Bayesian hierarchical models

Bruno Nicenboim / Shravan Vasishth

2020-03-14

Bayesian hierarchical models (also known as multilevel or mixed-effects models)

Bayesian hierarchical models (also known as multilevel or mixed-effects models)

The N400 effect (hierarchical normal likelihood)

In the EEG literature, it has been shown that words with low-predictability are accompanied by an *N400 effect* in comparison with high-predictable words, this is a relative negativity that peaks around 300-500 after word onset over central parietal scalp sites (first noticed in Kutas and Hillyard 1980, for semantic anomalies and in 1984 for low predictable word; for a review: Kutas and Federmeier 2011).

1. Example from DeLong, Urbach, and Kutas (2005)
    a. The day was breezy so the boy went outside to fly a kite.
    b. The day was breezy so the boy went outside to fly an airplane.

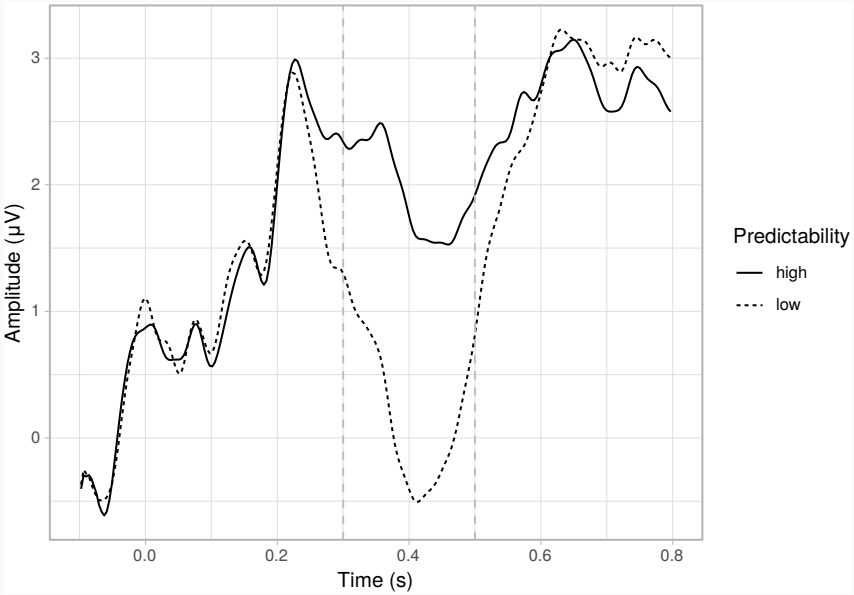


Figure 1: (ref:mot)

- We simplify the high-dimensional EEG data by focusing on the average amplitude of the EEG signal at the typical spatio-temporal window of the N400.
- We focus on the N400 effect for nouns from a subset of the data from Nieuwland et al. (2018). (To speed-up computation, we'll restrict the dataset to the participants from the Edinburgh lab)

```r
df_eeg_data <- read_tsv("data/public_noun_data.txt") %>%
  filter(lab == "edin") %>%
  mutate(c_cloze = cloze / 100 - mean(cloze / 100))
df_eeg_data$c_cloze %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -0.47   -0.44    0.03    0.00    0.43    0.53
```

One nice aspect of this dataset is that the dependent variable is roughly normally distributed:

