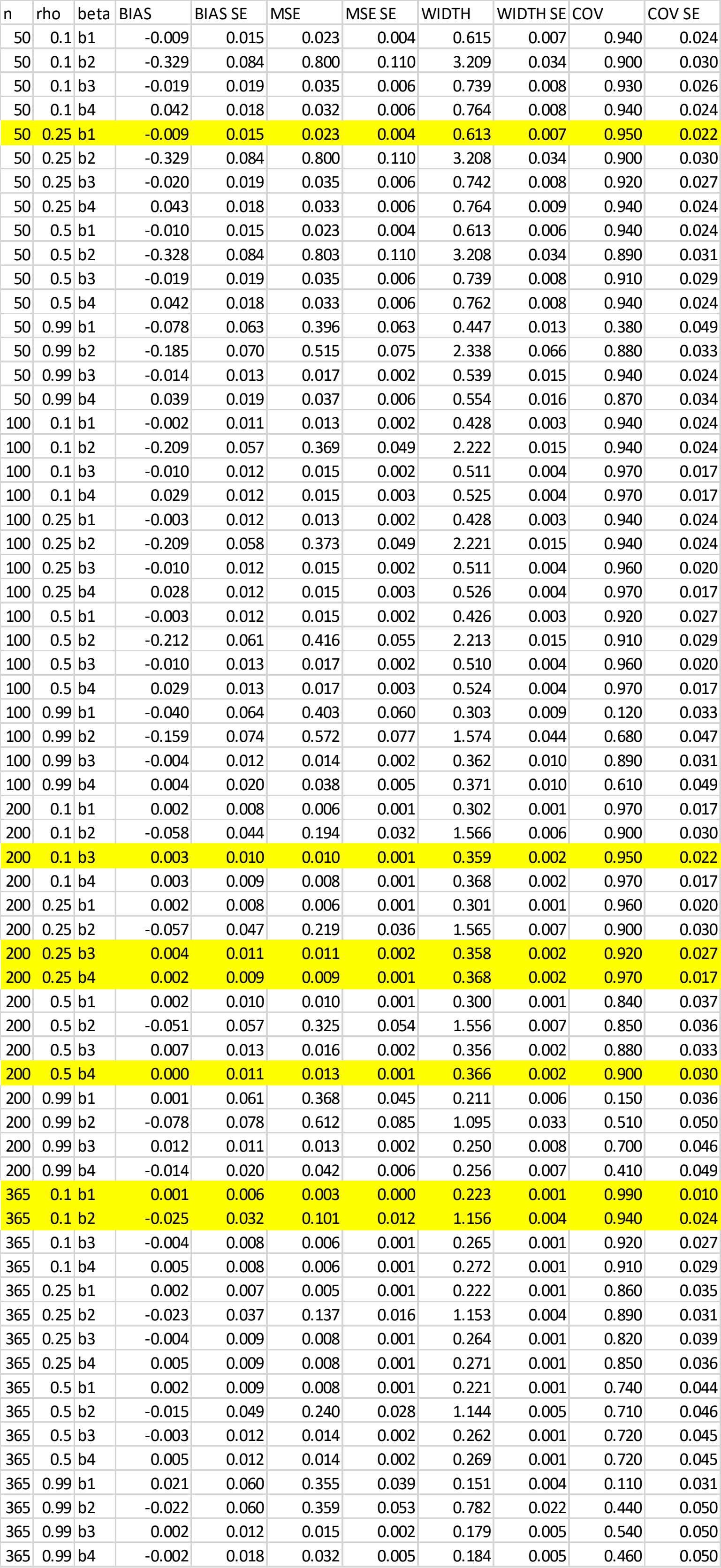
There are four metrics that were used for analyzing the performance of the Bayesian inference. These are Bias, Mean Squared Error (MSE), Width and Coverage. These metrics were calculated for each *b* value in each simulation, and the results as well as standard errors for these metrics are shown in table 1.

