Multinomial logistic regression analysis for Experiment 1 [acoustic data]

This analysis provided estimates for the probability that the vowels in the 56 auditory syllable stimuli would be classified as the vowels intended by the speakers based on vowel duration and the first three formants (F1, F2, F3). The analysis comprised two parts. In Part 1, a multinomial logistic regression model was trained on a small corpus of S.Eng vowel data collected from 10 young female speakers (Williams and Escudero, 2014a). There were 16 vowel categories in the corpus, namely, CHOICE, DRESS, FACE, FLEECE, FOOT, GOAT, GOOSE, KIT, LOT, MOUTH, NURSE, PALM, PRICE, STRUT, THOUGHT and TRAP, and vowel tokens were produced in /bVp/, /dVt/, /fVf/, /gVk/ and /sVs/ syllable frames. Part 2 involved obtaining probability estimates that the vowels in the 56 auditory syllable stimuli would be classified as the intended vowel category on the basis of the model trained on the S.Eng corpus.

Logistic regression can sometimes result in quasi or complete separation, i.e., the model's predictors predict at least one of the categorical outcomes perfectly. This situation is undesirable because it may lead to computational problems (non-identifiability) and, consequently, sensible estimates may not be possible. A practical solution is to rescale the predictors and set weakly informative priors. Gelman et al. (2008) propose (1) rescaling continuous predictors to exhibit a mean of 0 and standard deviation of 0.5, (2) using Cauchy priors with center 0 and scale 2.5 for predictors and (3) using Cauchy priors with center 0 and scale 10 for intercept terms.

In Part 1, the multinomial logistic regression was run using the package *brms* (Bürkner, 2017; Bürkner, 2018) in the program *R (*R Core Team, 2021*)*. The code used to fit the model is provided below. Ten acoustic predictors were entered, namely, rescaled versions (as described above) of log duration (in milliseconds) and three parameters describing the trajectories of the first three formants (F1, F2 and F3 frequencies in Bark). The three formant trajectory parameters were the 0^{th}, 1^{st} and 2^{nd} discrete cosine transform (DTC) coefficients which correspond to a formant's mean frequency, its slope and curvature, respectively (for further details, see Williams and Escudero, 2014a). The dependent variable was vowel category and comprised 16 discrete levels (one for each phonemic vowel category). For Part 2, acoustic information on the vowels in the 56 auditory syllable stimuli was tested on the model trained on the S.Eng corpus in order to obtain predicted probabilities.

Experiment 1 multinomial logistic regression code

```
## Part 1: Training a model on a S.Eng vowel corpus using 10 rescaled (RS)
## acoustic predictors

## Load S.Eng training data [Acoustic_data_for_S.Eng_training.csv]
S.Eng = read.csv(file.choose())

## Load N.Eng & S.Eng stimulus test data [Acoustic_data_for_S.Eng_test.csv]
N.Eng = read.csv(file.choose())

## Load brms
library(brms)

## Specify model formula
formula_S.Eng = bf(vowel_name ~ duration_logRS +
                                F1_DCT0RS + F1_DCT1RS + F1_DCT2RS +
                                F2_DCT0RS + F2_DCT1RS + F2_DCT2RS +
                                F3_DCT0RS + F3_DCT1RS + F3_DCT2RS,
                              family = "categorical")

prior =       c(prior("cauchy(0, 10)", class = "Intercept"),
              set_prior("cauchy(0, 2.5)", class = "b") )

## Run the training model [may take a long time]
training_S.Eng = brm(formula = formula_S.Eng, data = S.Eng,
                prior = prior,
```

```
                 chains = 4, cores = 4, warmup = 1000, iter = 1000 + 3000,
                 control = list(adapt_delta = 0.99, max_treedepth = 13))

## Did the model classify the corpus's vowel tokens accurately? [may take some
## time]
trained_S.Eng = fitted(training_S.Eng)

## Looking at the obtained predicted probabilities in the trained_S.Eng object,
## the vowel tokens in the corpus received on average a predicted probability of
## being correctly classified as the intended vowel category of 0.94 (SD =
## 0.12), indicating they were well separated based on the 10 (rescaled)
## acoustic measures.

## Part 2: Testing the auditory stimulus data on the trained S.Eng model
test_S.Eng = fitted(training_S.Eng, newdata = N.Eng)

## A summary of the results in the test_S.Eng object is displayed in Appendix A
## in the manuscript.
```

Probit regression analysis for Experiment 1 [phoneme detection data]