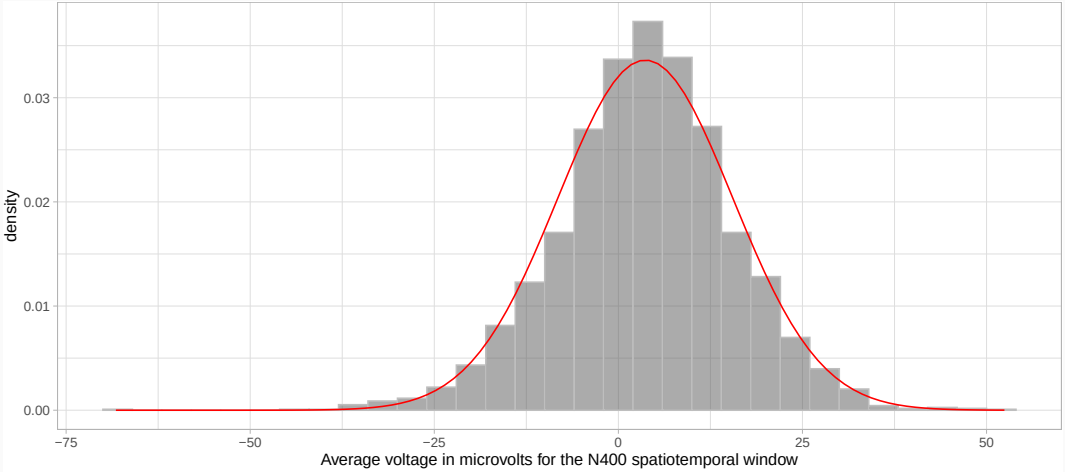


Figure 2: Histogram of the N400 averages for every trial in gray; density plot of a normal distribution in red.

A complete pooling model

We'll start from the simplest model which is basically a linear regression. **Note that this model is incorrect for these data due to point 2 below.**

- Model M_{cp} assumptions:

1. EEG averages for the N400 spatiotemporal window are normally distributed.
2. Observations are *independent*.
3. There is a linear relationship between cloze and the EEG average for the trial.

- Likelihood:

$$signal_n \sim Normal(\alpha + c\_cloze_n \cdot \beta, \sigma) \tag{1}$$

- Priors:

$$\begin{aligned} \alpha &\sim Normal(0, 10) \\ \beta &\sim Normal(0, 10) \\ \sigma &\sim Normal_+(0, 50) \end{aligned} \tag{2}$$

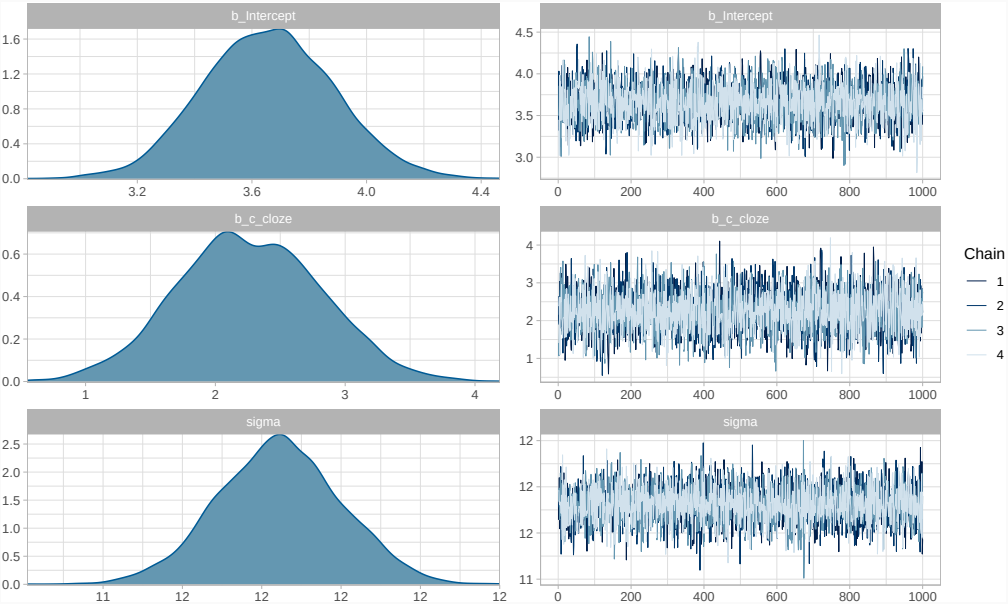
Fitting the model

```
fit_N400_cp <- brm(n400 ~ c_cloze,
  prior =
    c(prior(normal(0, 10), class = Intercept),
      prior(normal(0, 10), class = b),
      prior(normal(0, 50), class = sigma)),
  data = df_eeg_data
)
```

```
fit_N400_cp
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: n400 ~ c_cloze
##    Data: df_eeg_data (Number of observations: 2827)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat
## Intercept     3.66      0.23     3.22     4.10 1.00
## c_cloze       2.26      0.55     1.19     3.33 1.00
##           Bulk_ESS Tail_ESS
## Intercept     4301     3214
## c_cloze       4038     3036
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat
## sigma    11.84      0.16    11.54    12.15 1.00
##       Bulk_ESS Tail_ESS
## sigma     4865     3060
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
```

```
plot(fit_N400_cp)
```



No pooling model

- Model M_{np} assumptions:

1. EEG averages for the N400 spatio-temporal window are normally distributed.
2. Observations depend *completely* on the participant. (Participants have nothing in common.)
3. There is a linear relationship between cloze and the EEG average for the trial.

