STAT 447C Final Project Report

Team member: Jack Wu
Project theme: Hierarchical Bayesian regression
Github link: https://github.com/jack-wu05/Bayes-Regression/tree/main

Real-world & Methodological Background

According to Alzheimer's Disease International, someone in the world develops Alzheimer's dementia every 3 seconds. With roughly 55 million suffering with the disease globally, this number is projected to practically double every 20 years, hitting 139 million in 2050. The need for early detection and treatment will only become more pressing. To this end, in this project, I will be showcasing a hierarchical Bayesian regression model using STAN to capture the development of Alzheimer's in individual patients.

A common numeric metric to evaluate the severity of Alzheimer's in a patient is their score on the Mini-Mental State Examination (MMSE), a simple test for orientation and memory. The lower the score, the more severe the condition. Socioeconomic status (SES) is a composite numeric measure for an individual's economic and social position within society. The EDUC metric captures an individual's education level, and the Group metric is binary, either "demented" (1) or "non-demented" (0). My dependent variable is the MMSE score, and my independent variables are Age, SES, EDUC, Group, and visit number. The visit number highlights the unique structure of my chosen dataset.

The Kaggle Oasis Longitudinal Alzheimer's dataset contains several observations (1-4) for any given patient, indexed by the visit number when the data in that row were taken. My regression model will forecast, for each individual patient, the progression of MMSE scores in future visits. To motivate the hierarchical aspect of the modeling, observe that for the vast majority of patients, there are very few visits/observations (2-3, on average). I aim to exploit the hierarchical structure to pool together data from all the patients and construct robust and informed regressions for individual patients.

Literature Review

Li et al. (2020) used a Hierarchical Bayesian change point model with parameter constraints to forecast the deterioration of brain function in Alzheimer's patients. Each patient had unique random intercepts, random slopes before the change points, random change point times, and random slopes after the change points. This approach allowed the authors to distinguish between normal aging and accelerated brain decline from Alzheimer's, allowing for effective classification of patients. Bartolucci et al. (2009) used a Bayesian hierarchical change point model to detect periods of accelerated deterioration in already terminal patients. Methodologically, unconstrained parameters were utilized, in contrast to the approach taken by Li et al. To motivate this project, note that these cited studies were conducted over long periods of time. Bartolucci et al. conducted a 4-year long longitudinal study with many observations taken. Li et al.'s study lasted 20 years. The dataset I am using has very little data for each patient, again with 2-3 observations on average. The change point approach, which is popular in literature for Alzheimer's detection, requires much more data, and would not be sound in my case. I propose a simple hierarchical regression as an alternative.

The Model
The Bayesian model considered is as follows:

$$\begin{aligned}
\delta_1 &\sim \mathcal{N}(0,1), \quad \sigma_1 \sim \text{Exponential}(0.5)\\
\delta_2 &\sim \mathcal{N}(-1,2), \quad \sigma_2 \sim \text{Exponential}(0.5)\\
\alpha_i &\sim \mathcal{N}(\delta_1, \sigma_1^2)\\
\gamma_i &\sim \mathcal{N}(\delta_2, \sigma_2^2)
\end{aligned}$$

$$\begin{aligned}
\beta_1, \beta_4 &\sim \mathcal{N}(-1,3)\\
\beta_2, \beta_3 &\sim \mathcal{N}(1,3)\\
\mu_{ij} &= \alpha_i + \beta_1 \cdot \text{Age}_i + \beta_2 \cdot \text{SES}_i + \beta_3 \cdot \text{EDUC}_i + \beta_4 \cdot \text{Group}_i + \gamma_i \cdot \text{Time}_{ij}
\end{aligned}$$

$$\begin{aligned}
\sigma &\sim \text{Exponential}(0.1)\\
\text{MMSE}_{ij} &\sim \mathcal{N}(\mu_{ij}, \sigma^2)
\end{aligned}$$

The indexing with $Something_{ij}$ is read as "the value of Something for patient i at time j." The Time variable refers to the visit number. We now motivate the priors.

