An Introduction to Bayesian Multilevel Models Using brms: A Case Study of Gender Effects on Vowel Variability in Standard Indonesian

Supplementary Materials

Contents

Moderation Analysis .......................... 1
   Interpreting the output ........................................ 2
   Visualising predictions ............................................... 4

Lognormal and Skew-Normal models ........... 5
   Model comparison .................................................... 6
   Posterior predictive checking ..................................... 7

Session information ........................... 8

References ........................................... 9

Moderation Analysis

In the current paper, we were interested in knowing whether men and women differ in the way they pronounce Indonesian vowels. Acknowledging that gender differences might be more similar across repetitions of the same vowel, we allowed the mean variability (and the difference between men and women) to vary by vowel (by including a by-vowel varying intercept and varying slope). This strategy aimed at increasing the precision of our estimation of the gender effect. But it does not allow us to know whether the effect of gender differs according to the vowel and which vowels are more affected by gender. To answer this question, one might add vowel as a constant factor in the model.

This model corresponds to bmod2 from the main manuscript, to which we added a constant effect for vowel as well as an interaction term gender:vowel. In R, vowel will be internally recoded as a factor, resulting in attributing a specific coefficient by vowel (and for each interaction gender:vowel). More precisely, the intercept will be assigned to the first level of the factor, and slopes will be assigned to the other levels.

In the current situation, vowels are alphabetically ordered (/a-e-i-o-u/) so that the intercept represents the mean predicted value of distance for vowel /a/ for all participants (without distinction of gender). This model can be implemented using brms as follows.
library(brms)

prior <- c(
  prior(normal(0, 10), class = Intercept),
  prior(normal(0, 10), class = b),
  prior(cauchy(0, 10), class = sd),
  prior(cauchy(0, 10), class = sigma)
)

bmod <- brm(
  distance ~ gender * vowel + (1|subj),
  data = indo, family = gaussian(),
  prior = prior,
  warmup = 2000, iter = 10000,
  chains = 2, cores = parallel::detectCores(),
  control = list(adapt_delta = 0.99)
)

Interpreting the output

We can retrieve the coefficients for the constant effects using the broom::tidy() function, which gives the mean of the posterior distribution along with its standard error and credible interval.

library(broom)
tidy(bmod, par_type = "non-varying", prob = 0.95)

We mentioned above that the intercept represents the mean predicted value of distance for vowel /a/ for all participants (which is indeed very similar to mean(indo$distance[indo$vowel=="/a/"]).

The slope for gender represents the predicted difference between men and women concerning the
distance for vowel /a/. As gender was contrast coded, the intercept + 0.5 times the slope for gender gives the model predictions for men concerning vowel /a/, while the intercept + (-0.5) times the slope for gender gives the model prediction for women concerning vowel /a/. One can then study the marginal posterior distribution of each condition, working directly with the posterior samples.

```r
library(BEST)

# extracting posterior samples
post <- posterior_samples(bmod, pars = "b_*")

# extracting predicted values for men
men <- post["b_Intercept"] + 0.5 * post["b_gender"]

# plotting the posterior distribution of men
plotPost(men, xlab = "", col = "#b3cde0", cex = 1)
```

Following the same strategy, one can investigate any comparison of interest. For instance, if we are interested in the difference between women and men concerning a specific vowel –say vowel /i/–, we can compute the difference between the model predictions for men and women, and plot this distribution. We simply need to know that interaction terms (e.g., gender:vowelDiD) represent deviations from the baseline slope for gender. Thus, we need to add them to obtain the effect of gender for a specific vowel.

```r
# computing the difference between men and women for vowel /i/
diff_i <- post["b_gender"] + post["b_gender:vowelDiD"]

# plotting it
```
Visualising predictions

One of the easiest ways to understand the output of the model is probably to plot its predictions. Below we plot the raw data along with the predictions of the model, represented by a horizontal line.
This is consistent with Figure 1 from the main manuscript. While women generally produce vowels with more variability, this gender effect seems more pronounced for open or mid vowels /a-e-o/, and less pronounced for close vowels /u/ and /i/.