- Likelihood:

$$signal_n \sim Normal(\alpha_{i[n]} + c\_cloze_n \cdot \beta_{i[n]}, \sigma) \tag{3}$$

- Priors:

$$\begin{aligned} \alpha_i &\sim Normal(0, 10) \\ \beta_i &\sim Normal(0, 10) \\ \sigma &\sim Normal_+(0, 50) \end{aligned} \tag{4}$$

We fit it in brms by removing the common intercept with `0 +` and thus having an intercept and effect for each level of `subject`:

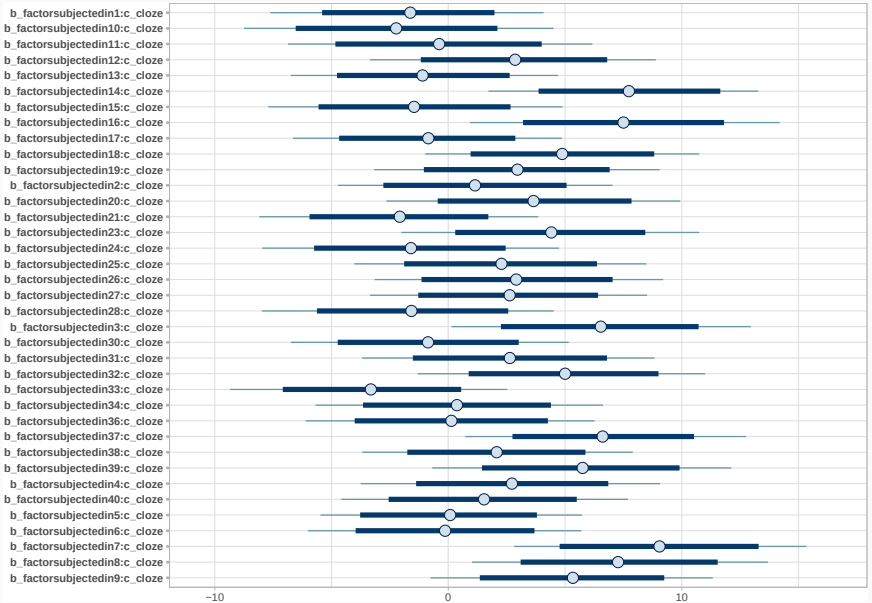
```
fit_N400_np <- brm(n400 ~ 0 +
                     factor(subject) + c_cloze:factor(subject),
                   prior =
                     c(prior(normal(0, 10), class = b),
                       prior(normal(0, 50), class = sigma)),
                   data = df_eeg_data)
```

```
fit_N400_np
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: n400 ~ 0 + factor(subject) + c_cloze:factor(subject)
##    Data: df_eeg_data (Number of observations: 2827)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Population-Level Effects:
##                              Estimate Est.Error
## factorsubjectedin1               5.35      1.35
## factorsubjectedin10              2.72      1.43
## factorsubjectedin11              2.71      1.33
## factorsubjectedin12              7.61      1.30
## factorsubjectedin13              1.30      1.31
## factorsubjectedin14             -0.07      1.35
## factorsubjectedin15              1.20      1.31
## factorsubjectedin16              5.59      1.33
## factorsubjectedin17              2.54      1.28
## factorsubjectedin18              2.52      1.31
## factorsubjectedin19              5.52      1.36
## factorsubjectedin2               3.38      1.36
## factorsubjectedin20              2.61      1.26
## factorsubjectedin21             -0.53      1.34
## factorsubjectedin23              2.84      1.32
## factorsubjectedin24             -0.13      1.38
## factorsubjectedin25              6.29      1.42
## factorsubjectedin26              1.34      1.39
## factorsubjectedin27              7.65      1.30
## factorsubjectedin28              6.27      1.30
```

