Chain Beta-Binomial model

Soham Ghosh

2024-12-03

The Chain Beta-Binomial model

Consider a dataset with N observations, where y_i is the number of secondary infections for observation i, and n_i is the number of contacts for observation i. Let age_i be the age of the primary patient for observation i.

- **Model Structure**: The number of secondary infections y_i follows a beta-binomial distribution with parameters α_i and β_i.
- **Priors**: Priors are specified for the intercept α, the coefficients β_{age} and $\beta_{\text{vaccination}}$, and the overdispersion parameter ϕ.
- **Transformed Parameters**: The parameter μ_i is modeled on the logit scale as a function of age and vaccination status. The parameters α_i and β_i are derived from μ_i and ϕ.
- **Likelihood**: The likelihood function describes how the observed data y_i is generated from the beta-binomial distribution.

Model Structure

$$
\begin{aligned}
y_i &\sim \text{Beta-Binomial}(n_i, \alpha_i, \beta_i) \\
\text{logit}(\mu_i) &= \alpha + \beta_{\text{age}} \cdot \text{age}_i + \beta_{\text{vaccination}} \cdot \text{vaccination}_i \\
\alpha_i &= \mu_i \cdot \phi \\
\beta_i &= (1 - \mu_i) \cdot \phi
\end{aligned}
$$

Priors

$$
\begin{aligned}
\alpha &\sim \mathcal{N}(0, 10) \\
\beta_{\text{age}} &\sim \mathcal{N}(0, 10) \\
\beta_{\text{vaccination}} &\sim \mathcal{N}(0, 10) \\
\phi &\sim \text{Cauchy}(0, 5)
\end{aligned}
$$

Transformed Parameters

$$
\begin{aligned}
\mu_i &= \text{logit}^{-1}(\alpha + \beta_{\text{age}} \cdot \text{age}_i + \beta_{\text{vaccination}} \cdot \text{vaccination}_i) \\
\alpha_i &= \mu_i \cdot \phi \\
\beta_i &= (1 - \mu_i) \cdot \phi
\end{aligned}
$$

Likelihood

$$
y_i \sim \text{Beta-Binomial}(n_i, \alpha_i, \beta_i)
$$

Hierarchical Model

Combining the above components, the full hierarchical model is:

$$
\begin{aligned}
y_i &\sim \text{Beta-Binomial}(n_i, \alpha_i, \beta_i) \\
\text{logit}(\mu_i) &= \alpha + \beta_{\text{age}} \cdot \text{age}_i + \beta_{\text{vaccination}} \cdot \text{vaccination}_i \\
\alpha_i &= \mu_i \cdot \phi \\
\beta_i &= (1 - \mu_i) \cdot \phi \\
\alpha &\sim \mathcal{N}(0, 10) \\
\beta_{\text{age}} &\sim \mathcal{N}(0, 10) \\
\beta_{\text{vaccination}} &\sim \mathcal{N}(0, 10) \\
\phi &\sim \text{Cauchy}(0, 5)
\end{aligned}
$$

```
stan_model_code <- "
data {
  int<lower=0> N;  // number of observations
  int<lower=0> y[N];  // number of secondary infections
  int<lower=0> n[N];  // number of contacts
  vector[N] age;  // age of the primary patient
  int vaccination_status[N];  // vaccination status of the primary patient
}
parameters {
  real alpha;  // intercept
  real beta_age;  // coefficient for age
  real beta_vaccination;  // coefficient for vaccination status
  real<lower=0> phi;  // overdispersion parameter
}
transformed parameters {
  vector[N] mu;  // logit scale of the mean of the beta distribution
  vector[N] alpha_minus_mu;  // auxiliary variable for mean calculation
  vector<lower=0>[N] alpha_param;  // shape parameter of beta distribution
  vector<lower=0>[N] beta_param;  // shape parameter of beta distribution

  for (i in 1:N) {
    mu[i] = inv_logit(alpha + beta_age * age[i] + beta_vaccination * vaccination_status[i]);
    alpha_minus_mu[i] = mu[i] * phi;
    alpha_param[i] = alpha_minus_mu[i];
    beta_param[i] = (1 - mu[i]) * phi;
  }
}
model {
  // Priors
  alpha ~ normal(0, 10);
  beta_age ~ normal(0, 10);
  beta_vaccination ~ normal(0, 10);
  phi ~ cauchy(0, 5);

  // Likelihood
  for (i in 1:N) {
    y[i] ~ beta_binomial(n[i], alpha_param[i], beta_param[i]);
  }
```

```
}
"
# Example data
data <- data.frame(
  y = c(2, 1, 3, 0, 1),
  n = c(10, 8, 12, 6, 9),
  age = c(30, 45, 25, 60, 35),
  vaccination_status = c(1, 0, 1, 0, 1)
)
stan_data <- list(
  N = nrow(data),
  y = data$y,
  n = data$n,
  age = data$age,
  vaccination_status = data$vaccination_status
)

# Fit the model
fit <- stan(
  model_code = stan_model_code,
  data = stan_data,
  iter = 2000,
  chains = 1,
  warmup = 1000,
  thin = 1,
  seed = 123
)
```