The prior on δ_1 describes belief about the mean intercept. The weakly informative prior of a standard Gaussian is natural, as intercepts should be centered around 0 with some flexibility to move around. The prior on δ_2 describes belief about the mean effect of time, which we model using the visits. We expect that in future visits, when more time has passed and the patient has aged, there is a higher likelihood of Alzheimer's, lower MMSE, and hence the -1 prior mean. We place uninformative priors on $\sigma_1, \sigma_2, \sigma$ that have $\mathbb{R}_+$ support (standard deviations are positive). The larger mean for σ is intended to reflect the presence of a greater variability in population MMSE. For β_1 and β_4, we expect Age and having dementia (Group) to negatively impact MMSE, hence the -1 mean. For β_2 and β_3, we expect SES and EDUC to positively impact MMSE, hence their $+1$ mean.

Diagnostics

Appendix A gives the regression outputs for the first 30 subjects, acting as a sanity check for the model. I will examine trace plots to detect slow mixing and ESS to detect harmful autocorrelation. I will also construct credible intervals using synthetic data to check for model mis-specification.

Broadly speaking, the trace plots and the ESS values are ideal (fast mixing exhibited, most with ESS ≥ 6000, vast majority $\gg 6000$). The $\hat{R}$ values are also strong ($\hat{R} = 1$ consistently). Appendix B illustrates some sample trace plots and the ESS values of the model. There is one parameter worth pointing out. Observe that δ_1 exhibits slower mixing, with the chain struggling more to traverse the posterior space effectively in the trace plot. Also, its ESS is 1246, which is not radical but also considerably lower than the others. However, its $\hat{R} \approx 1$. Estimates for δ_1 are still trustworthy. An explanation could be that δ_1 is highly correlated with the other parameters in the posterior. One easy fix would be to re-parametrize the model so we instead construct α_i by adding $\sigma_1 \times \epsilon$, for $\epsilon \sim \mathcal{N}(0,1)$, to δ_1.

For the calibration check, I performed 250 iterations. In each iteration, I generated true parameters, and from those constructed a synthetic dataset of the same size as the original. The dataset is passed into a STAN model with 3000 iterations, 1000 for warm-up, and 3 chains. This choice is largely due to limited computation power. Still in this iteration, I extract a 95% credible interval for each parameter and check coverage over the true parameters. The code for this can be found in Appendix E. The results are as follows:

```
  delta_1    sigma_1    delta_2    sigma_2      sigma   beta_Age   beta_SES
    0.948      0.820      0.940      0.808      0.000      0.252      0.816
beta_EDUC beta_Group
    0.788      0.952
```

We can observe that the actual coverages for $\delta_1, \delta_2, \beta_{Group}$ are sufficiently close to 95%. With more iterations, we would likely observe convergence to 95% or higher. There is some struggle in $\sigma_1, \sigma_2, \beta_{SES}$, and β_{EDUC}. Due to consistent but not radical undercoverage, we can affirm the presence of mild but non-ignorable model mis-specification in these parameters. There is extreme struggle in σ, β_{Age}. A likely cause for mis-specification in σ_1, σ_2, and, more extremely, σ, is their Exponential priors. In hindsight, it is highly unlikely that these deviations are Exponential in reality. A simple fix would be to choose a more reasonable prior, such as a truncated Gaussian or Half-Cauchy. Due to time and computation constraints, these fixes are not tested in this project. The β's could be explained with the $+1$ or -1 centers in their priors and small standard deviations. These could be too strong and informative (and were rough guesses), taking mass away from the density of the true values. A fix is to adopt weaker priors with larger spread.

Limitations

The key limitation is the scarcity of data. With more data, I could have constructed better informed priors and made more reliable inference for each patient (observe some poor fits in Appendix A). Another is computation power. I would have preferred more iterations for my model and my calibration check. Another limitation is the lack of access to expert advice. I unknowingly induced strong beliefs into my priors that do not well reflect reality. With expert advice, these priors could have been better chosen. Another is the assumption of linearity. The covariates used could have complex non-linear relationships with MMSE that my model is not rich enough to capture. For future work, it would be valuable to consider higher-order regressions, maybe even of a non-parametric nature (i.e. Gaussian processes), informed by expert advice.

