Covariate Selection

Supplementary material

Contents

Data pre-processing

Packages & Options

```
library(tidyverse)
library(brms)
library(projpred)
library(bayesplot)

# retrieve # of cores
ncores <- parallel::detectCores()

# for output clarity
options(scipen = 999)
```

Data

```
# load data
apes1 <- read_csv("../data/laac_data_trial.csv")
apes2 <- read_csv("../data/laac_data_task.csv")

fn0 <- function(x, ...) {
  # helper function
  # sum over correct choice variable (code)
  to_return = tibble(cogn = sum(x$code))
  return(to_return)
}

code_sum <- apes1 %>%
  # contains summed code variable [for each task, time point, session and subject]
  group_by(time_point, session, subject, task) %>%
  group_modify(fn0)
```

```r
apes1_tmp <- apes1 %>%
  # helper for merging
  select(-c(date, trial_session, trial_time_point, code)) %>%
  unique(by = c("time_point", "session", "subject"))

apes1_new <-
  as_tibble(merge(apes1_tmp, code_sum, by = c("time_point", "session", "subject", "task"))) %>%
  mutate(across(c(subject, group, heat, test_day, le_present, dist_present, sex, rearing, observer), as
  mutate(observer = fct_relevel(observer, "no")) %>%
  jtools::center(.,vars = c("sick_severity",
                            "le_mean",
                            "time_outdoors",
                            "age",
                            "time_in_leipzig")) %>%
  group_by(group, time_point) %>%
  mutate(rank_gmc = rank - mean(rank, na.rm = TRUE)) %>%
  ungroup() %>%
  arrange(time_point)

grp_size <- tibble(
  # number of apes for each species
  a_chimp = 20,
  b_chimp = 6,
  bonobo = 12,
  gorilla = 6,
  orangutan = 6
)

apes1_new <- apes1_new %>%
  # create rank variable depending on species
  group_by(group, time_point) %>%
  mutate(
    rel_rank = case_when(
      group == "a_chimp" ~ percent_rank(grp_size$a_chimp:1)[rank],
      group == "b_chimp" ~ percent_rank(grp_size$b_chimp:1)[rank],
      group == "bonobo" ~ percent_rank(grp_size$bonobo:1)[rank],
      group == "gorilla" ~ percent_rank(grp_size$gorilla:1)[rank],
      group == "orangutan" ~ percent_rank(grp_size$orangutan:1)[rank]
    )
  ) %>%
  ungroup()

apes1_new <- apes1_new %>%
  # create coding for heat variable
  mutate(heat_mod = case_when(
    sex == "f" & heat == "yes" ~ "_f_fheat",
    sex == "m" & heat == "yes" ~ "_m_fheat",
    sex == "f" & heat == "no" ~ "_f_noheat",
    sex == "m" & heat == "no" ~ "_m_noheat"),
    heat_mod = as_factor(heat_mod)
  ) %>%
  mutate(heat_mod = fct_relevel(heat_mod, "_f_noheat"))
```

```r
apes1_new <- apes1_new %>%
  select(-heat, -heat_mod)

apes1_new <- apes1_new %>%
  # recode rearing categories: hand -> unknown
  mutate(rearing = fct_recode(rearing, "hand" = "unknown"))

t_cau <- filter(apes1_new, task == "causality")
t_inf <- filter(apes1_new, task == "inference")
t_quant <- filter(apes1_new, task == "quantity")
t_gaze <- filter(apes1_new, task == "gaze_following")

t_gaze <- t_gaze %>%
  # create dummy variable indicating if session 1 or 2
  group_by(time_point, session) %>%
  mutate(tp_mod = cur_group_id()) %>%
  ungroup() %>%
  mutate(day2 = case_when(session == 1 ~ "no",
                          session == 2 ~ "yes"),
         day2 = factor(day2)) %>%
  select(tp_mod, day2, everything())

t_gaze <- t_gaze %>%
  # remove duplicates created by day2
  group_by(subject) %>%
  filter(!duplicated(tp_mod)) %>%
  ungroup()
```

Covariate selection

```r
# covariate needed for projection prediction
# placed here for easy editing of formula
all_fixed_effects <- c("sick_severity",
                       "test_day", "test_tp",
                       "rel_rank",
                       "observer",
                       "age", "time_in_leipzig",
                       "sex", "group",
                       "rearing",
                       "le_mean",
                       "dist_mean",
                       "time_outdoors",
                       "sociality")

fm <- formula(cogn ~ sick_severity +
                test_day + test_tp +
                rel_rank + # rank_gmc +
                observer +
                age + time_in_leipzig +
                sex + group +
                rearing +
                le_mean + # le_max + # le_present +
```

```
                dist_mean + # dist_max + # + dist_present +
                time_outdoors +
                sociality + # sociality_total
                # heat_mod + # heat +
                (1|subject)
              )

fm_gaze <- update(fm, . ~ . +day2)
```

