Online Appendix to Methodology for the Generation of Normative Data for the U.S. Adult Spanish-Speaking Population: A Bayesian Approach

Diego Rivera, Anabel Forte, Laiene Olabarrieta-Landa, Paul B. Perrin, Juan Carlos Arango-Lasprilla

Contents

Abstract

This Web Appendix aims to briefly illustrate the script of implementing the Bayesian generalized linear model (BGLM) for the generation of normative data in neuropsychological tests. The theoretical aspects can be found in Rivera et al. (2024) and its implementation was carried out using R software. The Bayesian inference procedure was performed using Markov Chain Monte Carlo (MCMC) methods.

This appendix is organized into four sections. In Section 1, the covariates in the model and necessary libraries are presented. In Section 2, Bayesian variable selection is conducted. In Section 3, the BGLM regression-based approach is performed, and in Section 4, the normative procedure is presented.

1. Covariates in the Model and Necessary Libraries

The main model comprised seven demographic variables and two-level interactions:

- `agec:` The age (in years) centered of the participant $(Age_i - \bar{X}_{\text{Age}})$.
- `agec2:` Quadratic effect for age centered $(Age_i - \bar{X}_{\text{Age}})^2$.
- `education:` Years of education for participants.
- `sex:` The sex of the participants.
- `timeusa:` The time living in U.S. for participants.
- `bashisp:` Bidimensional Acculturation Scale score for participants.
- `bds:` Bilingual Dominance Scale score for participants.

Regarding the `R libraries` needed to conduct the data analysis, it is necessary to install and load the following:

```
if (!requireNamespace("rjags", quietly = TRUE)) {
  install.packages("rjags")
}

if (!requireNamespace("BAS", quietly = TRUE)) {
  install.packages("BAS")
}

library(rjags)
source("DBDA2E-utilities.R")
library(BAS)
```

2. Bayesian Variable selection

Main model definition

The main model for each of the neuropsychological scores (Y) follows a GLM structure:

$$g(\mu_i) = \mathbf{X}_i\boldsymbol{\beta} = \eta_i,$$

where $\mu_i \equiv \mathbb{E}(Y_i)$, g is a smooth monotonic *link function*, $\mathbf{X}_i$ is the i-th row of a model matrix, $\mathbf{X}$, and $\boldsymbol{\beta}$ is a vector of unknown parameters. In addition, a GLM usually makes the distributional assumptions that the Y_i are independent and $(Y_i \sim)$ some exponential family distribution. In this case η_i is composed by:

$$\eta_i = \beta_0 + \beta_1 \cdot \text{age} + \beta_2 \cdot \text{age}^2 + \beta_3 \cdot \log(\text{education}) + \beta_4 \cdot \text{sex} + \beta_5 \cdot \text{time} + \beta_6 \cdot \text{bas} + \beta_7 \cdot \text{bds} + \beta_k \cdot interacions_k$$

As an example, Y will be the number of words generated for the R (`R`), and therefore the main model will be defined as follows in the R script:

```
main_model <- R ~ (agec + agec2 + log(education) + sex + timeusa + bashisp + bds)^2
```

Posterior inclusion probabilities (PIPs) estimation

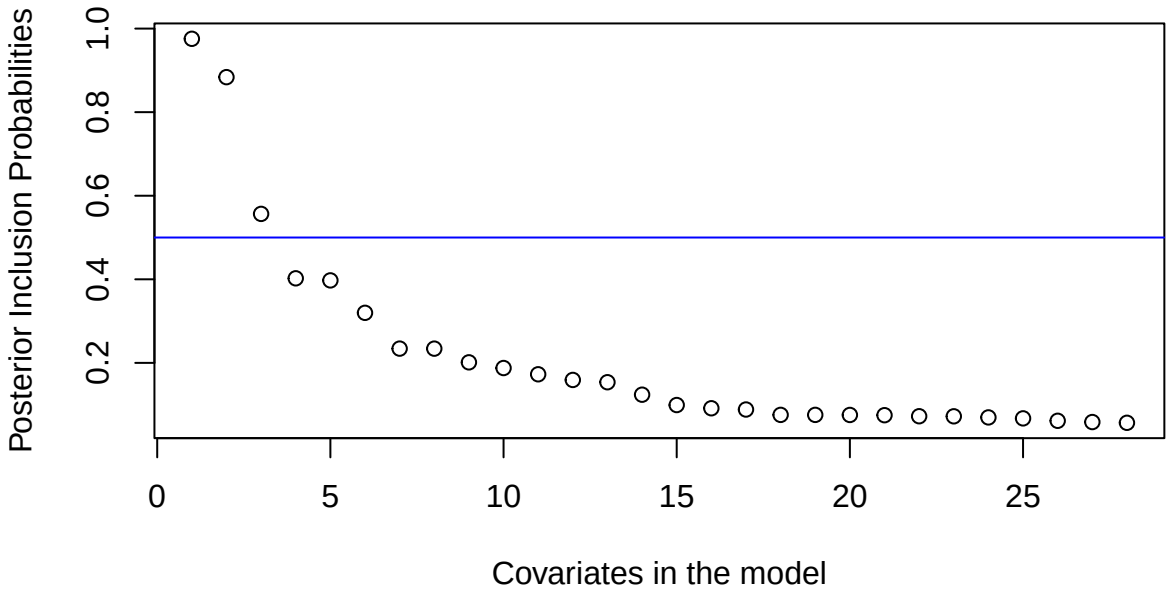
The `BAS` library was used for PIPs estimation based on the main model for each of the neuropsychological scores (Y):

```
fit.bas_r <- bas.glm(
  main_model,
  family = poisson(),
  data = df
)
```