We plot the estimates using `bayesplot`.

```r
# I first peek at the internal names of the parameters.
# parnames(fit_N400_np)
ind_effects_np <- paste0(
  "b_factorsubject",
  unique(df_eeg_data$subject), ":c_cloze"
)
mcmc_intervals(fit_N400_np,
  pars = ind_effects_np,
  prob = 0.8,
  prob_outer = 0.95,
  point_est = "mean"
)
```


b_factorsubjectedin1:c_cloze
b_factorsubjectedin10:c_cloze
b_factorsubjectedin11:c_cloze
b_factorsubjectedin12:c_cloze
b_factorsubjectedin13:c_cloze
b_factorsubjectedin14:c_cloze
b_factorsubjectedin15:c_cloze
b_factorsubjectedin16:c_cloze
b_factorsubjectedin17:c_cloze
b_factorsubjectedin18:c_cloze
b_factorsubjectedin19:c_cloze
b_factorsubjectedin2:c_cloze
b_factorsubjectedin20:c_cloze
b_factorsubjectedin21:c_cloze
b_factorsubjectedin23:c_cloze
b_factorsubjectedin24:c_cloze
b_factorsubjectedin25:c_cloze
b_factorsubjectedin26:c_cloze
b_factorsubjectedin27:c_cloze
b_factorsubjectedin28:c_cloze
b_factorsubjectedin3:c_cloze
b_factorsubjectedin30:c_cloze
b_factorsubjectedin31:c_cloze
b_factorsubjectedin32:c_cloze
b_factorsubjectedin33:c_cloze
b_factorsubjectedin34:c_cloze
b_factorsubjectedin36:c_cloze
b_factorsubjectedin37:c_cloze
b_factorsubjectedin38:c_cloze
b_factorsubjectedin39:c_cloze
b_factorsubjectedin4:c_cloze
b_factorsubjectedin40:c_cloze
b_factorsubjectedin5:c_cloze
b_factorsubjectedin6:c_cloze
b_factorsubjectedin7:c_cloze
b_factorsubjectedin8:c_cloze
b_factorsubjectedin9:c_cloze
−10
0
10

We can then calculate the average of the β's, even though the model doesn't assume that there's one common β:

```
average_beta_across_subj <-
    posterior_samples(fit_N400_np,
                      pars = ind_effects_np) %>%
    rowMeans()
c(mean=mean(average_beta_across_subj),
  quantile(average_beta_across_subj,
           c(.025,.975)))
```

```
## mean 2.5%  98%
##  2.2  1.2  3.2
```

Varying intercept and varying slopes model (M_v)

- Model M_v assumptions:

1. EEG averages for the N400 spatio-temporal window are normally distributed.
2. Each subject deviates to some extent (this is made precise below) from the grand mean and from the mean effect of predictability.
3. There is a linear relationship between cloze and the EEG average for the trial.

- Likelihood:

$$signal_n \sim Normal(\alpha + u_{0,i[n]} + c\_cloze_n \cdot (\beta + u_{1,i[n]}), \sigma) \tag{5}$$

- Prior:

$$\begin{aligned} \alpha &\sim Normal(0, 10) \\ \beta &\sim Normal(0, 10) \\ u_0 &\sim Normal(0, \tau_{u_0}) \\ u_1 &\sim Normal(0, \tau_{u_1}) \\ \tau_{u_0} &\sim Normal_+(0, 20) \\ \tau_{u_1} &\sim Normal_+(0, 20) \\ \sigma &\sim Normal_+(0, 50) \end{aligned} \tag{6}$$

Some important (and sometimes confusing) points:

- Why does u have a mean of 0?

Because we want u to capture only differences between subjects, we could achieve the same by assuming that

$$
\begin{aligned}
\mu_n &= \alpha_{i[n]} + \beta_{i[n]} \cdot c\_cloze_n \text{and} \\
\alpha_i &\sim Normal(\alpha, \tau_{u_0}) \\
\alpha &\sim Normal(0, 10) \\
\beta_i &\sim Normal(\beta, \tau_{u_1}) \\
\beta &\sim Normal(0, 10)
\end{aligned}
\tag{7}
$$

And in fact, that's another common way to write the model.

- Why do the adjustments u have a normal distribution?

Mostly because of “convention”, that’s the way it’s implemented in most frequentist mixed models.

But also because if we don’t know anything about the distribution besides its mean and variance, the normal distribution is the most conservative assumption (see also chapter 9 of McElreath 2015).

