Prior Sensitivity Analysis

Nele Albers, Ramya Ghantasala

18 April, 2023

Contents

Introduction

This document contains the prior sensitivity analysis conducted to assess the impact of different settings for the priors in our Bayesian analysis. The posterior probabilities obtained from the different settings are compared to the one from the original model.

Required files: `scripts/preprocessing.r`

Setup

First, let's install the packages that we need.

```
library(formatR)  # To wrap lines
library(reshape2)
library(rethinking)
```

Now we run the pre-processing.

```
source("scripts/preprocessing.r")
```

Original Model

Let's first fit the model with the original priors.

```
set.seed(4)  # For reproducibility

m_orig <- ulam(
  alist(
    motivation ~ dstudent(v, mu, sigma),
    mu <- a + (a_person[pid] * sigma_p) + (b * messageType),
    # Priors
    a_person[pid] ~ dnorm(0, 1),
    sigma_p ~ dexp(1),
    v ~ gamma(2, 0.1),
    a ~ dnorm(0, 10),
    b ~ dnorm(0, 10),
    sigma ~ dexp(1)
  ), data = combinedMotivation, iter = 10000, chains = 4, cores = 4, log_lik = TRUE,
  control=list(adapt_delta=.99, max_treedepth = 15)
)
```

And we extract samples from the fit model and use them to calculate the posterior probability that b is greater than 0.

```
set.seed(4)  # For reproducibility
post <- extract.samples(m_orig, 10000)

originalPosteriorProbability <- length(post[which(post$b > 0)])/length(post$b)
cat("Calculated posterior probability value is ", originalPosteriorProbability)
```

```
## Calculated posterior probability value is  1
```

Models with Different Priors

Now we fit models with different priors.

Model with sigma = 5 for b

```
set.seed(4)  # For reproducibility

m_b_sigma_5 <- ulam(
alist(
  motivation ~ dstudent(v, mu, sigma),
  mu <- a + (a_person[pid] * sigma_p) + (b * messageType),
  # Priors
  a_person[pid] ~ dnorm(0, 1),
  sigma_p ~ dexp(1),
  v ~ gamma(2, 0.1),
  a ~ dnorm(0, 10),
  b ~ dnorm(0, 5),  # changed sigma from 10 to 5
  sigma ~ dexp(1)
), data = combinedMotivation, iter = 10000, chains = 4, cores = 4, log_lik = TRUE,
control=list(adapt_delta=.99, max_treedepth = 15)
)
```

Show parameter estimate information for fit model.

```
precis(m_b_sigma_5, prob = 0.95)
```

```
## 60 vector or matrix parameters hidden. Use depth=2 to show them.
```

```
##             mean         sd      2.5%    97.5%     n_eff     Rhat4
## sigma_p 1.112392 0.12708467 0.8879293 1.383302  4893.977 1.0001918
## v       3.524395 0.41885167 2.7791480 4.419318 20266.148 1.0000166
## a       1.532757 0.17301995 1.1907458 1.871538  5869.627 1.0006084
## b       1.013940 0.13184594 0.7577616 1.275610 24787.391 0.9998901
## sigma   1.432247 0.06552744 1.3061795 1.561271 18258.701 1.0000837
```

Now we extract samples from the fit model and use them to calculate the posterior probability that b is greater than 0. And then we calculate the difference between the new posterior probability value we obtain and the one we obtained based on the original model.

```
set.seed(4)  # For reproducibility
post_b_sigma_5 <- extract.samples(m_b_sigma_5, 10000)

pp_b_sigma_5 <- length(post_b_sigma_5[which(post_b_sigma_5$b > 0)])/length(post_b_sigma_5$b)
cat("Calculated posterior probability value is ", pp_b_sigma_5)
```

```
## Calculated posterior probability value is  1
```

```
cat("Difference in posterior probability value = ", abs(pp_b_sigma_5 -
    originalPosteriorProbability))
```

```
## Difference in posterior probability value =  0
```

Model with sigma = 15 for b

```
set.seed(4)  # For reproducibility

m_b_sigma_15 <- ulam(
alist(
  motivation ~ dstudent(v, mu, sigma),
  mu <- a + (a_person[pid] * sigma_p) + (b * messageType),
  # Priors
  a_person[pid] ~ dnorm(0, 1),
  sigma_p ~ dexp(1),
  v ~ gamma(2, 0.1),
  a ~ dnorm(0, 10),
  b ~ dnorm(0, 15),  # Changed sigma from 10 to 15
  sigma ~ dexp(1)
), data = combinedMotivation, iter = 10000, chains = 4, cores = 4, log_lik = TRUE,
control=list(adapt_delta=.99, max_treedepth = 15)
)
```

Show parameter estimate information for fit model.

```
precis(m_b_sigma_15, prob = 0.95)
```

```
## 60 vector or matrix parameters hidden. Use depth=2 to show them.
```

```
##             mean         sd      2.5%    97.5%     n_eff     Rhat4
## sigma_p 1.114973 0.12579265 0.8916214 1.382072  5859.484 1.0002222
## v       3.523209 0.42330142 2.7650665 4.424191 18551.929 1.0000575
## a       1.527558 0.17403790 1.1849298 1.866721  5007.328 1.0005127
```

```
## b       1.015495 0.13337465 0.7541300 1.278205 23214.677 0.9998658
## sigma   1.431915 0.06554429 1.3039988 1.560900 17020.147 1.0001863
```