Reference Model: 2-level Multilevel Model (random intercepts only)

```
m_cau_2l <- brm(fm, data = t_cau,
                warmup = 1e3, iter = 4e3, cores = ncores, chains = 4,
                seed = 2021,
                save_pars = save_pars(all = TRUE)
                )
m_inf_2l <- brm(fm, data = t_inf,
                warmup = 1e3, iter = 4e3, cores = ncores, chains = 4,
                seed = 2021,
                save_pars = save_pars(all = TRUE)
                )
m_quant_2l <- brm(fm, data = t_quant,
                  warmup = 1e3, iter = 4e3, cores = ncores, chains = 4,
                  seed = 2021,
                  save_pars = save_pars(all = TRUE)
                  )
m_gaze_2l <- brm(fm_gaze, data = t_gaze,
                 warmup = 1e3, iter = 4e3, cores = ncores, chains = 4,
                 seed = 2021,
                 save_pars = save_pars(all = TRUE)
                 )
```

```
library(loo)
lapply(list(m_cau_2l, m_inf_2l, m_quant_2l, m_gaze_2l), loo)
```

```
summary(m_cau_2l)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: cogn ~ sick_severity + test_day + test_tp + rel_rank + observer + age + time_in_leipzig + s
##    Data: t_cau (Number of observations: 450)
## Samples: 4 chains, each with iter = 4000; warmup = 1000; thin = 1;
##          total post-warmup samples = 12000
##
## Group-Level Effects:
## ~subject (Number of levels: 41)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     2.19      0.32     1.65     2.89 1.00     4533     6817
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept           7.10      1.66     3.82    10.41 1.00     7279     8270
## sick_severity       0.19      0.19    -0.19     0.56 1.00    16260     9087
## test_dayyes         0.71      1.09    -1.42     2.88 1.00    16695     9366
```

```
## test_tp             0.00      0.04    -0.08     0.08 1.00    18745     9463
## rel_rank            0.28      1.35    -2.40     2.98 1.00    10121     8424
## observeryes         0.30      0.34    -0.35     0.96 1.00    17567    10104
## age                -0.02      0.05    -0.12     0.08 1.00     7709     7819
## time_in_leipzig     0.11      0.08    -0.05     0.27 1.00     7671     8415
## sexf                0.17      0.86    -1.55     1.84 1.00     7627     8547
## groupb_chimp        3.09      1.61    -0.12     6.30 1.00     6635     7782
## groupa_chimp       -0.82      1.20    -3.18     1.53 1.00     5739     7020
## grouporangutan      1.98      1.77    -1.62     5.45 1.00     6917     7074
## groupbonobo         1.10      1.31    -1.42     3.69 1.00     6247     8045
## rearinghand         0.01      1.22    -2.41     2.36 1.00     7969     8288
## le_mean            -0.67      0.54    -1.73     0.39 1.00    18562     9531
## dist_mean          -0.36      0.21    -0.78     0.06 1.00    15668     9770
## time_outdoors      -0.07      0.06    -0.19     0.04 1.00    15460     9047
## sociality           0.07      0.24    -0.39     0.54 1.00    16087     8636
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     2.22      0.08     2.07     2.38 1.00    14168     8736
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
summary(m_inf_2l)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: cogn ~ sick_severity + test_day + test_tp + rel_rank + observer + age + time_in_leipzig + s
##    Data: t_inf (Number of observations: 451)
## Samples: 4 chains, each with iter = 4000; warmup = 1000; thin = 1;
##          total post-warmup samples = 12000
##
## Group-Level Effects:
## ~subject (Number of levels: 41)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     2.80      0.38     2.17     3.65 1.00     4419     7020
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept           6.31      1.89     2.63    10.07 1.00     7436     8217
## sick_severity      -0.09      0.17    -0.43     0.25 1.00    18592     9161
## test_dayyes         0.80      0.95    -1.07     2.64 1.00    21465     9046
## test_tp            -0.09      0.03    -0.16    -0.02 1.00    21423     9470
## rel_rank           -0.23      1.32    -2.82     2.36 1.00    14037     9317
## observeryes         0.51      0.29    -0.06     1.06 1.00    22336     8669
## age                -0.08      0.06    -0.19     0.05 1.00     7460     7962
## time_in_leipzig     0.37      0.10     0.17     0.56 1.00     7804     7460
## sexf                0.18      1.07    -1.98     2.27 1.00     7449     7339
## groupb_chimp        0.57      2.02    -3.43     4.58 1.00     6699     7571
## groupa_chimp       -0.01      1.50    -2.97     3.02 1.00     5908     6591
## grouporangutan     -2.17      2.12    -6.37     2.01 1.00     6876     7605
## groupbonobo        -1.58      1.67    -4.92     1.77 1.00     6433     7179
## rearinghand        -0.49      1.48    -3.43     2.43 1.00     7625     7590
## le_mean            -0.57      0.45    -1.46     0.33 1.00    22706     8269
```

```
## dist_mean          -0.11      0.18    -0.46     0.24 1.00    18597     8971
## time_outdoors      -0.01      0.05    -0.11     0.08 1.00    17195     8955
## sociality           0.03      0.20    -0.37     0.42 1.00    20660     8781
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     1.86      0.07     1.74     2.00 1.00    16348     8687
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
summary(m_quant_21)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: cogn ~ sick_severity + test_day + test_tp + rel_rank + observer + age + time_in_leipzig + s
##    Data: t_quant (Number of observations: 421)
## Samples: 4 chains, each with iter = 4000; warmup = 1000; thin = 1;
##          total post-warmup samples = 12000
##
## Group-Level Effects:
## ~subject (Number of levels: 41)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     1.75      0.26     1.31     2.35 1.00     4406     7213
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept           8.74      1.34     6.13    11.38 1.00     9562     8749
## sick_severity      -0.26      0.16    -0.58     0.05 1.00    15739     9718
## test_dayyes         0.42      0.82    -1.18     2.03 1.00    16761     9939
## test_tp             0.02      0.03    -0.05     0.08 1.00    18450     9405
## rel_rank            0.33      1.08    -1.83     2.38 1.00    13256     8858
## observeryes        -0.37      0.31    -0.98     0.23 1.00    15966     9596
## age                -0.03      0.04    -0.11     0.04 1.00     9270     8017
## time_in_leipzig     0.23      0.07     0.11     0.36 1.00     9527     8539
## sexf                0.78      0.70    -0.59     2.14 1.00    10367     8996
## groupb_chimp        1.59      1.31    -1.01     4.19 1.00     8814     8486
## groupa_chimp       -1.52      0.99    -3.49     0.44 1.00     8174     8244
## grouporangutan     -2.07      1.42    -4.88     0.74 1.00     8971     9208
## groupbonobo         0.11      1.07    -2.02     2.23 1.00     8239     7539
## rearinghand        -2.31      0.96    -4.25    -0.44 1.00     9596     8491
## le_mean             0.14      0.44    -0.72     1.01 1.00    18361     9509
## dist_mean          -0.04      0.19    -0.41     0.32 1.00    16275     9959
## time_outdoors      -0.01      0.05    -0.12     0.09 1.00    15895     9574
## sociality          -0.03      0.21    -0.44     0.39 1.00    18675     8957
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     1.77      0.07     1.64     1.90 1.00    14069     8719
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