Estimating SAR

We can compute the Secondary Attack Risk, along with a 95% credible interval extracted from the posterior samples.

```
# Extract posterior samples
posterior_samples <- rstan::extract(fit)

# Compute mu for each observation
compute_mu <- function(alpha, beta_age, beta_vaccination, age, vaccination_status) {
  invlogit(alpha + beta_age * age + beta_vaccination * vaccination_status)
}

# Apply the function to each sample to get the distribution of mu for each observation
mu_samples <- sapply(1:nrow(data), function(i) {
  compute_mu(
    posterior_samples$alpha,
    posterior_samples$beta_age,
    posterior_samples$beta_vaccination,
    data$age[i],
    data$vaccination_status[i]
  )
})

# # Average mu across all observations and samples to get the overall SAR
# sar_samples <- rowMeans(mu_samples)
```

```
#
# # Summarize SAR
# sar_mean <- mean(sar_samples)
# sar_cred_int <- quantile(sar_samples, probs = c(0.025, 0.975))
#
# cat("Secondary Attack Risk (SAR):\n")
# cat("Mean:", sar_mean, "\n")
# cat("95% Credible Interval:", sar_cred_int, "\n")
sar_samples <- rowMeans(mu_samples)

# Summarize SAR
sar_mean <- mean(sar_samples)
sar_cred_int <- quantile(sar_samples, probs = c(0.025, 0.975))

cat("Secondary Attack Risk (SAR):\n")
```

```
## Secondary Attack Risk (SAR):
```

```
cat("Mean:", sar_mean, "\n")
```

```
## Mean: 0.0001302313
```

```
cat("95% Credible Interval:", sar_cred_int, "\n")
```

```
## 95% Credible Interval: 1.600167e-05 0.0003693348
```

```
# Credible intervals for vaccination status effect
beta_vaccination_samples <- posterior_samples$beta_vaccination
beta_vaccination_cred_int <- quantile(beta_vaccination_samples, probs = c(0.025, 0.975))

cat("Effect of Vaccination Status (Beta_vaccination):\n")
```

```
## Effect of Vaccination Status (Beta_vaccination):
```

```
cat("Mean:\n", mean(beta_vaccination_samples))
```

```
## Mean:
##  1.921052
```

```
cat("Credible Interval:", beta_vaccination_cred_int)
```

```
## Credible Interval: 0.8302972 2.535033
```

Computing Secondary cases (ORCHARDS data example)

To prepare the ORCHARDS data for the Stan model, we compute the following inputs for each household or observation:

$$
\begin{aligned}
y[i] &= \max\left(\text{Total Positive Cases on Day 14} - \text{Primary Cases on Day 0}, 0\right), \\
n[i] &= \text{Number in Household} - 1, \\
\text{age}[i] &= \text{Age of Primary Case}, \\
\text{vaccination\_status}[i] &= \text{Vaccination Status of Primary Case}.
\end{aligned}
$$

1. **Number of Secondary Cases ($y[i]$):**
   - For each household i, determine the total number of positive cases on Day 14 and subtract the number of primary cases (positive on Day 0).

- Ensure $y[i]$ is non-negative:

$$y[i] = \max\left(\text{Total Positive Cases on Day 14} - \text{Primary Cases on Day 0}, 0\right).$$

2. **Number of Contacts ($n[i]$)**:
   - Use the total number of household members and subtract 1 to exclude the primary case:

$$n[i] = \text{Number in Household} - 1.$$

```
data <- read.csv("~/Downloads/ORCHARDS2020-20222023HouseholdStu_DATA_2024-05-23_0936.csv")
processed_data <- data %>%
  mutate(
    is_primary_case = ifelse(student_covid_day0 == 1, 1, 0),
    is_secondary_case_day7 = ifelse(student_covid_day7 == 1, 1, 0),
    is_secondary_case_day14 = ifelse(student_covid_day14 == 1, 1, 0)
  ) %>%
  group_by(record_id) %>%
  summarise(
    primary_cases = sum(is_primary_case, na.rm = TRUE),
    secondary_cases_day7 = sum(is_secondary_case_day7, na.rm = TRUE),
    secondary_cases_day14 = sum(is_secondary_case_day14, na.rm = TRUE),
    number_in_household = first(number_in_household),
    age = first(age)) %>%
  mutate(
    y = pmax(secondary_cases_day14 - primary_cases, 0),  # Use Day 14 for total positives
    n = number_in_household - 1                          # Contacts excluding primary case
  )
```