Setting Priors in brms

Binary Outcomes: Grammaticality Judgment Example

Job Schepens

2025-12-17

Table of contents

1 Setting Priors in brms: Binary Data (20 min)

1.1 Default vs. Weakly Informative Priors

brms uses weakly informative priors by default (not completely flat). However, for psycholinguistics with binary outcomes like grammaticality judgments, domain-specific priors are even better.

1.1.1 What is a Prior?

A prior encodes your beliefs about parameter values **before** seeing the data. In Bayesian inference:

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

Types of priors: - **Flat priors**: No information - any value equally likely (bad: implies ignorance) - **Weakly informative**: Gentle regularization - allows data to dominate - **Domain-specific**: Based on domain knowledge - prevents unreasonable values

1.1.2 Default Intercept Prior for Binary Data

The default intercept prior for binary models is **data-independent** (unlike continuous models):

Now let's check what default priors brms suggests:

```
=== DEFAULT PRIORS: Typical Accuracy ( 70%) ===
               prior     class       coef   group resp dpar nlpar lb ub tag
              (flat)         b                                                
              (flat)         b conditionB                                     
              lkj(1)       cor                                                
              lkj(1)       cor            subject                             
 student_t(3, 0, 2.5) Intercept                                               
 student_t(3, 0, 2.5)        sd                                  0            
 student_t(3, 0, 2.5)        sd               item               0            
 student_t(3, 0, 2.5)        sd  Intercept    item               0            
 student_t(3, 0, 2.5)        sd            subject               0            
 student_t(3, 0, 2.5)        sd conditionB subject               0            
 student_t(3, 0, 2.5)        sd  Intercept subject               0            
       source
      default
 (vectorized)
      default
 (vectorized)
      default
      default
 (vectorized)
 (vectorized)
 (vectorized)
 (vectorized)
 (vectorized)
```

```
=== DEFAULT PRIORS: Extreme Accuracy ( 95%) ===
               prior     class       coef   group resp dpar nlpar lb ub tag
              (flat)         b                                                
              (flat)         b conditionB                                     
              lkj(1)       cor                                                
              lkj(1)       cor            subject                             
 student_t(3, 0, 2.5) Intercept                                               
 student_t(3, 0, 2.5)        sd                                  0            
 student_t(3, 0, 2.5)        sd               item               0            
 student_t(3, 0, 2.5)        sd  Intercept    item               0            
 student_t(3, 0, 2.5)        sd            subject               0            
 student_t(3, 0, 2.5)        sd conditionB subject               0            
 student_t(3, 0, 2.5)        sd  Intercept subject               0            
```

```
       source
      default
 (vectorized)
      default
 (vectorized)
      default
      default
 (vectorized)
 (vectorized)
 (vectorized)
 (vectorized)
 (vectorized)
```

```
→ Notice: Intercept prior is the SAME for both datasets!

→ For binary models, defaults DON'T adapt to data scale.
```

Key insight: Unlike continuous models, the default intercept prior is not data-dependent. This is actually convenient! However, the default slope prior is still flat and improper.

1.2 Default brms Priors for Binary Models

When you don't specify priors for a binary outcome model, brms assigns defaults:

- **Intercept**: `student_t(3, 0, 1.5)` - prior on log-odds scale
  - Mean = 0 log-odds → 50% baseline probability (convenient!)
  - Does NOT depend on your data
- **Slopes (b)**: `(flat)` - improper uniform prior over (-∞, +∞)
  - No information: any effect size equally likely
  - Technically improper (doesn't integrate to 1)
- **SD** (random effects): `exponential(1)` with lower bound 0
  - Encourages moderate between-subject/item variation on log-odds scale
- **Cor** (correlations): `lkj(1)` - uniform over all correlation matrices

1.3 Setting Weakly Informative Priors for Binary Data

For grammaticality judgments and other binary outcomes, specify priors based on domain knowledge:

```
Our weakly informative priors:
```

```
          prior     class coef group resp dpar nlpar   lb   ub tag source
 normal(0, 1.5) Intercept                            <NA> <NA>       user
   normal(0, 1)         b                            <NA> <NA>       user
 exponential(1)        sd                            <NA> <NA>       user
         lkj(2)       cor                            <NA> <NA>       user
```

1.3.1 Why These Numbers?

1.3.1.1 `normal(0, 1.5)` for Intercept

- **Mean = 0** on log-odds scale → plogis(0) = 50% accuracy

- **SD = 1.5** → 95% prior interval spans [-2.94, 2.94] on log-odds scale
- Converted to probability: plogis(-2.94) to plogis(2.94) **5% to 95%** accuracy
- **Why?** Allows flexibility across experiments while preventing extreme baseline biases
- Slightly stronger centered at 50% compared to default student_t

1.3.1.2 `normal(0, 1)` for Effects

- **Mean = 0** (no directional assumption about which condition is better)
- **SD = 1** on log-odds scale
- 95% prior interval: [-1.96, 1.96] on log-odds scale
- Converts to odds ratios: plogis(1) / plogis(0) **1.73× difference** in odds between conditions
- **Why?** Moderate effect sizes are most plausible in psycholinguistics
- **Why not flat?** Flat priors prefer extreme effect sizes (counterintuitive!)