Now we extract samples from the fit model and use them to calculate the posterior probability that b is greater than 0. And then we calculate the difference between the new posterior probability value we obtain and the one we obtained based on the original model.

```
set.seed(4)  # For reproducibility
post_b_sigma_15 <- extract.samples(m_b_sigma_15, 10000)

pp_b_sigma_15 <- length(post_b_sigma_15[which(post_b_sigma_15$b > 0)])/length(post_b_sigma_15$b)
cat("Calculated posterior probability value is ", pp_b_sigma_15)
```

```
## Calculated posterior probability value is  1
```

```
cat("Difference in posterior probability value = ", abs(pp_b_sigma_15 -
    originalPosteriorProbability))
```

```
## Difference in posterior probability value =  0
```

Model with dexp(0.1) for sigma_p

```
set.seed(4)  # For reproducibility

m_dexp01_sigmap <- ulam(
alist(
  motivation ~ dstudent(v, mu, sigma),
  mu <- a + (a_person[pid] * sigma_p) + (b * messageType),
  # Priors
  a_person[pid] ~ dnorm(0, 1),
  sigma_p ~ dexp(0.1), # Changed from 1 to 0.1
  v ~ gamma(2, 0.1),
  a ~ dnorm(0, 10),
  b ~ dnorm(0, 10),
  sigma ~ dexp(1)
), data = combinedMotivation, iter = 10000, chains = 4, cores = 4, log_lik = TRUE,
control=list(adapt_delta=.99, max_treedepth = 15)
)
```

Show parameter estimate information for fit model.

```
precis(m_dexp01_sigmap, prob = 0.95)
```

```
## 60 vector or matrix parameters hidden. Use depth=2 to show them.
```

```
##             mean         sd      2.5%    97.5%    n_eff    Rhat4
## sigma_p 1.130056 0.13019437 0.8995874 1.410171  5882.23 1.000287
## v       3.524934 0.41874182 2.7746958 4.412214 21186.65 1.000011
## a       1.534347 0.17555533 1.1889498 1.878098  5566.36 1.000770
## b       1.014196 0.13014966 0.7573481 1.269671 26756.69 1.000086
## sigma   1.431995 0.06535953 1.3045185 1.560731 18760.39 1.000226
```

Now we extract samples from the fit model and use them to calculate the posterior probability that b is greater than 0. And then we calculate the difference between the new posterior probability value we obtain and the one we obtained based on the original model.

```
set.seed(4)  # For reproducibility
post_dexp01_sigmap <- extract.samples(m_dexp01_sigmap, 10000)
```

```
pp_dexp01_sigmap <- length(post_dexp01_sigmap[which(post_dexp01_sigmap$b >
    0)])/length(post_dexp01_sigmap$b)
cat("Calculated posterior probability value is ", pp_dexp01_sigmap)
```

```
## Calculated posterior probability value is  1
```

```
cat("Difference in posterior probability value = ", abs(pp_dexp01_sigmap -
    originalPosteriorProbability))
```

```
## Difference in posterior probability value =  0
```

Model with sigma = 1 for a

```
set.seed(4)  # For reproducibility

m_sigma_1_a <- ulam(
alist(
  motivation ~ dstudent(v, mu, sigma),
  mu <- a + (a_person[pid] * sigma_p) + (b * messageType),
  # Priors
  a_person[pid] ~ dnorm(0, 1),
  sigma_p ~ dexp(1),
  v ~ gamma(2, 0.1),
  a ~ dnorm(0, 1),  # Changed sigma from 10 to 1
  b ~ dnorm(0, 10),
  sigma ~ dexp(1)
), data = combinedMotivation, iter = 10000, chains = 4, cores = 4, log_lik = TRUE,
control=list(adapt_delta=.99, max_treedepth = 15)
)
```

Show parameter estimate information for fit model.

```
precis(m_sigma_1_a, prob = 0.95)
```

```
## 60 vector or matrix parameters hidden. Use depth=2 to show them.
```

```
##             mean         sd      2.5%    97.5%     n_eff     Rhat4
## sigma_p 1.117404 0.12901669 0.8907467 1.395280  4770.657 1.0011516
## v       3.533225 0.41768818 2.8003595 4.420878 17001.110 0.9999920
## a       1.486275 0.17065068 1.1479993 1.814732  5551.747 1.0001818
## b       1.028483 0.12871512 0.7784196 1.279741 20376.977 1.0000266
## sigma   1.433579 0.06525378 1.3067093 1.562941 16037.591 0.9998233
```