The Bias indicates how much the estimated *b* values differ from the true values of *b* on average. The formula for this is: $Bias = mean(b_{mean} - b_{true})$ for each *b*.

The MSE describes the mean of the squared differences between the estimated *b* and the true *b*. The formula used for this calculation is: $MSE = mean((b_{mean} - b_{true})^2)$ for each *b*.

The Width calculation simply summarizes the average range of the 95% credible intervals for each estimated *b*. The formula for this calculation is: $Width = mean(b_{0.975} - b_{0.025})$ for each *b*.

Finally the Coverage is used to calculate how often the true *b* appears in the 95% credible intervals from each simulation study on average, with the formula being: $Coverage = mean((b_{0.025} < b_{true})\ \&\ (b_{true} < b_{0.975}))$ for each *b*.



n	rho	beta	BIAS	BIAS SE	MSE	MSE SE	WIDTH	WIDTH SE	COV	COV SE
50	0.1	b1	-0.009	0.015	0.023	0.004	0.615	0.007	0.940	0.024
50	0.1	b2	-0.329	0.084	0.800	0.110	3.209	0.034	0.900	0.030
50	0.1	b3	-0.019	0.019	0.035	0.006	0.739	0.008	0.930	0.026
50	0.1	b4	0.042	0.018	0.032	0.006	0.764	0.008	0.940	0.024
50	0.25	b1	-0.009	0.015	0.023	0.004	0.613	0.007	0.950	0.022
50	0.25	b2	-0.329	0.084	0.800	0.110	3.208	0.034	0.900	0.030
50	0.25	b3	-0.020	0.019	0.035	0.006	0.742	0.008	0.920	0.027
50	0.25	b4	0.043	0.018	0.033	0.006	0.764	0.009	0.940	0.024
50	0.5	b1	-0.010	0.015	0.023	0.004	0.613	0.006	0.940	0.024
50	0.5	b2	-0.328	0.084	0.803	0.110	3.208	0.034	0.890	0.031
50	0.5	b3	-0.019	0.019	0.035	0.006	0.739	0.008	0.910	0.029
50	0.5	b4	0.042	0.018	0.033	0.006	0.762	0.008	0.940	0.024
50	0.99	b1	-0.078	0.063	0.396	0.063	0.447	0.013	0.380	0.049
50	0.99	b2	-0.185	0.070	0.515	0.075	2.338	0.066	0.880	0.033
50	0.99	b3	-0.014	0.013	0.017	0.002	0.539	0.015	0.940	0.024
50	0.99	b4	0.039	0.019	0.037	0.006	0.554	0.016	0.870	0.034
100	0.1	b1	-0.002	0.011	0.013	0.002	0.428	0.003	0.940	0.024
100	0.1	b2	-0.209	0.057	0.369	0.049	2.222	0.015	0.940	0.024
100	0.1	b3	-0.010	0.012	0.015	0.002	0.511	0.004	0.970	0.017
100	0.1	b4	0.029	0.012	0.015	0.003	0.525	0.004	0.970	0.017
100	0.25	b1	-0.003	0.012	0.013	0.002	0.428	0.003	0.940	0.024
100	0.25	b2	-0.209	0.058	0.373	0.049	2.221	0.015	0.940	0.024
100	0.25	b3	-0.010	0.012	0.015	0.002	0.511	0.004	0.960	0.020
100	0.25	b4	0.028	0.012	0.015	0.003	0.526	0.004	0.970	0.017
100	0.5	b1	-0.003	0.012	0.015	0.002	0.426	0.003	0.920	0.027
