Partnership example

Jouni Helske and Santtu Tikka

2024-04-08

The data used in this example is from the companion website of the book "Sequence Analysis" by Marcel Raab and Emanuela Struffolino (2022): https://sa-book.github.io/.

Load some packages and read the data:

```
library(dplyr)
library(tidyr)
library(posterior)
library(dynamite)
library(ggplot2)
library(RColorBrewer)
# See the Rmd source for the code on how to create this file from SA book files
d <- readRDS("family_data.rds") |>
  filter(!is.na(church)) # remove 32 individuals with missing church variable
```

```
# cmdstanr backend as it is currently faster than rstan with categorical responses
set.seed(1)
fit <- dynamite(
  obs(status ~ -1 + lag(status) + sex + church + random(~1), "categorical") +
    random_spec(correlated = TRUE),
    data = d, group = "id", time = "time",
  backend = "cmdstanr", parallel_chains = 4,
  iter_sampling = 5000, iter_warmup = 1000, refresh = 0,
  save_warmup = FALSE, stanc_options = list("O1"))
# this is not stored in the repo due to its size
saveRDS(fit, file = "fit_partnership.rds")
```

Check MCMC diagnostics:

```
mcmc_diagnostics(fit)
```

```
## NUTS sampler diagnostics:
##
## No divergences, saturated max treedepths or low E-BFMIs.
##
## Smallest bulk-ESS values:
##
## sigma_nu_status_alpha_COH                  2343
## corr_nu_status_alpha_COH__status_alpha_MAR 3912
## sigma_nu_status_alpha_LAT                  4499
##
## Smallest tail-ESS values:
##
## sigma_nu_status_alpha_COH                  3180
## corr_nu_status_alpha_COH__status_alpha_MAR 5461
```

```
## corr_nu_status_alpha_LAT__status_alpha_COH 5915
##
## Largest Rhat values:
##
## sigma_nu_status_alpha_COH 1
## nu_status_alpha_COH_id384 1
## nu_status_alpha_LAT_id683 1
```

Parameter estimates:

```
as_draws(fit, types = c("beta", "sigma_nu", "corr_nu")) |>
  posterior::summarise_draws(
    "mean",
    "sd",
    ~quantile(.x, probs = c(0.025, 0.975)),
     "rhat", "ess_bulk", "ess_tail") |>
  print(n = Inf)
```

```
## # A tibble: 24 x 8
##    variable                 mean     sd  `2.5%`  `97.5%`  rhat ess_bulk ess_tail
##    <chr>                   <dbl>  <dbl>   <dbl>    <dbl> <dbl>    <dbl>    <dbl>
##  1 beta_status_churchYes~ -0.106 0.0584 -0.221   0.00953  1.00   17086.   16541.
##  2 beta_status_churchYes~  0.185 0.0532  0.0800  0.289    1.00   15176.   16466.
##  3 beta_status_churchYes~  0.668 0.0690  0.533   0.802    1.00   16339.   15761.
##  4 beta_status_sexFemale~  0.328 0.0553  0.221   0.437    1.00   15455.   16236.
##  5 beta_status_sexFemale~  0.245 0.0515  0.145   0.347    1.00   15174.   15471.
##  6 beta_status_sexFemale~  0.289 0.0664  0.160   0.420    1.00   16695.   15644.
##  7 beta_status_status_la~  2.99  0.0754  2.84    3.14     1.00    9140.   13796.
##  8 beta_status_status_la~ -0.831 0.114  -1.06   -0.609    1.00   16223.   16131.
##  9 beta_status_status_la~  1.14  0.0817  0.976   1.30     1.00   14786.   15182.
## 10 beta_status_status_la~  0.343 0.0598  0.227   0.461    1.00   15870.   15545.
## 11 beta_status_status_la~  1.53  0.0568  1.42    1.64     1.00   10681.   14652.
## 12 beta_status_status_la~ -1.33  0.0876 -1.50   -1.16     1.00   14415.   15816.
## 13 beta_status_status_la~ -0.971 0.178  -1.33   -0.627    1.00   23608.   16304.
## 14 beta_status_status_la~ -0.289 0.143  -0.567  -0.0104   1.00   21796.   16193.
## 15 beta_status_status_la~  4.48  0.105   4.28    4.69     1.00   17461.   15183.
## 16 beta_status_status_la~ -2.90  0.0619 -3.02   -2.78     1.00   14196.   14599.
## 17 beta_status_status_la~ -1.75  0.0473 -1.84   -1.66     1.00    9481.   12730.
## 18 beta_status_status_la~ -4.72  0.115  -4.95   -4.50     1.00   24283.   15392.
## 19 corr_nu_status_alpha_~  0.340 0.241  -0.180   0.756    1.00    3912.    5461.
## 20 corr_nu_status_alpha_~  0.686 0.125   0.406   0.894    1.00    4507.    5915.
## 21 corr_nu_status_alpha_~  0.830 0.110   0.566   0.980    1.00    5406.    7654.
## 22 sigma_nu_status_alpha~  0.325 0.0675  0.181   0.446    1.00    2343.    3180.
## 23 sigma_nu_status_alpha~  0.484 0.0458  0.392   0.572    1.00    4499.    7461.
## 24 sigma_nu_status_alpha~  0.316 0.0703  0.175   0.451    1.00    4513.    6510.
```

