Social Familiarity and Reward Value Result

Haoran Wan

11 May 2023

1 Preliminaries

1.1 Clear the Console Panes and Load Packages

```
options(replace.assign = TRUE, width = 65, digits = 4, scipen = 4, fig.width = 4,
    fig.height = 4)
# Clear the workspace and console.
rm(list = ls(all.names = TRUE))
cat("\f")
```

```
# Turn off showing of significance asterisks.
options(show.signif.stars = F)
# Set the contrast option; the sum contrast is required for ANOVA.
options(contrasts = c("contr.sum", "contr.poly"))
how_long <- Sys.time()
set.seed(1222023)
library(knitr)
```

1.2 Packages

```
library(minpack.lm)
library(readr)
library(nlme)
library(scales)
library(nlstools)
library(lemon)
library(fastDummies)
library(multcomp)
library(broom)
library(modelr)
library(ggpubr)
library(tidyverse)
library(here)
```

1.3 Get the Data

```r
# Get the data from the working directory.
setwd(here("code"))
source("Function.R")
source("Data.R")
```

2 Demand Curve Analyses

The following analyses were conducted to fit the demand data with the Zero-Bounded Exponential model (Gilroy et al., 2021).

2.1 Individual Level

2.1.1 Model Fitting and Prediction

```
# Model Parameter Data Frame
r2_grp_raw <- matrix(NA, nrow = 24, ncol = 1)
par_zbm_raw <- as.data.frame(matrix(NA, nrow = 4, ncol = 30)) %>%
    `colnames<-`(c("alpha_familiar_10sec", "alpha_familiar_30sec",
        "alpha_familiar_60sec", "alpha_unfamiliar_10sec", "alpha_unfamiliar_30sec",
        "alpha_unfamiliar_60sec", "Q0_familiar_10sec", "Q0_familiar_30sec",
        "Q0_familiar_60sec", "Q0_unfamiliar_10sec", "Q0_unfamiliar_30sec",
        "Q0_unfamiliar_60sec", "EV_familiar_10sec", "EV_familiar_30sec",
        "EV_familiar_60sec", "EV_unfamiliar_10sec", "EV_unfamiliar_30sec",
        "EV_unfamiliar_60sec", "Pmax_familiar_10sec", "Pmax_familiar_30sec",
        "Pmax_familiar_60sec", "Pmax_unfamiliar_10sec", "Pmax_unfamiliar_30sec",
        "Pmax_unfamiliar_60sec", "Omax_familiar_10sec", "Omax_familiar_30sec",
        "Omax_familiar_60sec", "Omax_unfamiliar_10sec", "Omax_unfamiliar_30sec",
        "Omax_unfamiliar_60sec"))
glht_p_raw <- as.data.frame(matrix(NA, nrow = 4, ncol = 4)) %>%
    `colnames<-`(c("a: F vs. UnF", "Q0: F vs. UnF", "a: Cond", "Q0: Cond"))
# Define Linear Contrast
lc_matrix <- matrix(c(1, 1, 1, -1, -1, -1, 0, 0, 0, 0, 0, 0, 0, 0,
    0, 0, 0, 0, 1, 1, 1, -1, -1, -1), nc = 12, byrow = T)
# Model Prediction Data Frame
pred_df <- tibble(fr = rep(seq(min(dat$fr), max(dat$fr), 0.1), times = 6),
    cond = rep(c("f1", "f3", "f6", "u1", "u3", "u6"), each = length(seq(min(dat$fr),
        max(dat$fr), 0.1)))) %>%
    dummy_cols(select_columns = "cond", remove_selected_columns = T) %>%
    `colnames<-`(c("fr", "f1", "f3", "f6", "u1", "u3", "u6"))
# R2 for Demand Curve Fitting
for (x in 1:24) {
    r2_grp_raw[x, ] <- rsquare(nlsLM(lq ~ lhs(q_0) * (exp((-exp(alpha)/lhs(q_0)) *
        (q_0) * fr)), data = subset(dat, id == x), start = list(alpha = -6,
        q_0 = 50), control = list(maxfev = 100000, maxiter = 1024)),
        data = subset(dat, id == x))
}
r2_grp <- r2_grp_raw %>%
    as.data.frame() %>%
    mutate(id = 1:n()) %>%
    full_join(dat %>%
        group_by(id, cond) %>%
        summarise(across(.cols = c(pair, f, u), mean)), by = "id") %>%
    mutate(cond = case_when(cond == "10 Sec" & f == 1 ~ "R2_familiar_10sec",
        cond == "30 Sec" & f == 1 ~ "R2_familiar_30sec", cond == "60 Sec" &
            f == 1 ~ "R2_familiar_60sec", cond == "10 Sec" & u ==
            1 ~ "R2_unfamiliar_10sec", cond == "30 Sec" & u == 1 ~
            "R2_unfamiliar_30sec", cond == "60 Sec" & u == 1 ~ "R2_unfamiliar_60sec")) %>%
    select(-c(id, f, u)) %>%
```

```
    pivot_wider(names_from = cond, values_from = V1)
# Subject Level Parameter
for (x in 1:4) {
    # Model 1: Unique alpha and Q_0 parameters for each social
    # familiarity and duration.
    model1 <- nlsLM(lq ~ lhs(qf1 * f1 + qf3 * f3 + qf6 * f6 + qu1 *