100	0.5	b2	-0.212	0.061	0.416	0.055	2.213	0.015	0.910	0.029
100	0.5	b3	-0.010	0.013	0.017	0.002	0.510	0.004	0.960	0.020
100	0.5	b4	0.029	0.013	0.017	0.003	0.524	0.004	0.970	0.017
100	0.99	b1	-0.040	0.064	0.403	0.060	0.303	0.009	0.120	0.033
100	0.99	b2	-0.159	0.074	0.572	0.077	1.574	0.044	0.680	0.047
100	0.99	b3	-0.004	0.012	0.014	0.002	0.362	0.010	0.890	0.031
100	0.99	b4	0.004	0.020	0.038	0.005	0.371	0.010	0.610	0.049
200	0.1	b1	0.002	0.008	0.006	0.001	0.302	0.001	0.970	0.017
200	0.1	b2	-0.058	0.044	0.194	0.032	1.566	0.006	0.900	0.030
200	0.1	b3	0.003	0.010	0.010	0.001	0.359	0.002	0.950	0.022
200	0.1	b4	0.003	0.009	0.008	0.001	0.368	0.002	0.970	0.017
200	0.25	b1	0.002	0.008	0.006	0.001	0.301	0.001	0.960	0.020
200	0.25	b2	-0.057	0.047	0.219	0.036	1.565	0.007	0.900	0.030
200	0.25	b3	0.004	0.011	0.011	0.002	0.358	0.002	0.920	0.027
200	0.25	b4	0.002	0.009	0.009	0.001	0.368	0.002	0.970	0.017
200	0.5	b1	0.002	0.010	0.010	0.001	0.300	0.001	0.840	0.037
200	0.5	b2	-0.051	0.057	0.325	0.054	1.556	0.007	0.850	0.036
200	0.5	b3	0.007	0.013	0.016	0.002	0.356	0.002	0.880	0.033
200	0.5	b4	0.000	0.011	0.013	0.001	0.366	0.002	0.900	0.030
200	0.99	b1	0.001	0.061	0.368	0.045	0.211	0.006	0.150	0.036
200	0.99	b2	-0.078	0.078	0.612	0.085	1.095	0.033	0.510	0.050
200	0.99	b3	0.012	0.011	0.013	0.002	0.250	0.008	0.700	0.046
200	0.99	b4	-0.014	0.020	0.042	0.006	0.256	0.007	0.410	0.049
365	0.1	b1	0.001	0.006	0.003	0.000	0.223	0.001	0.990	0.010
365	0.1	b2	-0.025	0.032	0.101	0.012	1.156	0.004	0.940	0.024
365	0.1	b3	-0.004	0.008	0.006	0.001	0.265	0.001	0.920	0.027
365	0.1	b4	0.005	0.008	0.006	0.001	0.272	0.001	0.910	0.029
365	0.25	b1	0.002	0.007	0.005	0.001	0.222	0.001	0.860	0.035
365	0.25	b2	-0.023	0.037	0.137	0.016	1.153	0.004	0.890	0.031
365	0.25	b3	-0.004	0.009	0.008	0.001	0.264	0.001	0.820	0.039
365	0.25	b4	0.005	0.009	0.008	0.001	0.271	0.001	0.850	0.036
365	0.5	b1	0.002	0.009	0.008	0.001	0.221	0.001	0.740	0.044
365	0.5	b2	-0.015	0.049	0.240	0.028	1.144	0.005	0.710	0.046
365	0.5	b3	-0.003	0.012	0.014	0.002	0.262	0.001	0.720	0.045
365	0.5	b4	0.005	0.012	0.014	0.002	0.269	0.001	0.720	0.045
365	0.99	b1	0.021	0.060	0.355	0.039	0.151	0.004	0.110	0.031
365	0.99	b2	-0.022	0.060	0.359	0.053	0.782	0.022	0.440	0.050
365	0.99	b3	0.002	0.012	0.015	0.002	0.179	0.005	0.540	0.050
365	0.99	b4	-0.002	0.018	0.032	0.005	0.184	0.005	0.460	0.050