Let's see how we need to set up the priors:

```
get_prior(n400 ~ c_cloze + (c_cloze || subject), data = df_eeg_data)
```

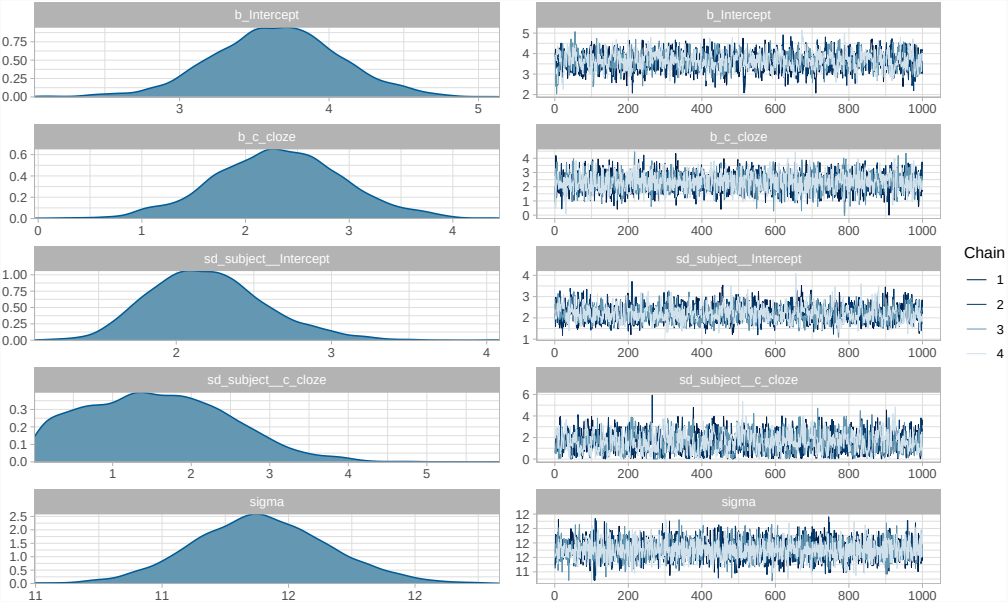
```
##                 prior     class      coef   group resp
## 1                             b
## 2                             b   c_cloze
## 3 student_t(3, 4, 11) Intercept
## 4 student_t(3, 0, 11)        sd
## 5                            sd           subject
## 6                            sd   c_cloze subject
## 7                            sd Intercept subject
## 8 student_t(3, 0, 11)     sigma
##   dpar nlpar bound
## 1
## 2
## 3
## 4
## 5
## 6
## 7
## 8
```

```
fit_N400_v <- brm(n400 ~ c_cloze + (c_cloze || subject),
                  prior =
                    c(prior(normal(0, 10), class = Intercept),
                      prior(normal(0, 10), class = b, coef = c_cloze),
                      prior(normal(0, 50), class = sigma),
                      prior(normal(0, 20), class = sd, coef = Intercept,
                            group = subject),
                      prior(normal(0, 20), class = sd, coef = c_cloze,
                            group = subject)
                      ),
                  data = df_eeg_data)
```

```
fit_N400_v
```

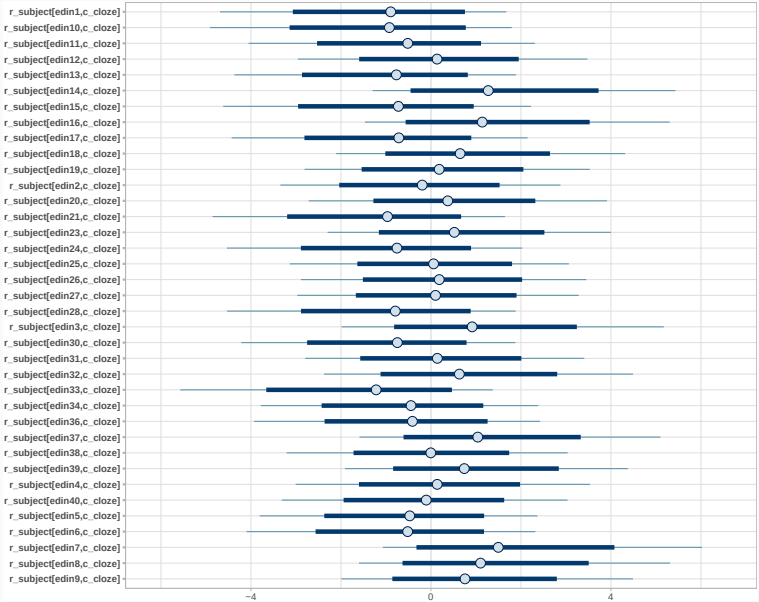
```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: n400 ~ c_cloze + (c_cloze || subject)
##    Data: df_eeg_data (Number of observations: 2827)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~subject (Number of levels: 37)
##               Estimate Est.Error l-95% CI u-95% CI
## sd(Intercept)     2.20      0.37     1.56     3.01
## sd(c_cloze)       1.56      0.90     0.08     3.42
##               Rhat Bulk_ESS Tail_ESS
## sd(Intercept) 1.00     1392     1893
## sd(c_cloze)   1.00     1130     1437
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat
## Intercept     3.65      0.42     2.80     4.48 1.00
## c_cloze       2.32      0.62     1.07     3.58 1.00
##           Bulk_ESS Tail_ESS
```

```
plot(fit_N400_v, N = 6)
```