Use of AI

In the beginning, I did not know how to identify a set of parameters with a given patient in my STAN model. This is because of the irregular intervals of patients, where some patients have more observations, some fewer. Based on suggestions from AI, I constructed a vector of length the distinct number of patients in the dataset, and was able to hash a patient across several observations to a single set of parameters, namely α_i and γ_i. I also used AI to generate the complex wrap plot of sample regressions in Appendix A, and to generate my citations. Finally, I used AI to debug my code, as I was frequently running into compilation errors that I had never before encountered.

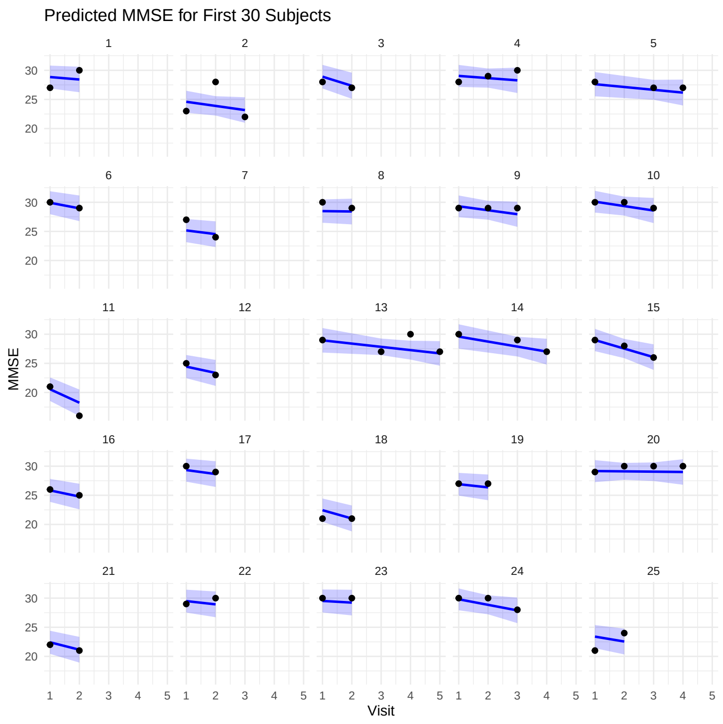
Citations

1. Bartolucci, A., Bae, S., Singh, K., & Griffith, H. R. (2009). An examination of Bayesian statistical approaches to modeling change in cognitive decline in an Alzheimer's disease population. Mathematics and Computers in Simulation, 80(3), 561–571. https://doi.org/10.1016/j.matcom.2009.09.002

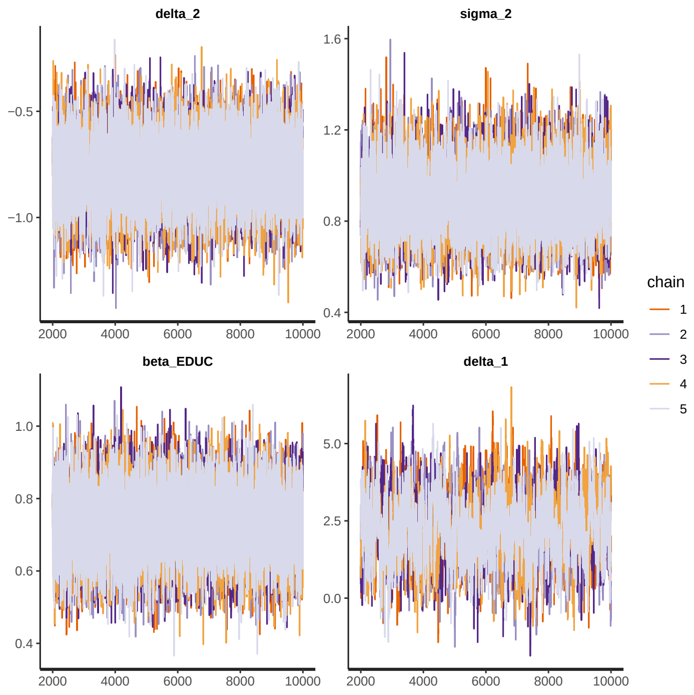
2. Li, H., Benitez, A., & Neelon, B. (2020). A Bayesian hierarchical change point model with parameter constraints. Statistical Methods in Medical Research, 30(1), 316–330. https://doi.org/10.1177/0962280220948097