Table 1: b value metrics for each n and rho simulation data

Results

For each b_1, we see that smaller values of rho and larger values of n lead to overall better results. In fact, when n = 365 and rho = 0.1, there is a 99% coverage with a 0.01 standard error. For almost every b_2, as n increases and rho decreases, the absolute value of the bias and the MSE improves, but coverage actually decreases. This is interesting as the precision improves where many more b_2 values are close to the true b_2, but the width of the credible interval decreases and thus the coverage decreases as well when rho increases. Overall, the best combination for b_2 was n = 365 and rho = 0.1. For b_3, smaller values of n lead to worse bias but better coverage. On the other hand, as rho increases the coverage decreases but the bias improves. The best results for b_3 came from n = 200 and rho = 0.1. Finally, b_4 follows a similar pattern, with bias and MSE and coverage overall improving as n increases. As rho increases though, the bias and MSE improve, but the coverage decreases dramatically. The best parameters for b_4 were n = 200 and rho = 0.25.

The best parameter combinations for each *b* are highlighted in yellow in table 1. Looking at the parameter combinations shows that the best overall combination to estimate the *b* values accurately was n = 365 and rho = 0.1. The bias and MSE for each of the *b*'s at this simulation point are relatively low and the coverage is above 91% for each as well.

Discussion

Overall, as n increases and at lower values of rho, the *b* predictions performed better. b_4 appears to be the best estimated parameter, followed by b_1, b_3, and b_2. b_2 especially has relatively poor performance when compared to the other three *b* values which could be related to the correlation that exists in the data.

It is very interesting that as rho increases, the coverage decreases. This is because the model we have chosen does not account for the correlation that exists between each Y value. There is built in correlation for the simulated data and as shown in the results, not modelling and accounting for this correlation leads to more mistakes when the true correlation is especially high.

In order to model true pollen data, correlation would have to be taken into account. It intuitively makes sense that the amount of pollen on any given day is correlated with the pollen count on both the previous and following day. The model used to define the real data would have to account for this. It would be important to measure this correlation as a whole and attempt to model it, which may require more parameters to effectively do this.

In addition, we would ideally have data for every day of the year, to have an n = 365 for the dataset. Knowing that having more data is better, even if we are unable to have n = 365 observations we can use MCMC sampling to create a posterior distribution and estimate the parameters.

The last point to consider is that there may be multiple peaks in the true data rather than just the one that appeared in the simulated data. In order to account for this, we would need to explore the shape of the true data and take this all into account when modelling this pollen data.

Code Appendix

```
### 540 Final

pollen_fit <- function(t,Y){

  # INIT VALUES
  init1 <- mean(Y)
  init2 <- max(Y) - init1
  init3 <- t[which.max(Y)]
  init4 <- t[which.max(Y) + 1] - t[which.max(Y) - 1]

  inits <- list(b=c(init1,init2,init3,init4))

  model_string <- textConnection("model{
    for(i in 1:n){
      Y[i]  ~ dnorm(mu[i],tau)
      mu[i] <- b[1] + b[2] * exp(-((t[i] - b[3])^2)/(2 * b[4]^2))
    }
    b[1] ~ dnorm(0,precision)
    b[2] ~ dnorm(0,precision)
    b[3] ~ dunif(0,365)
    b[4] ~ dgamma(0.1,0.1)
    tau ~ dgamma(0.1,0.1)
 }")

  data <- list(Y=Y, t=t, n=length(Y),precision=1/(sd(Y))^2)
  model <- jags.model(model_string,data = data, init=inits,n.chains=2,quiet=TRUE)
  update(model, 10000, progress.bar="none")
  samples <- coda.samples(model, variable.names=c("b"),
                       n.iter=20000, progress.bar="none")[[1]]
  b1 <- samples[,1]
  b2 <- samples[,2]
  b3 <- samples[,3]
  b4 <- samples[,4]
  out <- list("mean"=list(mean(b1),mean(b2),mean(b3),mean(b4)),
            "CI" = list(quantile(b1,c(0.025,0.975)),quantile(b2,c(0.025,0.975)),
                       quantile(b3,c(0.025,0.975)),quantile(b4,c(0.025,0.975))))

  return(out)}
```