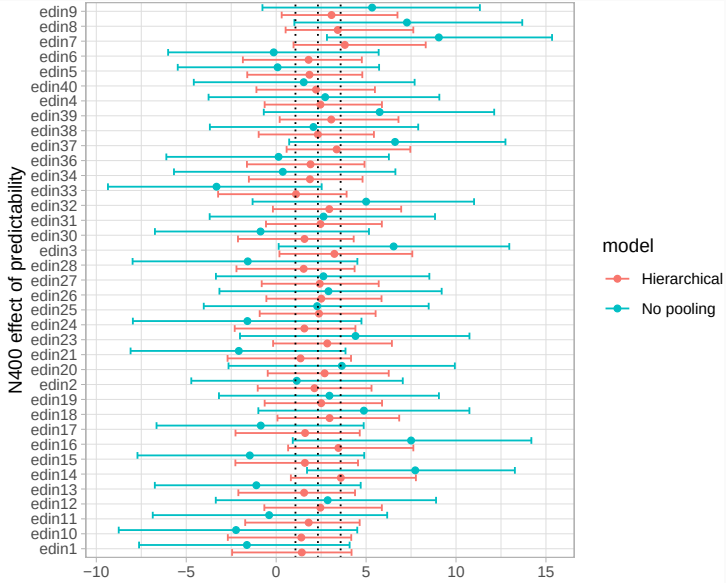


Individual effects

```
# parnames(m_N400_v)
ind_effects_v <- paste0("r_subject[", unique(eeg_data$subject), ",ccloze]")
mcmc_intervals(fit_N400_v,
  pars = ind_effects_v,
  prob = 0.8,
  prob_outer = 0.95,
  point_est = "mean"
)
```


r_subject[edin1,c_cloze]
r_subject[edin10,c_cloze]
r_subject[edin11,c_cloze]
r_subject[edin12,c_cloze]
r_subject[edin13,c_cloze]
r_subject[edin14,c_cloze]
r_subject[edin15,c_cloze]
r_subject[edin16,c_cloze]
r_subject[edin17,c_cloze]
r_subject[edin18,c_cloze]
r_subject[edin19,c_cloze]
r_subject[edin2,c_cloze]
r_subject[edin20,c_cloze]
r_subject[edin21,c_cloze]
r_subject[edin23,c_cloze]
r_subject[edin24,c_cloze]
r_subject[edin25,c_cloze]
r_subject[edin26,c_cloze]
r_subject[edin27,c_cloze]
r_subject[edin28,c_cloze]
r_subject[edin3,c_cloze]
r_subject[edin30,c_cloze]
r_subject[edin31,c_cloze]
r_subject[edin32,c_cloze]
r_subject[edin33,c_cloze]
r_subject[edin34,c_cloze]
r_subject[edin36,c_cloze]
r_subject[edin37,c_cloze]
r_subject[edin38,c_cloze]
r_subject[edin39,c_cloze]
r_subject[edin4,c_cloze]
r_subject[edin40,c_cloze]
r_subject[edin5,c_cloze]
r_subject[edin6,c_cloze]
r_subject[edin7,c_cloze]
r_subject[edin8,c_cloze]
r_subject[edin9,c_cloze]
−4
0
4

Shrinkage


edin9
edin8
edin7
edin6
edin5
edin40
edin4
edin39
edin38
edin37
edin36
edin34
edin33
edin32
edin31
edin30
edin3
edin28
edin27
edin26
edin25
edin24
edin23
edin21
edin20
edin2
edin19
edin18
edin17
edin16
edin15
edin14
edin13
edin12
edin11
edin10
edin1
N400 effect of predictability
−10
−5
0
5
10
15
model
Hierarchical
No pooling

Correlated varying intercept varying slopes model (M_h)

- In M_h, we model the EEG data with the following assumptions:

1. EEG averages for the N400 spatio-temporal window are normally distributed.
2. Some aspects of the signal voltage and the effect of predictability on the signal depend on the participant, and these two might be correlated, i.e., we assume random intercept, slope and correlation by-subject.
3. There is a linear relationship between cloze and the EEG average for the trial.