```
summary(m_gaze_21)
```

```
##  Family: gaussian
##   Links: mu = identity; sigma = identity
## Formula: cogn ~ sick_severity + test_day + test_tp + rel_rank + observer + age + time_in_leipzig + s
##    Data: t_gaze (Number of observations: 879)
## Samples: 4 chains, each with iter = 4000; warmup = 1000; thin = 1;
##          total post-warmup samples = 12000
##
## Group-Level Effects:
## ~subject (Number of levels: 41)
##               Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sd(Intercept)     0.75      0.10     0.58     0.98 1.00     4218     6995
##
## Population-Level Effects:
##                 Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## Intercept           0.56      0.50    -0.41     1.55 1.00     6860     8365
## sick_severity      -0.13      0.05    -0.23    -0.03 1.00    15844     8407
## test_dayyes         0.22      0.23    -0.22     0.66 1.00    18709     8805
## test_tp             0.01      0.01    -0.01     0.02 1.00    22027     8992
## rel_rank           -0.14      0.35    -0.82     0.54 1.00    12451     9498
## observeryes        -0.12      0.08    -0.27     0.04 1.00    18950     9392
## age                 0.02      0.02    -0.01     0.05 1.00     6351     7397
## time_in_leipzig    -0.00      0.03    -0.06     0.05 1.00     7323     7899
## sexf                0.23      0.29    -0.34     0.79 1.00     6948     7378
## groupb_chimp        0.55      0.53    -0.48     1.62 1.00     6547     7347
## groupa_chimp        0.78      0.40    -0.00     1.58 1.00     5559     6881
## grouporangutan      0.08      0.57    -1.04     1.21 1.00     6504     7480
## groupbonobo         0.59      0.44    -0.26     1.45 1.00     6055     6594
## rearinghand        -0.85      0.40    -1.63    -0.05 1.00     6825     7601
## le_mean            -0.03      0.12    -0.27     0.21 1.00    21443     8520
## dist_mean          -0.10      0.05    -0.19    -0.00 1.00    16821     9112
## time_outdoors       0.10      0.01     0.07     0.12 1.00    15600     9673
## sociality          -0.14      0.06    -0.25    -0.03 1.00    20094     8699
## day2yes            -0.12      0.05    -0.21    -0.02 1.00    20069     7768
##
## Family Specific Parameters:
##       Estimate Est.Error l-95% CI u-95% CI Rhat Bulk_ESS Tail_ESS
## sigma     0.71      0.02     0.68     0.75 1.00    17879     8363
##
## Samples were drawn using sampling(NUTS). For each parameter, Bulk_ESS
## and Tail_ESS are effective sample size measures, and Rhat is the potential
## scale reduction factor on split chains (at convergence, Rhat = 1).
```

Predictive Projection