Appendix

A: Sample regressions



B: Sample trace plots and ESS table



```
              mean se_mean   sd  2.5%   25%   50%   75% 97.5% n_eff Rhat
delta_1       2.27    0.03 1.01  0.30  1.59  2.26  2.95  4.26  1246 1.01
sigma_1       2.53    0.00 0.23  2.09  2.38  2.53  2.69  3.01 10548 1.00
sigma         1.51    0.00 0.08  1.35  1.45  1.50  1.56  1.68 17322 1.00
delta_2      -0.77    0.00 0.15 -1.06 -0.87 -0.77 -0.67 -0.49 12701 1.00
sigma_2       0.89    0.00 0.13  0.64  0.80  0.89  0.98  1.17  6303 1.00
beta_Age      0.18    0.00 0.02  0.13  0.16  0.18  0.19  0.22  7564 1.00
beta_SES      1.34    0.00 0.26  0.83  1.16  1.34  1.51  1.84 12190 1.00
beta_EDUC     0.73    0.00 0.09  0.56  0.67  0.73  0.79  0.90  8280 1.00
beta_Group   -3.00    0.00 0.50 -3.98 -3.33 -3.01 -2.66 -2.00 24412 1.00
```

C: STAN code

```
data {
  int N;
  int S;

  int<lower=1> subject[N];
  vector[N] Age;
  vector[N] SES;
  vector[N] EDUC;
  vector[N] Group;
  vector[N] Time;
  vector[N] MMSE;
}

parameters {
  real delta_1;
  real<lower=0> sigma_1;

  real<lower=0> sigma;

  real delta_2;
  real<lower=0> sigma_2;

  real beta_Age;
  real beta_SES;
  real beta_EDUC;
  real beta_Group;

  vector[S] alpha_i;
  vector[S] gamma_i;
}

model {
  delta_1 ~ normal(0, 1);
  sigma_1 ~ exponential(0.5);

  delta_2 ~ normal(-1, 2);
  sigma_2 ~ exponential(0.5);

  sigma ~ exponential(0.1);

  beta_Age ~ normal(-1, 3);
  beta_SES ~ normal(1, 3);
  beta_EDUC ~ normal(1, 3);
  beta_Group ~ normal(-1, 3);

  alpha_i ~ normal(delta_1, sigma_1);
  gamma_i ~ normal(delta_2, sigma_2);

  for (i in 1:N) {
    real mu = beta_Age * Age[i] +
              beta_SES * SES[i] +
              beta_EDUC * EDUC[i] +
              beta_Group * Group[i] +
              alpha_i[subject[i]] +
              gamma_i[subject[i]] * Time[i];

    MMSE[i] ~ normal(mu, sigma);
  }
}
```

D: R code for model

```
## Change categorical covariates into numerics
data$index = as.integer(as.factor(data$Subject.ID))
data$Group_categorical = ifelse(data$Group == "Demented", 1, 0)

## Impute NAs (do not drop them b/c of limited data)
any(is.na(data$SES))
any(is.na(data$MMSE))
data$SES[is.na(data$SES)] = mean(data$SES, na.rm=TRUE)
data$MMSE[is.na(data$MMSE)] = mean(data$MMSE, na.rm=TRUE)

require(rstan)
fit = stan(
  seed = 123,
  file = "stan_code.stan",
  data = list (
    N = nrow(data),
    S = length(unique(data$Subject.ID)),

    subject = data$index,
    Age = data$Age,
    SES = data$SES,
    EDUC = data$EDUC,
    Group = data$Group_categorical,
    Time = data$Visit,
    MMSE = data$MMSE
    ),
  iter = 10000,
  warmup = 2000,
  chains = 5
)
```