Now we extract samples from the fit model and use them to calculate the posterior probability that b is greater than 0. And then we calculate the difference between the new posterior probability value we obtain and the one we obtained based on the original model.

```
set.seed(4)  # For reproducibility
post_sigma_1_a <- extract.samples(m_sigma_1_a, 10000)

pp_sigma_1_a <- length(post_sigma_1_a[which(post_sigma_1_a$b > 0)])/length(post_sigma_1_a$b)
cat("Calculated posterior probability value is ", pp_sigma_1_a)
```

```
## Calculated posterior probability value is  1
```

```
cat("Difference in posterior probability value = ", abs(pp_sigma_1_a -
    originalPosteriorProbability))
```

```
## Difference in posterior probability value =  0
```

Model with sigma = 15 for a

```
set.seed(4)  # For reproducibility

m_sigma_15_a <- ulam(
alist(
  motivation ~ dstudent(v, mu, sigma),
  mu <- a + (a_person[pid] * sigma_p) + (b * messageType),
  # Priors
  a_person[pid] ~ dnorm(0, 1),
  sigma_p ~ dexp(1),
  v ~ gamma(2, 0.1),
  a ~ dnorm(0, 15),  # Changed sigma from 10 to 15
  b ~ dnorm(0, 10),
  sigma ~ dexp(1)
), data = combinedMotivation, iter = 10000, chains = 4, cores = 4, log_lik = TRUE,
control=list(adapt_delta=.99, max_treedepth = 15)
)
```

Show parameter estimate information for fit model.

```
precis(m_sigma_15_a, prob = 0.95)
```

```
## 60 vector or matrix parameters hidden. Use depth=2 to show them.
```

```
##             mean         sd      2.5%    97.5%     n_eff     Rhat4
## sigma_p 1.118306 0.12897928 0.8936195 1.394908  5336.889 1.0002147
## v       3.521770 0.41593594 2.7703782 4.396886 17711.908 1.0000806
## a       1.534146 0.17240401 1.1961370 1.871381  4611.909 1.0005679
## b       1.012925 0.13076788 0.7557323 1.270603 22458.134 0.9999281
## sigma   1.431763 0.06522054 1.3042487 1.559750 15509.062 1.0000859
```

Now we extract samples from the fit model and use them to calculate the posterior probability that b is greater than 0. And then we calculate the difference between the new posterior probability value we obtain and the one we obtained based on the original model.

```
set.seed(4)  # For reproducibility
post_sigma_15_a <- extract.samples(m_sigma_15_a, 10000)

pp_sigma_15_a <- length(post_sigma_15_a[which(post_sigma_15_a$b > 0)])/length(post_sigma_15_a$b)
cat("Calculated posterior probability value is ", pp_sigma_15_a)
```

```
## Calculated posterior probability value is  1
```

```
cat("Difference in posterior probability value = ", abs(pp_sigma_15_a -
    originalPosteriorProbability))
```

```
## Difference in posterior probability value =  0
```

Model with sigma = 15 for a and dexp(0.1) for sigma

```
set.seed(4)  # For reproducibility

m_sigma15_dexp01_sigma <- ulam(
alist(
```

```
  motivation ~ dstudent(v, mu, sigma),
  mu <- a + (a_person[pid] * sigma_p) + (b * messageType),
  # Priors
  a_person[pid] ~ dnorm(0, 1),
  sigma_p ~ dexp(1),
  v ~ gamma(2, 0.1),
  a ~ dnorm(0, 15),  # Changed from 10 to 15
  b ~ dnorm(0, 10),
  sigma ~ dexp(0.1)  # Changed from 1 to 0.1
), data = combinedMotivation, iter = 10000, chains = 4, cores = 4, log_lik = TRUE,
control=list(adapt_delta=.99, max_treedepth = 15)
)
```

Show parameter estimate information for fit model.

```
precis(m_sigma15_dexp01_sigma, prob = 0.95)
```

```
## 60 vector or matrix parameters hidden. Use depth=2 to show them.
```

```
##             mean         sd      2.5%    97.5%     n_eff     Rhat4
## sigma_p 1.113969 0.12586320 0.8913569 1.386040  5768.991 1.0004441
## v       3.542803 0.42860727 2.7933998 4.468493 20043.816 0.9999482
## a       1.533472 0.17461868 1.1856340 1.873775  4928.284 1.0004540
## b       1.014612 0.13381005 0.7516316 1.276144 25861.640 0.9998952
## sigma   1.435980 0.06625929 1.3071290 1.567630 18172.385 1.0000941
```

Now we extract samples from the fit model and use them to calculate the posterior probability that b is greater than 0. And then we calculate the difference between the new posterior probability value we obtain and the one we obtained based on the original model.

```
set.seed(4)  # For reproducibility
post_sigma15_dexp01_sigma <- extract.samples(m_sigma15_dexp01_sigma, 10000)

pp_sigma15_dexp01_sigma <- length(post_sigma15_dexp01_sigma[which(post_sigma15_dexp01_sigma$b >
    0)])/length(post_sigma15_dexp01_sigma$b)
cat("Calculated posterior probability value is ", pp_sigma15_dexp01_sigma)
```

```
## Calculated posterior probability value is  1
```

```
cat("Difference in posterior probability value = ", abs(pp_sigma15_dexp01_sigma -
    originalPosteriorProbability))
```

```
## Difference in posterior probability value =  0
```

Summary

The maximum change in posterior probability when using different priors is 0.