```
# delay random intercept to last place so that it doesn't soak up all the variance
s_terms <- c("1", all_fixed_effects,
             paste0(paste(all_fixed_effects, collapse = " + "), " + (1 | subject)"))
s_terms_gaze <- c("1", c(all_fixed_effects, "day2"),
                  paste0(paste(c(all_fixed_effects, "day2"), collapse = " + "), " + (1 | subject)"))
```

```
refM_cau <- get_refmodel(m_cau_21)
refM_inf <- get_refmodel(m_inf_21)
```

```
refM_quant <- get_refmodel(m_quant_21)
refM_gaze <- get_refmodel(m_gaze_21)
```

```
vs_cau <- varsel(refM_cau, search_terms = s_terms)
summary(vs_cau); plot(vs_cau, stats = c('elpd', 'rmse'))

randint_ind_vscau <- length(solution_terms(vs_cau))
relevant_cov_vscau <- c(1, 2, 3, 4, 5, randint_ind_vscau)

# proj_cau <- project(vs_cau, solution_terms = relevant_cov_vscau)
# mcmc_areas(as.matrix(proj_cau), pars = solution_terms(vs_cau)[relevant_cov_vscau])
```

```
vs_inf <- varsel(refM_inf, search_terms = s_terms)
summary(vs_inf); plot(vs_inf, stats = c('elpd', 'rmse'))

randint_ind_vsinf <- length(solution_terms(vs_inf))
relevant_cov_vsinf <- c(1, 2, 3, 4, 5, randint_ind_vsinf)

#proj_inf <- project(vs_inf, solution_terms = relevant_cov_vsinf, ndraws = 10)
#mcmc_areas(as.matrix(proj_inf), pars = solution_terms(vs_inf)[relevant_cov_vsinf])
```

```
vs_quant <- varsel(refM_quant, search_terms = s_terms)
summary(vs_quant); plot(vs_quant, stats = c('elpd', 'rmse'))

randint_ind_vsquant <- length(solution_terms(vs_quant))
relevant_cov_vsquant <- c(1, 2, 3, 4, 5, randint_ind_vsquant)

# proj_quant <- project(vs_quant, solution_terms = relevant_cov_vsquant)
# mcmc_areas(as.matrix(proj_quant), pars = solution_terms(vs_quant)[relevant_cov_vsquant])
```

```
vs_gaze <- varsel(refM_gaze, search_terms = s_terms_gaze)
summary(vs_gaze); plot(vs_gaze, stats = c('elpd', 'rmse'))

randint_ind_vsgaze <- length(solution_terms(vs_gaze))
relevant_cov_vsgaze <- c(1, 2, 3, 4, 5, randint_ind_vsgaze)