E: R code for calibration check

```
## Check for model mis-spec using simulated data
results = list()
numSubjects = length(unique(data$index))

for (j in 1:250) {
  simulated_data = data.frame(
    subject = integer(),
    time = numeric(),
    Age = numeric(),
    SES = numeric(),
    EDUC = numeric(),
    Group = integer(),
    MMSE = numeric()
  )

  for (i in 1:length(data$index)) {
  delta_1 = rnorm(1,0,1);
  sigma_1 = rexp(1,0.5);

  delta_2 = rnorm(1,-1,2);
  sigma_2 = rexp(1,0.5);

  sigma = rexp(1,0.1);

  beta_Age = rnorm(1,-1,3);
  beta_SES = rnorm(1,1,3);
  beta_EDUC = rnorm(1,1,3);
  beta_Group = rnorm(1,-1,3);

  alpha_i = rnorm(numSubjects, delta_1,sigma_1)
  gamma_i = rnorm(numSubjects, delta_2,sigma_2)
  Age = rnorm(numSubjects, 70, 5)
  SES = rnorm(numSubjects, 0, 3)
  EDUC = rnorm(numSubjects, 15, 3)
```

```
Group = rbinom(numSubjects, 1, 0.5)



visit = data$Visit[i]
mu = alpha_i[data$index[i]] + gamma_i[data$index[i]]*visit +
     beta_Group*Group[data$index[i]] + beta_EDUC*EDUC[data$index[i]] +
     beta_SES*SES[data$index[i]] + beta_Age*Age[data$index[i]]
mmse = rnorm(1, mu, sigma)

simulated_data[nrow(simulated_data) + 1, ] =
  list(subject=data$index[i], time=visit, Age=Age[data$index[i]],
       SES=SES[data$index[i]], EDUC=EDUC[data$index[i]], Group=Group[data$index[i]], MMSE=
}

val=sample(1:200, 1)

fit = stan(
  seed = val,
  file = "stan_code.stan",
  data = list (
    N = nrow(simulated_data),
    S = length(unique(simulated_data$subject)),

    subject = simulated_data$subject,
    Age = simulated_data$Age,
    SES = simulated_data$SES,
    EDUC = simulated_data$EDUC,
    Group = simulated_data$Group,
    Time = simulated_data$time,
    MMSE = simulated_data$MMSE
  ),
  iter = 3000,
  warmup = 1000,
  chains = 3
)

posterior = rstan::extract(fit)
ci = quantile(posterior$delta_1, c(0.025, 0.975))
delta_1_in_ci = (delta_1 >= ci[1]) & (delta_1 <= ci[2])
```

```
  ci = quantile(posterior$sigma_1, c(0.025, 0.975))
  sigma_1_in_ci = (sigma_1 >= ci[1]) & (sigma_1 <= ci[2])
  ci = quantile(posterior$delta_2, c(0.025, 0.975))
  delta_2_in_ci = (delta_2 >= ci[1]) & (delta_2 <= ci[2])
  ci = quantile(posterior$sigma_2, c(0.025, 0.975))
  sigma_2_in_ci = (sigma_2 >= ci[1]) & (sigma_2 <= ci[2])
  ci = quantile(posterior$sigma, c(0.025, 0.975))
  sigma_in_ci = (sigma >= ci[1]) & (sigma <= ci[2])

  ci = quantile(posterior$beta_Age, c(0.025, 0.975))
  beta_Age_in_ci = (beta_Age >= ci[1]) & (beta_Age <= ci[2])
  ci = quantile(posterior$beta_SES, c(0.025, 0.975))
  beta_SES_in_ci = (beta_SES >= ci[1]) & (beta_SES <= ci[2])
  ci = quantile(posterior$beta_EDUC, c(0.025, 0.975))
  beta_EDUC_in_ci = (beta_EDUC >= ci[1]) & (beta_EDUC <= ci[2])
  ci = quantile(posterior$beta_Group, c(0.025, 0.975))
  beta_Group_in_ci = (beta_Group >= ci[1]) & (beta_Group <= ci[2])

  coverage_result = list(
    delta_1 = delta_1_in_ci,
    sigma_1 = sigma_1_in_ci,
    delta_2 = delta_2_in_ci,
    sigma_2 = sigma_2_in_ci,
    sigma = sigma_in_ci,
    beta_Age = beta_Age_in_ci,
    beta_SES = beta_SES_in_ci,
    beta_EDUC = beta_EDUC_in_ci,
    beta_Group = beta_Group_in_ci
  )

  results[[j]] = coverage_result
}

sapply(results, function(x) unlist(x)) |>
  as.data.frame() |>
  rowMeans()
```