- Likelihood:

$$signal_n \sim Normal(\alpha + u_{i[n],0} + c\_cloze_n \cdot (\beta + u_{i[n],1}), \sigma) \tag{8}$$

We need to have priors on the adjustments for intercept and slopes, $u_{,0-1}$.

- Priors:

$$\begin{aligned} \alpha &\sim Normal(0, 10) \\ \beta &\sim Normal(0, 10) \\ \sigma &\sim Normal_+(0, 50) \\ \begin{pmatrix} u_{i,0} \\ u_{i,1} \end{pmatrix} &\sim \mathcal{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \Sigma_u\right) \end{aligned} \tag{9}$$

$$\Sigma_u = \begin{pmatrix} \tau_{u_0}^2 & \rho_u \tau_{u_0} \tau_{u_1} \\ \rho_u \tau_{u_0} \tau_{u_1} & \tau_{u_1}^2 \end{pmatrix} \tag{10}$$

And now we need priors for the τ_us and for ρ_u:

$$\begin{aligned} \tau_{u_0} &\sim Normal_+(0, 20) \\ \tau_{u_1} &\sim Normal_+(0, 20) \\ \rho_u &\sim LKJcorr(2) \end{aligned} \tag{11}$$

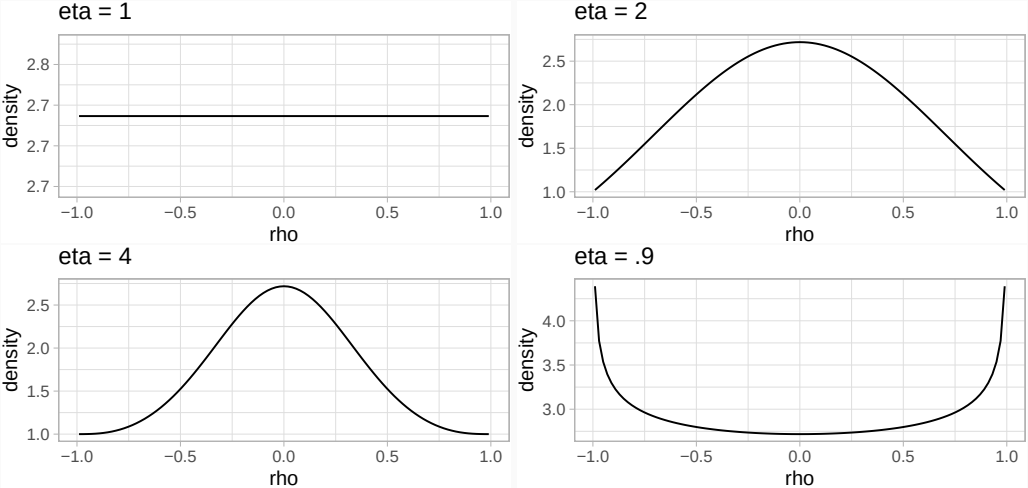


Figure 3: Visualization of the LKJ prior with four different values of the η parameter.

Let's see how we need to set up the priors:

```
get_prior(n400 ~ c_cloze + (c_cloze | subject), data = df_eeg_data)
```

```
##                  prior     class      coef   group
## 1                              b
## 2                              b   c_cloze
## 3              lkj(1)       cor
## 4                            cor           subject
## 5  student_t(3, 4, 11) Intercept
## 6  student_t(3, 0, 11)        sd
## 7                            sd           subject
## 8                            sd   c_cloze subject
## 9                            sd Intercept subject
## 10 student_t(3, 0, 11)     sigma
##    resp dpar nlpar bound
## 1
## 2
## 3
## 4
## 5
## 6
## 7
```