Create function for computing transition probabilities:

```
transition_probs <- function(fit, from, church) {

d_time <- data.frame(time = 1:2)
d_id <- fit$data |>
  filter(time == 1) |>
  select(id, sex)

d_status <- data.frame(status = from, church = church)
```

```
d_new <- crossing(d_time, d_id, d_status) |>
  mutate(status = ifelse(time == 2, NA, status))

pred <- fitted(fit, newdata = d_new) |>
  filter(time == 2)

pred |>
    group_by(.draw) |>
    summarise(
        S = mean(status_fitted_S),
        LAT = mean(status_fitted_LAT),
        COH = mean(status_fitted_COH),
        MAR = mean(status_fitted_MAR)
    ) |>
    summarise(
        S_p = mean(S), S_lwr = quantile(S, 0.025), S_upr = quantile(S, 0.975),
        LAT_p = mean(LAT), LAT_lwr = quantile(LAT, 0.025), LAT_upr = quantile(LAT, 0.975),
        COH_p = mean(COH), COH_lwr = quantile(COH, 0.025), COH_upr = quantile(COH, 0.975),
        MAR_p = mean(MAR), MAR_lwr = quantile(MAR, 0.025), MAR_upr = quantile(MAR, 0.975),
    )
}
```

These take time due to the large number of posterior samples and less than optimal coding of the function above:

```
No <- rbind(
  transition_probs(fit, "S", "No"),
  transition_probs(fit, "LAT", "No"),
  transition_probs(fit, "COH", "No"),
  transition_probs(fit, "MAR", "No")
)
Yes <- rbind(
  transition_probs(fit, "S", "Yes"),
  transition_probs(fit, "LAT", "Yes"),
  transition_probs(fit, "COH", "Yes"),
  transition_probs(fit, "MAR", "Yes")
)
```

```
print(No, width = Inf)
```

```
## # A tibble: 4 x 12
##      S_p   S_lwr  S_upr   LAT_p LAT_lwr LAT_upr   COH_p COH_lwr COH_upr   MAR_p
##    <dbl>   <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 0.773  0.758   0.787  0.166   0.154    0.178  0.0532  0.0476  0.0592  0.00843
## 2 0.127  0.117   0.137  0.631   0.613    0.648  0.205   0.192   0.219   0.0377
## 3 0.0361 0.0315  0.0412 0.0180  0.0150   0.0213 0.814   0.800   0.828   0.131
## 4 0.0101 0.00822 0.0121 0.00851 0.00691  0.0103 0.00465 0.00342 0.00608 0.977
##   MAR_lwr MAR_upr
##     <dbl>   <dbl>
## 1 0.00671  0.0104
## 2 0.0329   0.0428
## 3 0.120    0.143
## 4 0.974    0.980
```

```
print(Yes, width = Inf)
```

```
## # A tibble: 4 x 12
##       S_p   S_lwr   S_upr   LAT_p LAT_lwr LAT_upr   COH_p COH_lwr COH_upr  MAR_p
##     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>  <dbl>
## 1 0.747   0.729   0.764   0.191   0.177   0.206   0.0462  0.0408  0.0520  0.0158
## 2 0.111   0.101   0.122   0.662   0.644   0.680   0.162   0.150   0.175   0.0642
## 3 0.0344  0.0295  0.0398  0.0206  0.0170  0.0245  0.703   0.679   0.726   0.242
## 4 0.00524 0.00421 0.00641 0.00531 0.00427 0.00648 0.00217 0.00158 0.00286 0.987
##   MAR_lwr MAR_upr
##     <dbl>   <dbl>
## 1  0.0126  0.0194
## 2  0.0564  0.0723
## 3  0.221   0.265
## 4  0.985   0.989
```