# proj_gaze <- project(vs_gaze, solution_terms = relevant_cov_vsgaze)
# mcmc_areas(as.matrix(proj_gaze), pars = solution_terms(vs_gaze)[relevant_cov_vsgaze])
```

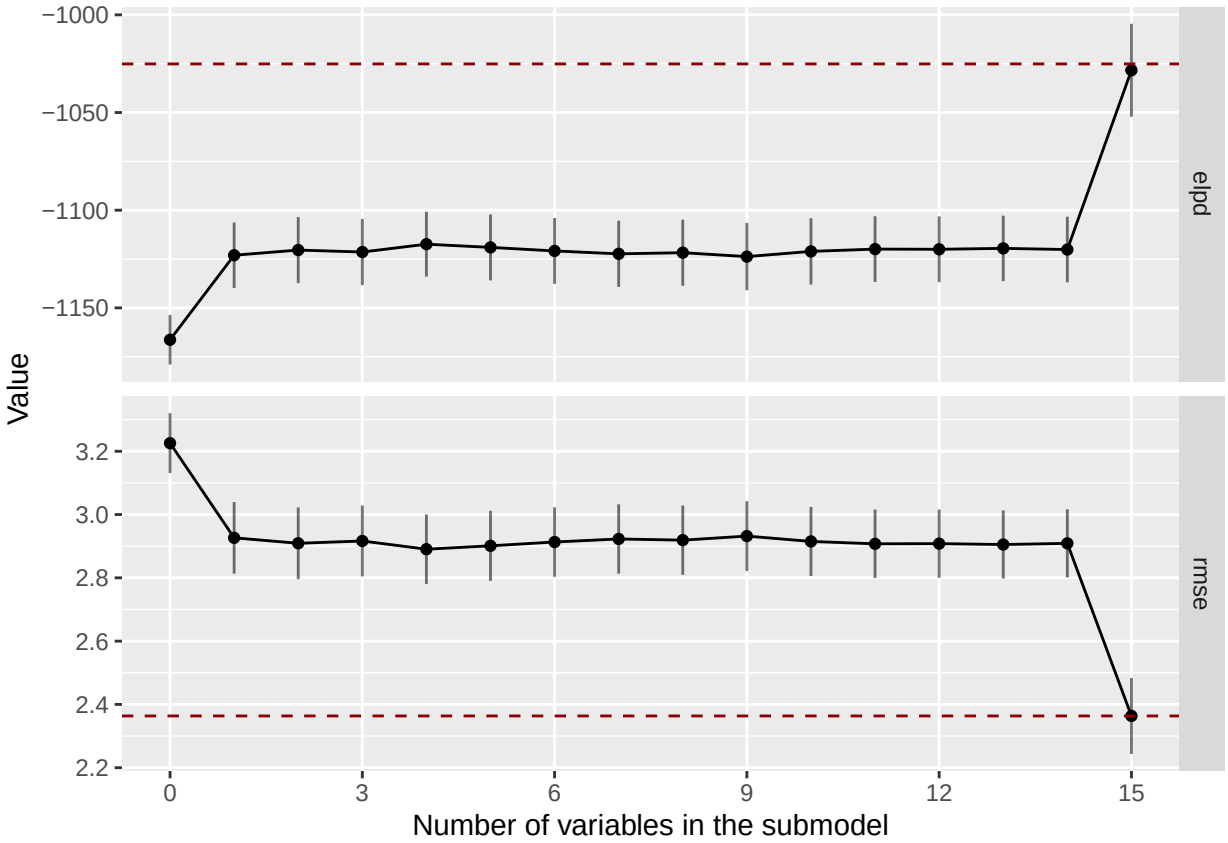
```
cvs_cau <- cv_varsel(refM_cau,
                     search_terms = s_terms, cv_method = "LOO", method = "forward",
                     seed = 2020)
```

```
## [1] "Computing LOOs..."
##   |                                                                      |
```

```
## [1] "100% of terms selected."
## [1] "Done."
```

```
summary(cvs_cau); plot(cvs_cau, stats = c('elpd', 'rmse'))
```

```
##    size  solution_terms      elpd  elpd.se
## 2     0            <NA> -1166.320 12.74171
## 3     1           group -1123.058 16.82876
## 4     2 time_in_leipzig -1120.403 17.01587
## 5     3       sociality -1121.423 17.01735
## 6     4       dist_mean -1117.403 16.68867
## 7     5   sick_severity -1119.044 16.97399
## 8     6             age -1120.857 16.95057
## 9     7         le_mean -1122.342 17.02853
## 10    8        test_day -1121.755 17.02313
## 11    9        rel_rank -1123.767 17.29413
## 12   10        observer -1121.095 17.07342
## 13   11   time_outdoors -1119.914 16.92239
## 14   12             sex -1120.004 16.94353
## 15   13         rearing -1119.549 16.85887
## 16   14         test_tp -1120.135 16.89832
## 17   15   (1 | subject) -1028.359 23.94456
```