Using the `summary` function, the user can extract the PIPs of each covariate by looking at the column `P(B ! = 0 | Y)`. Once the PIPs are obtained, an elbow plot can be constructed to observe the behavior pattern of the PIPs. For this example (phoneme R) it can be observed that there are three covariates with a PIP greater than 0.5 (blue line). These covariates are:

- sexWoman:bds = 0.97,
- log(education) = 0.88,
- agec:log(education) = 0.55,

determining which variables should be included in the final model for parameter estimation (see next Figure).



3. Model for phoneme R

In this section the reader will find the R script used for parameter estimation based on Bayesian inference. In our example, a Poisson regression will be used. The model includes the variables suggested in the previous section. The reader should remember that in the case of interaction, the first level covariates should be included in the model.

```
modelString <- "
model{
    for(i in 1:n){
    y[i] ~ dpois(lambda[i])
    log(lambda[i]) <- beta[1] + beta[2] * agec[i] + beta[3] * education[i] +
                      beta[4] * sex[i] + beta[5] * bds[i] +
                      beta[6] * agec[i] * education[i] +
                      beta[7] * sex[i] * bds[i]
  }

  # Priori distribution
  for (j in 1:7){
    beta[j] ~ dnorm(0.0, 0.0001)
  }
}"

# Save the model as a text file
writeLines(modelString, con = "model_r.txt")

# Function to generate initial values
init_values <- function() {
  list(beta = rnorm(7, 0, 0.01))
}
init_values()

# Data for JAGS
jags_data <- list(
  y         = df$R,
  agec      = df$agec,
  education = log(df$education),
  sex       = as.numeric(df$sex == "Woman"),
  bds       = df$bds,
  n         = length(df$R)
)

# Initialize the model in JAGS
jagsModel <- jags.model(
  file     = "model_r.txt",
  data     = jags_data,
  inits    = init_values,
  n.chains = 3,
  n.adapt  = 100000
)

# Burn-in period
update(jagsModel, n.iter = 2000)

# Simulation
codaSamples_r <- coda.samples(
  jagsModel,
  variable.names = c("beta"),
  n.iter         = 100000,
  thin           = 100
)

# Summary of coda samples
summary(codaSamples_r)
```

Summary model

Using (`DBDA2E-utilities.R`) the reader can review the performance of the samples based in MCMC thought credibility intervals and plots of posterior distribution for each parameter.

```
diagMCMC(codaObject = codaSamples_r, parName = "beta[3]")

plotPost(
  codaSamples_r[,"beta[3]"],
  xlab     = "r_beta[3]",
  cenTend  = "median",
  credMass = 0.95
)
```

4. Normative data estimation. An example

To facilitate the understanding of the procedure to obtain the percentile associated with a given score on this test, an example will be given. Suppose you need to find the probability for a woman, who is 50 years old and has 15 years of education. She obtained a BDS score of 10 and a score of 9 on the /r/ phoneme.

```
# Define the participant's predictors
age_participant <- 50 - 41.3
edu_participant <- log(15)
sex_participant <- "Woman"
bds_participant <- 10
score <- 9

# Extract MCMC samples for the parameters
Intercept <- unlist(codaSamples_r[, "beta[1]"])
age_parameter <- unlist(codaSamples_r[, "beta[2]"])
edu_parameter <- unlist(codaSamples_r[, "beta[3]"])
sex_parameter <- unlist(codaSamples_r[, "beta[4]"])
bds_parameter <- unlist(codaSamples_r[, "beta[5]"])
age_edu_parameter <- unlist(codaSamples_r[, "beta[6]"])
sex_bds_parameter <- unlist(codaSamples_r[, "beta[7]"])

# Convert sex to numeric
sex_numeric <- ifelse(sex_participant == "Woman", 1, 0)

# Calculate lambda for each sample
lambda_R <- exp(
  Intercept +
  age_parameter * age_participant +
  edu_parameter * edu_participant +
  sex_parameter * sex_numeric +
  bds_parameter * bds_participant +
  age_edu_parameter * (age_participant * edu_participant) +
  sex_bds_parameter * (sex_numeric * bds_participant)
)

# Calculate the posterior probability for the score
p_lambda_R <- numeric(length(lambda_R))
for (i in 1:length(lambda_R)) {
  p_lambda_R[i] <- ppois(score, lambda_R[i])
}

# Calculate the mean of the posterior probabilities
mean(p_lambda_R) *100
```

The above procedure is the basis of the calculator.