Compare to the conditional transitions matrices computed from the data:

```
Yes_obs <- matrix(
  d |> filter(church == "Yes") |>
    group_by(id) |>
    mutate(lag_status = lag(status)) |>
    filter(!is.na(lag_status)) |>
    group_by(lag_status, status) |>
    summarise(transition_count = n()) |>
    mutate(p = transition_count / sum(transition_count)) |>
    select(lag_status, status, p) |>
    pull(p),
  4, 4, TRUE, list(c("S", "LAT", "COH","MAR"), c("S", "LAT", "COH","MAR"))
)
No_obs <- matrix(
  d |> filter(church == "No") |>
    group_by(id) |>
    mutate(lag_status = lag(status)) |>
    filter(!is.na(lag_status)) |>
    group_by(lag_status, status) |>
    summarise(transition_count = n()) |>
    mutate(p = transition_count / sum(transition_count)) |>
    select(lag_status, status, p) |>
    pull(p),
  4, 4, TRUE, list(c("S", "LAT", "COH","MAR"), c("S", "LAT", "COH","MAR"))
)
No_obs
```

```
##              S         LAT         COH         MAR
## S   0.80107790 0.137677609 0.053731831 0.007512657
## LAT 0.11612903 0.660903226 0.189161290 0.033806452
## COH 0.03469975 0.017259978 0.822186264 0.125854009
## MAR 0.01032876 0.008943192 0.005038418 0.975689633
```

```
Yes_obs
```

```
##               S         LAT         COH        MAR
## S   0.780854907 0.172185430 0.032209512 0.01475015
## LAT 0.100000000 0.674144487 0.158555133 0.06730038
## COH 0.029582929 0.019883608 0.744907856 0.20562561
```

```
## MAR 0.003856592 0.004713612 0.001428367 0.99000143
```

We can assess the difference of these matrices for example by comparing the corresponding stationary distributions, although their interpretability is limited as the true partnership-formation process is naturally nonstationary:

```
library(expm)

round((as.matrix(Yes[, seq(1, ncol(Yes), by = 3)]) %^% 1000)[1, ], 2)
```

```
##   S_p LAT_p COH_p MAR_p
##  0.04  0.04  0.03  0.89
```

```
round((as.matrix(No[, seq(1, ncol(Yes), by = 3)]) %^% 1000)[1, ], 2)
```

```
##   S_p LAT_p COH_p MAR_p
##  0.08  0.06  0.11  0.75
```

```
round((Yes_obs %^% 1000)[1, ], 2)
```

```
##    S  LAT  COH  MAR
## 0.04 0.03 0.03 0.90
```

```
round((No_obs %^% 1000)[1, ], 2)
```

```
##    S  LAT  COH  MAR
## 0.10 0.06 0.12 0.72
```

We also tested a model where there is an interaction with `sex` and `church`:

```
# cmdstanr backend as it is currently faster than rstan with categorical responses
set.seed(1)
fit_interaction <- dynamite(
  obs(status ~ -1 + lag(status) + sex * church + random(~1), "categorical") +
    random_spec(correlated = TRUE),
    data = d, group = "id", time = "time",
  backend = "cmdstanr", parallel_chains = 4,
  iter_sampling = 5000, iter_warmup = 1000, refresh = 0,
  save_warmup = FALSE, stanc_options = list("O1"))
# this is not stored in the repo due to its size
saveRDS(fit, file = "fit_partnership_interaction.rds")
```

We see that the interaction terms are negligible:

```
as_draws(fit_interaction, parameters = "beta_status_sexFemale:churchYes") |>
  posterior::summarise_draws(
    "mean",
    "sd",
    ~quantile(.x, probs = c(0.025, 0.975)),
     "rhat", "ess_bulk", "ess_tail") |>
  print(n = Inf)
```

```
## # A tibble: 3 x 8
##   variable                    mean    sd `2.5%` `97.5%`  rhat ess_bulk ess_tail
##   <chr>                      <dbl> <dbl>  <dbl>   <dbl> <dbl>    <dbl>    <dbl>
## 1 beta_status_sexFemale:ch~  0.111  0.118 -0.125   0.340  1.00    1088.    1242.
## 2 beta_status_sexFemale:ch~  0.0613 0.105 -0.146   0.260  1.00    1033.     901.
## 3 beta_status_sexFemale:ch~ -0.108  0.133 -0.366   0.152  1.00    1076.    1208.
```

And the leave-one-out-cross-validation prefers the simpler model without the interaction:

```r
l1 <- loo(fit, thin = 10) # thin to make this less memory intensive
l2 <- loo(fit_interaction, thin = 10) # thin to make this less memory intensive
save(l1, l2, file = "partnership_loos.rda")
```

```r
loo::loo_compare(l1, l2)
```

```
##        elpd_diff se_diff
## model1  0.0       0.0
## model2 -6.8       2.6
```

References

Raab, M. & Struffolino, E. (2022). Sequence Analysis. Thousand Oaks, CA: Sage.