```
# proj_cau_cv <- project(cvs_cau, solution_terms = c(1, 2, 3, 14))
# mcmc_areas(as.matrix(proj_cau_cv), pars = solution_terms(cvs_cau)[c(1, 2, 3, 14)])
```

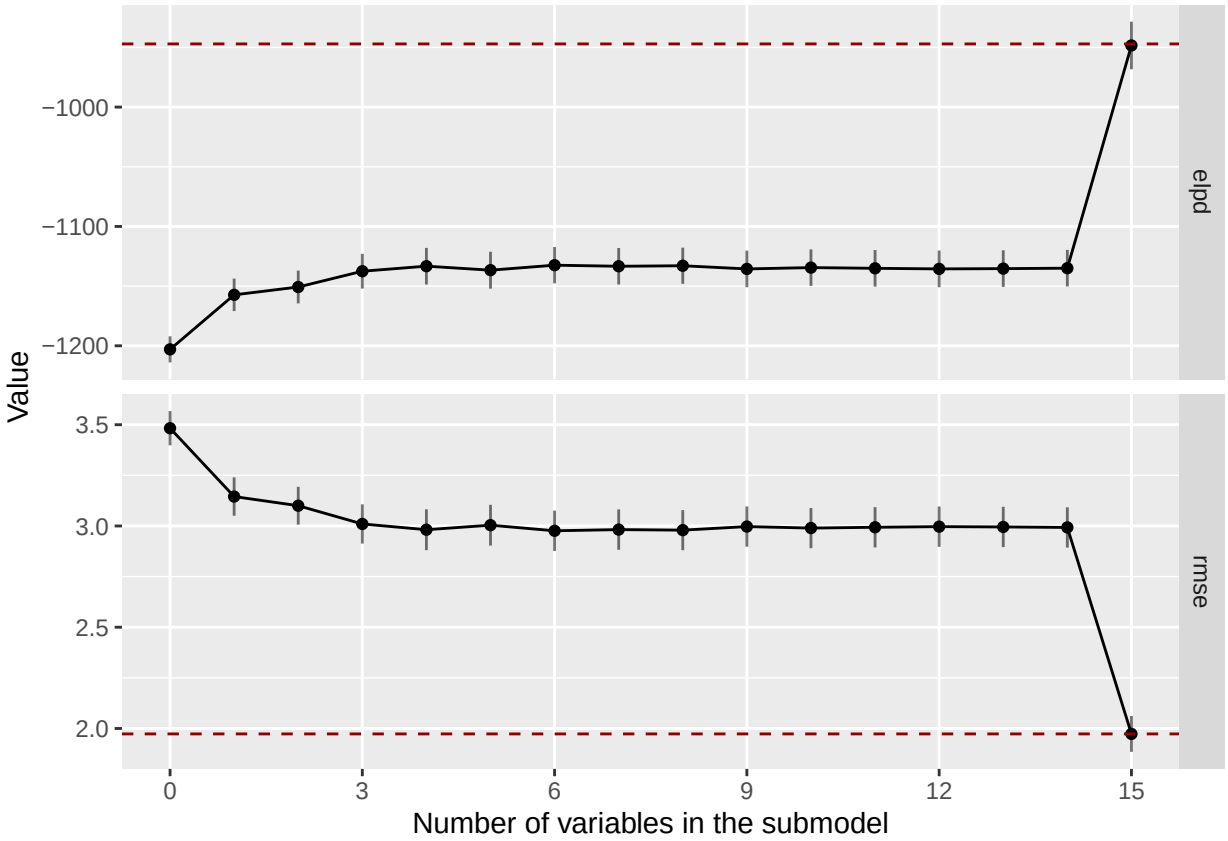
relevant covariates: (1 | subject), group

```
cvs_inf <- cv_varsel(refM_inf,
                     search_terms = s_terms, cv_method = "LOO", method = "forward",
                     seed = 2020)
```

```
## [1] "Computing LOOs..."
##   |                                                                      |
```

```
summary(cvs_inf); plot(cvs_inf, stats = c('elpd', 'rmse'))
```

```
##    size  solution_terms       elpd  elpd.se
## 2     0            <NA> -1202.9857 10.97886
## 3     1 time_in_leipzig -1157.3144 13.67677
## 4     2           group -1150.7333 13.81341
## 5     3             age -1137.5141 14.65069
## 6     4   sick_severity -1133.2952 15.42105
## 7     5        test_day -1136.6343 15.54014
## 8     6   time_outdoors -1132.4628 15.29881
## 9     7        observer -1133.3510 15.38904
## 10    8         rearing -1132.9248 15.24541
## 11    9       sociality -1135.5923 15.40453
## 12   10        rel_rank -1134.4788 15.36408
## 13   11       dist_mean -1135.0805 15.42556
## 14   12         test_tp -1135.5634 15.46405
## 15   13         le_mean -1135.3409 15.44985
## 16   14             sex -1134.9778 15.42959
## 17   15   (1 | subject)  -948.3915 20.07552
```