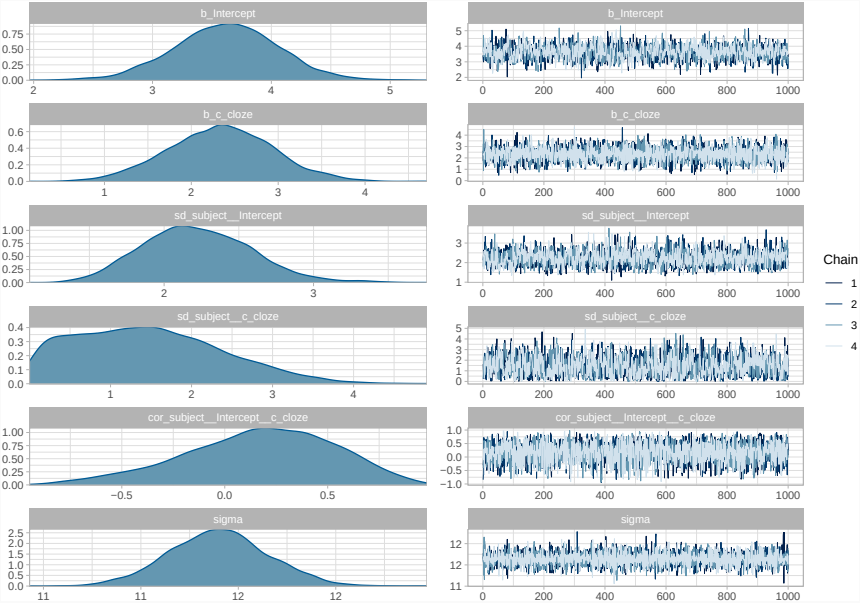
Fitting the model

```
fit_N400_h <- brm(n400 ~ c_cloze + (c_cloze | subject),
                  prior =
                      c(prior(normal(0, 10), class = Intercept),
                        prior(normal(0, 10), class = b, coef = c_cloze),
                        prior(normal(0, 50), class = sigma),
                        prior(normal(0, 20), class = sd, coef = Intercept,
                              group = subject),
                        prior(normal(0, 20), class = sd, coef = c_cloze,
                              group = subject),
                        prior(lkj(2), class = cor,
                              group= subject)),
              data = df_eeg_data)
```

```
fit_N400_h
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: n400 ~ c_cloze + (c_cloze | subject)
##    Data: df_eeg_data (Number of observations: 2827)
## Samples: 4 chains, each with iter = 2000; warmup = 1000; thin = 1;
##          total post-warmup samples = 4000
##
## Group-Level Effects:
## ~subject (Number of levels: 37)
##                         Estimate Est.Error l-95% CI
## sd(Intercept)               2.22      0.36     1.57
## sd(c_cloze)                 1.46      0.90     0.08
## cor(Intercept,c_cloze)      0.17      0.36    -0.60
##                         u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)               2.97 1.00     1418     2688
## sd(c_cloze)                 3.36 1.00     1128     1724
## cor(Intercept,c_cloze)      0.79 1.00     3602     2786
##
## Population-Level Effects:
##           Estimate Est.Error l-95% CI u-95% CI Rhat
## Intercept     3.62      0.44     2.77     4.47 1.00
## c_cloze       2.34      0.60     1.17     3.53 1.00
```

```
plot(fit_N400_h, N = 6)
```



Why should we take the trouble of fitting a Bayesian hierarchical model?

- We can better characterize the generative process by adding the relevant clusters in our data (participants, items, maybe labs, etc)
- The same approach we used here can be used to extend any parameter of any model:
  - (generalized) linear models
  - non-linear/cognitive models

How much structure should we add to our statistical models?

The level of complexity depends on

1. the answers we are looking for
2. the size of the data at hand
3. our computing power
4. our domain and experimental knowledge.

> *"Simplification is essential, but it comes at a cost, and real understanding depends in part on understanding the effects of the simplification" McClelland (2009)*

References

DeLong, Katherine A, Thomas P Urbach, and Marta Kutas. 2005. “Probabilistic Word Pre-Activation During Language Comprehension Inferred from Electrical Brain Activity.” *Nature Neuroscience* 8 (8): 1117–21. https://doi.org/10.1038/nn1504.

Kutas, Marta, and Kara D. Federmeier. 2011. “Thirty Years and Counting: Finding Meaning in the N400 Componentof the Event-Related Brain Potential (ERP).” *Annual Review of Psychology* 62 (1): 621–47. https://doi.org/10.1146/annurev.psych.093008.131123.

Kutas, Marta, and Steven A Hillyard. 1980. “Reading Senseless Sentences: Brain Potentials Reflect Semantic Incongruity.” *Science* 207 (4427): 203–5. https://doi.org/10.1126/science.7350657.