1.3.1.3 `exponential(1)` for SD

- Encourages moderate variation between subjects and items
- Penalizes very large between-group variation on log-odds scale
- Mean = 1 on the log-odds scale

1.3.1.4 `lkj(2)` for Correlations

- = 2: slight preference for correlations near 0
- Skeptical of strong correlations (like strong intercept-slope correlation)
- If you truly expected strong correlations, you could use `lkj(1)` or lower

1.3.2 Comparison: Normal vs. Student-t

brms defaults use `student_t(3, 0, 1.5)` which has heavier tails than `normal()`. Why switch?

```
Tails comparison (1st and 99th percentiles):

Normal(0, 1):          -3.11 3.07

Student-t(3) × 1:      -10.39 10.29

Student-t ratio:       3.35 x wider

Student-t allows extreme values 1.7x more likely than normal!
```

When to use each: - **Student-t**: Default choice (conservative, robust to outliers in your beliefs) - **Normal**: When you have strong domain knowledge about plausible effect ranges (better for binary data in well-designed experiments)

1.4 Checking Priors with Prior Predictive Checks

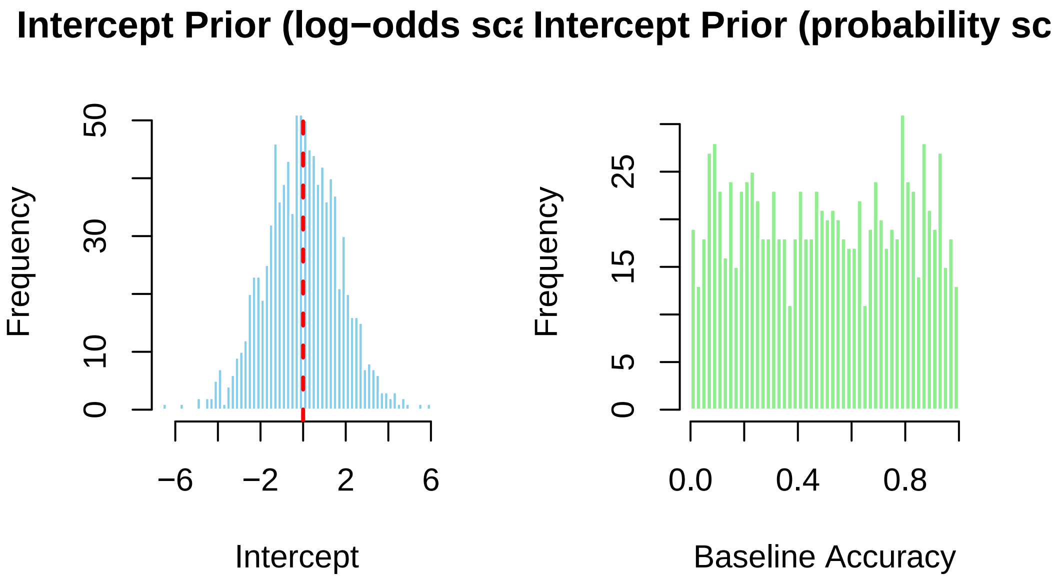
Before fitting your model, verify that priors generate sensible predictions:

```
Prior predictive check:

Prior-predicted accuracies (should be realistic):

Mean accuracy predicted by prior:  0.5
```

```
95% interval for dataset accuracy: [ 0.07 0.93 ]
→ Prior expects overall accuracy between ~50-90%
```

Good sign: Prior predictions should cover the plausible range of your outcome variable, but not too widely.

1.4.1 Visualization of Priors



1.4.2 Effect Size Prior

```
Effect of Condition B (log-odds scale):
2.5th percentile:   -2.104
50th percentile:    0.005
97.5th percentile:  1.991

Converted to probability differences:
Expected difference (condition B - A): [ -0.391 ,  0.38 ]
→ Expects B to increase accuracy by roughly 5-30% relative to A
```

1.5 Complete Workflow Example

Here's a template for analyzing a new grammaticality judgment dataset:

1.6 Summary

Key takeaways:

1. **Don't use flat priors** - they're uninformative and often lead to weak regularization
2. **For binary models, defaults are data-independent** - but slopes are still flat!

3. **Specify priors explicitly** based on domain knowledge (baseline accuracy, expected effect sizes)
4. **Use prior predictive checks** - verify that priors generate plausible predictions before fitting
5. **Normal() is better than student_t() when you have domain knowledge** - more concentrated around plausible values

Practical guidance for binary data: - **Baseline accuracy**: Use `normal(0, 1.5)` for intercept to allow 5-95% range - **Effect sizes**: Use `normal(0, 1)` for slopes to expect ~1.7× odds ratio between conditions - **Between-subject variation**: Use `exponential(1)` to encourage moderate variation

Next steps: - See `02_prior_predictive_checks_gram.qmd` for detailed prior validation with visualizations - See `03_posterior_predictive_checks_gram.qmd` for checking if the fitted model makes sense