```
# proj_inf_cv <- project(cvs_inf, solution_terms = c(1, 2, 14))
# mcmc_areas(as.matrix(proj_inf_cv), pars = solution_terms(cvs_inf)[c(1, 2, 14)])
```

relevant covariates: (1 | subject), time_in_leipzig, group, age

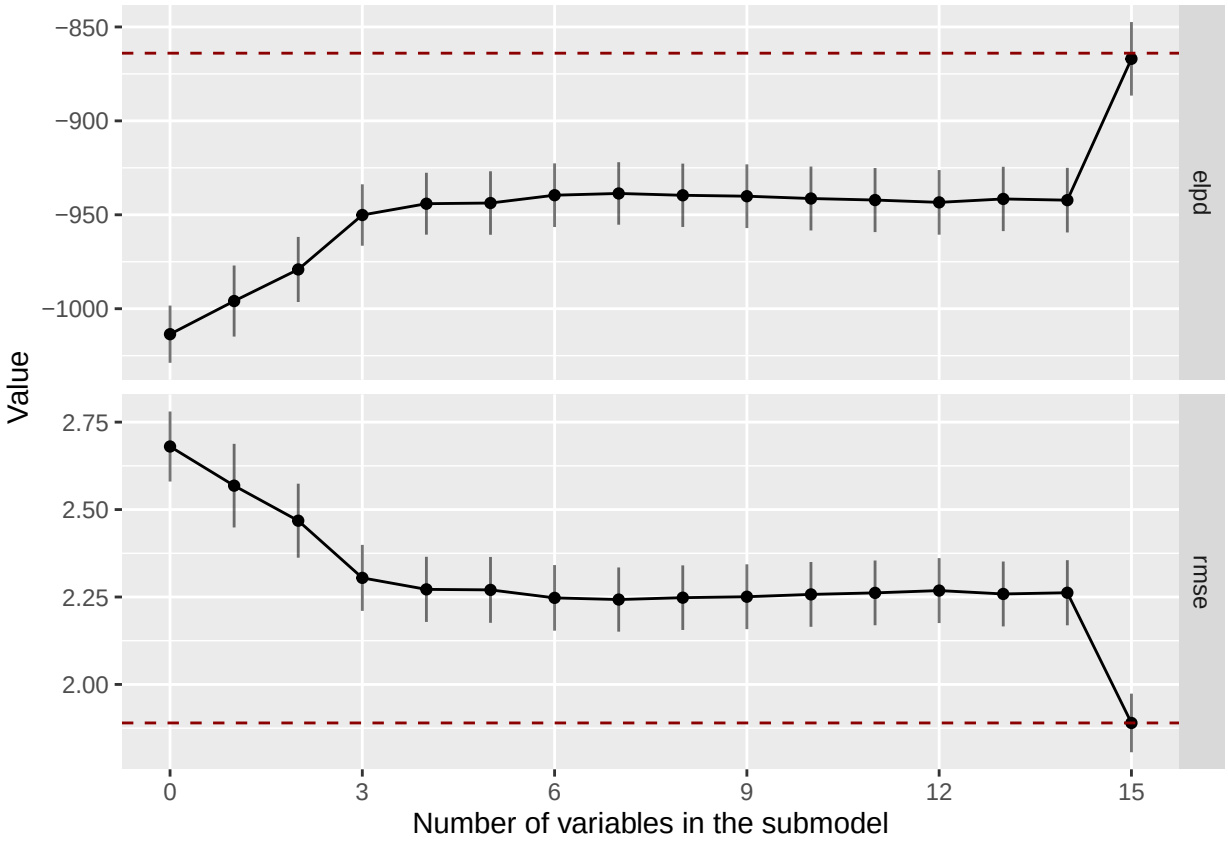
```
cvs_quant <- cv_varsel(refM_quant,
                       search_terms = s_terms, cv_method = "LOO", method = "forward",
                       seed = 2020)
```

```
## [1] "Computing LOOs..."
##   |                                                                      |
```

```
summary(cvs_quant); plot(cvs_quant, stats = c('elpd', 'rmse'))
```

```
##    size  solution_terms       elpd  elpd.se
## 2     0            <NA> -1013.5630 15.32066
```

```
## 3     1 time_in_leipzig  -995.9227 19.01982
## 4     2         rearing  -979.0997 17.41291
## 5     3           group  -950.1172 16.43463
## 6     4        observer  -944.0947 16.56630
## 7     5             sex  -943.7346 16.98652
## 8     6        rel_rank  -939.5734 17.04140
## 9     7             age  -938.6473 16.73218
## 10    8   sick_severity  -939.6164 16.96110
## 11    9         test_tp  -940.1058 17.06342
## 12   10        test_day  -941.3380 17.09880
## 13   11   time_outdoors  -942.1494 17.15309
## 14   12       dist_mean  -943.3841 17.29724
## 15   13         le_mean  -941.5656 17.21523
## 16   14       sociality  -942.2124 17.29028
## 17   15   (1 | subject)  -866.9580 19.68460
```