A probit regression model was run using the package *brms* (Bürkner, 2017; Bürkner, 2018) in the program *R (*R Core Team, 2021*)*. The code used to fit and summarize the model are provided below. Weak priors were set. Four chains sampled the posterior distribution adequately based on there being no divergent transitions and visual inspection of the trace plots, which showed good mixing of chains for each parameter. Additionally, the effective sample size for every parameter was at least 100 times the number of chains, and all R̂s ("Rhats") were less than 1.01 (Stan Development Team, 2022). To summarize the posterior distribution, the *bayestestR* package was used (Makowski et al., 2019), which provided obtain medians and 89% credible intervals (CIs). Rather than the conventional 95% intervals, we opted for 89% intervals because this is deemed to provide more stable estimates when the effective sample size is less than 10,000 samples (Kruschke, 2014), which was the case for all population-level effects parameters (see the pdf of model output).

```
                    ### Experiment 1 probit regression code ###
## Load trials from Experiment 1 phoneme detection task [Exp1_trials.csv]
Exp1 = read.csv(file.choose())

## Code ItemType and Similarity predictors
Exp1$ItemType = factor(Exp1$ItemType,
                           levels = c("Mismatching", "Matching"))

contrasts(Exp1$ItemType) = c(-0.5,0.5)

Exp1$Similarity = factor(Exp1$Similarity, levels = c("S.Eng",
                                  "Similar", "Dissimilar"))

diff.cd = matrix(c(-2/3, 1/3, 1/3,
                    -1/3, -1/3, 2/3), ncol = 2)

contrasts(Exp1$Similarity) = diff.cd

## Load brms
library(brms)

## Specify model formula
Exp1.formula = bf(Sameness ~ ItemType * Similarity +
                           (1 + ItemType * Similarity | Participant) +
                           (1 | Item),
                           family = bernoulli(link="probit"))

Exp1.priors = c(prior("student_t(3, 0, 3)", class = "Intercept"),
                    prior("student_t(3, 0, 3)", class = "b"),
```

```
                    prior("student_t(3, 0, 3)", class = "sd"),
                    prior("lkj_corr_cholesky(1)", class = "L"))

## Run model [may take a long time]
Exp1.model = brm(Exp1.formula,
                    prior = Exp1.priors,
                    data = Exp1,
                    chains = 4, cores = 4, warmup = 1000,
                    iter = 1000 + 3000,
                    control = list(adapt_delta = 0.95, max_treedepth = 12))


## Summary of population-level effects using the bayestestR package
library(bayestestR)

describe_posterior(Exp1.model, ci = 0.89)
```

Multinomial logistic regression analysis for Experiment 2 [acoustic data]

The multinomial logistic regression was identical to Part 1 except the training dataset was a small corpus of 12 young female Aus.Eng speakers (Elvin et al., 2016) and the test data in Part 2 comprised acoustic information from the 56 auditory stimuli used to create the items for Experiment 2.

```
             ### Experiment 2 multinomial logistic regression code ###
## Part 1: Training a model on the Aus.Eng vowel corpus using 10 rescaled (RS)
## acoustic predictors

## Load Aus.Eng training data [Acoustic_data_for_AusE_training.csv]
AusE = read.csv(file.choose())

## Load N.Eng & Aus.Eng stimulus test data [Acoustic_data_for_AusE_test.csv]
N.Eng = read.csv(file.choose())

## Load brms
library(brms)

## Specify model formula
formula_AusE = bf(vowel_name   ~ duration_logRS +
                                 F1_DCT0RS + F1_DCT1RS + F1_DCT2RS +
                                 F2_DCT0RS + F2_DCT1RS + F2_DCT2RS +
                                 F3_DCT0RS + F3_DCT1RS + F3_DCT2RS,
                               family = "categorical")

prior =      c(prior("cauchy(0, 10)", class = "Intercept"),
             set_prior("cauchy(0, 2.5)", class = "b") )

## Run training model [may take a long time]
training_AusE = brm(formula = formula_AusE, data = AusE,
               prior = prior,
               chains = 4, cores = 4, warmup = 1000, iter = 1000 + 3000,
               control = list(adapt_delta = 0.99, max_treedepth = 13))

## Did the model classify the corpus's vowel tokens accurately? [may take some
## time]
trained_AusE = fitted(training_AusE)

## Vowel tokens received an average predicted probability of being correctly
## classified as the intended vowel category of 0.84 (SD = 0.22), indicating they
## were generally well separated based on the chosen 10 (rescaled) acoustic
## measures.

## Part 2: Testing the auditory stimulus data on the trained Aus.Eng model
test_AusE = fitted(training_AusE, newdata = N.Eng)

## A summary of the results from this part is displayed in Appendix B in the
## manuscript.
```

Probit regression analysis for Experiment 2 [phoneme detection data]

The model fitting procedure was identical to that in Experiment 1 except that there an additional predictor for Group (Monolingual, Bilingual).

```
### Experiment 2 probit regression code ###
## Load trials from Experiment 2 phoneme detection task [Exp2_trials.csv]
Exp2 = read.csv(file.choose())

## Code ItemType, Similarity and Group predictors
Exp2$ItemType = factor(Exp2$ItemType,
                           levels = c("Mismatching", "Matching"))

contrasts(Exp2$ItemType) = c(-0.5,0.5)

Exp2$Similarity = factor(Exp2$Similarity, levels = c("Similar",
                                                        "Dissimilar"))
contrasts(Exp2$Similarity) = c(-0.5,0.5)

Exp2$Group = factor(Exp2$Group, levels = c("Monolingual", "Bilingual"))
contrasts(Exp2$Group) = c(-0.5,0.5)

## Load brms
library(brms)

## Specify model formula
Exp2.formula = bf(Sameness ~ ItemType * Similarity * Group +
                           (1 + ItemType * Similarity | Participant) +
                           (1 + Group | Item),
                           family = bernoulli(link="probit"))

Exp2.priors = c(prior("student_t(3, 0, 3)", class = "Intercept"),
                    prior("student_t(3, 0, 3)", class = "b"),
                    prior("student_t(3, 0, 3)", class = "sd"),
                    prior("lkj_corr_cholesky(1)", class = "L"))

## Run model [may take a long time]
Exp2.model = brm(Exp2.formula,
                    prior = Exp2.priors,
                    data = Exp2,
                    chains = 4, cores = 4, warmup = 1000,
                    iter = 1000 + 3000,
                    control = list(adapt_delta = 0.95, max_treedepth = 12))

## Summary of population-level effects using the bayestestR package
library(bayestestR)

describe_posterior(Exp2.model, ci = 0.89)
```