**Lognormal and Skew-Normal models**

In the following, we fit a model with the same structure as bmod5 (the last model from the main manuscript), except that we use a different likelihood (i.e., distribution for the residuals). In this case, we fit a lognormal and a skew normal models. Note that the only modification we bring to bmod5 is that we change the family argument in the brm function, replacing gaussian() by lognormal() or skew_normal() (but see the brms package documentation for more details).

```r
priors <- c(  prior(normal(0, 10), class = Intercept),  prior(normal(0, 10), class = b, coef = gender),  prior(cauchy(0, 10), class = sd),  prior(cauchy(0, 10), class = sigma),  prior(lkj(2), class = cor) )
```
bmod5 <- brm(
  distance ~ gender + (1|subj) + (1|gender:vowel) + (1|subj:vowel),
  data = indo,
  family = gaussian(),
  prior = priors,
  warmup = 2000, iter = 10000,
  chains = 2, cores = parallel::detectCores(),
  control = list(adapt_delta = 0.99)
)

bmod6 <- brm(
  distance ~ gender + (1|subj) + (1|gender:vowel) + (1|subj:vowel),
  data = indo,
  family = lognormal(),
  prior = priors,
  warmup = 2000, iter = 1e4,
  chains = 2, cores = parallel::detectCores(),
  control = list(adapt_delta = 0.99)
)

bmod7 <- brm(
  distance ~ gender + (1|subj) + (1|gender:vowel) + (1|subj:vowel),
  data = indo,
  family = skew_normal(),
  prior = priors,
  warmup = 2000, iter = 1e4,
  chains = 2, cores = parallel::detectCores(),
  control = list(adapt_delta = 0.99)
)

Model comparison

Once we have fitted these models, we can compare them using the LOO function (see section 3 on model comparison).

```
LOO(bmod5, bmod6, bmod7, cores = parallel::detectCores() )
```

<table>
<thead>
<tr>
<th></th>
<th>LOOIC</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>bmod5</td>
<td>-3591.14</td>
<td>68.13</td>
</tr>
<tr>
<td>bmod6</td>
<td>-3697.56</td>
<td>73.42</td>
</tr>
<tr>
<td>bmod7</td>
<td>-3690.61</td>
<td>64.56</td>
</tr>
<tr>
<td>bmod5 - bmod6</td>
<td>106.42</td>
<td>60.15</td>
</tr>
</tbody>
</table>
This comparison reveals that replacing the Normal likelihood by a lognormal or a skew-normal one, while adding a supplementary parameter to estimate (the alpha parameter for the skew-normal), induced a considerable decrease in LOOIC value (\texttt{bmod6} and \texttt{bmod7} LOOIC values are approximately 100 points below the LOOIC value of \texttt{bmod5}), although being associated with quite a large uncertainty (as expressed by the SE).

### Posterior predictive checking

Another useful diagnostic of the model’s predictive abilities is known as \textit{posterior predictive checking} (PPC) and consists in comparing observed data to data simulated from the posterior distribution (e.g., Gelman et al., 2013). The idea behind PPC is quite simple: if a model is a good fit, then we should be able to use it to generate data that resemble the data we observed (Gabry, Simpson, Betancourt, & Gelman, 2017). This is implemented in \texttt{brms} with the \texttt{pp_check} method, that provides various ways of visualising posterior predictive checks. Below we compare the posterior predictions of the three models we fitted previously (n = 100 simulated samples for each model), to the distribution of the original dataset.

```r
library(patchwork) # adding ggplots

# adding ggplots

pp_bmod5 <-
  brms::pp_check(bmod5, nsamples = 1e2) +
  ggtitle("PPC bmod5") +
  theme_bw(base_size = 10) +
  theme(legend.position = "none") +
  xlim(-0.5, 1)

pp_bmod6 <-
  brms::pp_check(bmod6, nsamples = 1e2) +
  ggtitle("PPC bmod6") +
  theme_bw(base_size = 10) +
  theme(legend.position = "none") +
  xlim(-0.5, 1)

pp_bmod7 <-
  brms::pp_check(bmod7, nsamples = 1e2) +
  ggtitle("PPC bmod7") +
  theme_bw(base_size = 10) +
  xlim(-0.5, 1)
```
This confirms that the Normal model ($bmod5$), while being a convenient and easy to interpret model, can be improved to better predict the particular features of our data. Using a lognormal or a skew-normal likelihood improves the predictions of the model (i.e., the simulated samples seem to be closer to the observed data).

**Session information**

```r
sessionInfo()
```

```
## R version 3.4.3 (2017-11-30)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS High Sierra 10.13.3
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.4/Resources/lib/libRlapack.dylib
##
## locale:
##
## attached base packages:
## [1] stats  graphics  grDevices  utils  datasets  methods  base
##
## other attached packages:
```
# References


http://doi.org/10.1007/978-1-4612-1694-0_16