```
# proj_quant_cv <- project(cvs_quant, solution_terms = c(1, 2, 3, 15))
# mcmc_areas(as.matrix(proj_quant_cv), pars = solution_terms(cvs_quant)[c(1, 2, 3, 15)])
```

relevant covariates: (1 | subject), time_in_leipzig, rearing, group

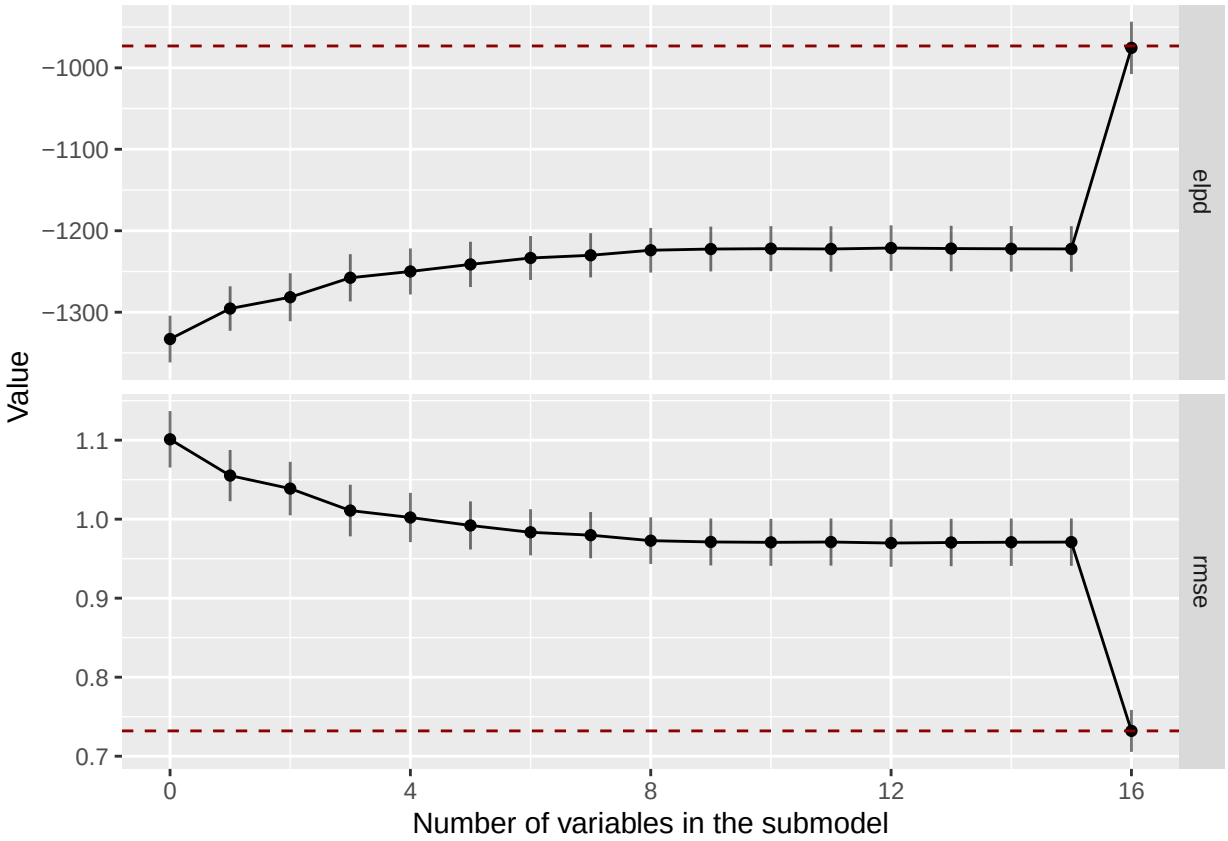
```
cvs_gaze <- cv_varsel(refM_gaze,
                      search_terms = s_terms_gaze, cv_method = "LOO", method = "forward",
                      seed = 2020)
```

```
## [1] "Computing LOOs..."
##   |                                                                      |
```

```
## [1] "10% of terms selected."
## [1] "20% of terms selected."
## [1] "30% of terms selected."
## [1] "40% of terms selected."
## [1] "50% of terms selected."
## [1] "60% of terms selected."
## [1] "70% of terms selected."
## [1] "80% of terms selected."
## [1] "90% of terms selected."
## [1] "100% of terms selected."
## [1] "Done."
```

```
summary(cvs_gaze); plot(cvs_gaze, stats = c('elpd', 'rmse'))
```

```
##    size  solution_terms      elpd  elpd.se
## 2     0            <NA> -1332.926 28.85612
## 3     1           group -1295.514 27.45796
## 4     2         rearing -1281.612 29.61006
## 5     3   time_outdoors -1257.730 29.16308
## 6     4             age -1249.990 28.41721
## 7     5       sociality -1241.305 27.96193
## 8     6             sex -1233.446 27.09530
## 9     7   sick_severity -1230.140 27.31724
## 10    8        observer -1223.934 27.57003
## 11    9 time_in_leipzig -1222.454 27.68608
## 12   10            day2 -1222.011 27.80184
## 13   11       dist_mean -1222.380 28.01366
## 14   12        test_day -1221.232 28.10263
## 15   13        rel_rank -1221.860 28.08296
## 16   14         test_tp -1222.162 28.09163
## 17   15         le_mean -1222.340 28.09300
## 18   16   (1 | subject)  -975.499 32.36394
```




```
# proj_gaze_cv <- project(cvs_gaze, solution_terms = c(1, 2, 3, 4, 5, 6, 7, 15))
# mcmc_areas(as.matrix(proj_gaze_cv), pars = solution_terms(cvs_gaze)[c(1, 2, 3, 4, 5, 6, 7, 15)])
```

relevant covariates: (1 | subject), group, rearing, time_outdoors, age, sociality, sex