        u1 + qu3 * u3 + qu6 * u6) * (exp((-exp((af1 * f1 + af3 * f3 +
        af6 * f6 + au1 * u1 + au3 * u3 + au6 * u6))/lhs(qf1 * f1 +
        qf3 * f3 + qf6 * f6 + qu1 * u1 + qu3 * u3 + qu6 * u6)) * (qf1 *
        f1 + qf3 * f3 + qf6 * f6 + qu1 * u1 + qu3 * u3 + qu6 * u6) *
        fr)), data = subset(dat, pair == x), start = list(af1 = -6,
        af3 = -6, af6 = -6, au1 = -6, au3 = -6, au6 = -6, qf1 = 50,
        qf3 = 50, qf6 = 50, qu1 = 50, qu3 = 50, qu6 = 50), control = list(maxfev = 100000,
        maxiter = 1024))
    # Model 2: Unique Q_0 parameter for each social familiarity
    # and duration.
    model2 <- nlsLM(lq ~ lhs(qf1 * f1 + qf3 * f3 + qf6 * f6 + qu1 *
        u1 + qu3 * u3 + qu6 * u6) * (exp((-exp((af * f + au * u))/lhs(qf1 *
        f1 + qf3 * f3 + qf6 * f6 + qu1 * u1 + qu3 * u3 + qu6 * u6)) *
        (qf1 * f1 + qf3 * f3 + qf6 * f6 + qu1 * u1 + qu3 * u3 + qu6 *
            u6) * fr)), data = subset(dat, pair == x), start = list(af = -6,
        au = -6, qf1 = 50, qf3 = 50, qf6 = 50, qu1 = 50, qu3 = 50,
        qu6 = 50), control = list(maxfev = 100000, maxiter = 1024))
    # Model 3: Unique alpha parameter for each social
    # familiarity and duration.
    model3 <- nlsLM(lq ~ lhs(qf * f + qu * u) * (exp((-exp((af1 *
        f1 + af3 * f3 + af6 * f6 + au1 * u1 + au3 * u3 + au6 * u6))/lhs(qf *
        f + qu * u)) * (qf * f + qu * u) * fr)), data = subset(dat,
        pair == x), start = list(af1 = -6, af3 = -6, af6 = -6, au1 = -6,
        au3 = -6, au6 = -6, qf = 50, qu = 50), control = list(maxfev = 100000,
        maxiter = 1024))
    par_zbm_raw[x, c(1:12)] <- coef(model1)
    par_zbm_raw[x, c(1:6)] <- exp(par_zbm_raw[x, c(1:6)])
    par_zbm_raw[x, c(13:18)] <- ev(par_zbm_raw[x, c(1:6)])
    new_df <- mutate(pred_df, epred = predict(model1, pred_df), pred_cons = antilog(epred),
        pred_resp = pred_cons * fr) %>%
        group_by(f1, f3, f6, u1, u3, u6) %>%
        filter(pred_resp == max(pred_resp))
    par_zbm_raw[x, c(19:24)] <- new_df$fr
    par_zbm_raw[x, c(25:30)] <- new_df$pred_resp
    glht_p_raw[x, c(1, 2)] <- tidy(glht(model1, linfct = lc_matrix,
        alternative = "two.sided", rhs = 0), test = adjusted("none"))$p.value
    glht_p_raw[x, c(3)] <- anova(model1, model2)$`Pr(>F)`[2]
    glht_p_raw[x, c(4)] <- anova(model1, model3)$`Pr(>F)`[2]
}
glht_p <- glht_p_raw %>%
    rownames_to_column() %>%
    pivot_longer(names_to = "name", values_to = "p", cols = -rowname) %>%
    mutate(p = p.adjust(p, method = "holm")) %>%
    pivot_wider(names_from = name, values_from = p) %>%
    select(-rowname)
par_zbm <- par_zbm_raw %>%
    mutate(pair = c(1:4)) %>%
```

```
    relocate(pair) %>%
    full_join(r2_grp, by = "pair")

print(par_zbm)  # Demand Curve Parameters

##   pair alpha_familiar_10sec alpha_familiar_30sec
## 1    1             0.011698             0.004716
## 2    2             0.002339             0.003273
## 3    3             0.003048             0.002530
## 4    4             0.001308             0.001132
##   alpha_familiar_60sec alpha_unfamiliar_10sec
## 1             0.006150              0.0028245
## 2             0.001860              0.0019095
## 3             0.003174              0.0007607
## 4             0.002306              0.0006085
##   alpha_unfamiliar_30sec alpha_unfamiliar_60sec
## 1               0.015084               0.005900
## 2               0.002840               0.001865
## 3               0.001427               0.001007
## 4               0.001579               0.001232
##   Q0_familiar_10sec Q0_familiar_30sec Q0_familiar_60sec
## 1             47.95             55.10             57.83
## 2             68.03             94.05             82.28
## 3             51.31            126.63             28.95
## 4             39.09             25.02             18.01
##   Q0_unfamiliar_10sec Q0_unfamiliar_30sec Q0_unfamiliar_60sec
## 1               46.34               53.98               35.18
## 2               75.03               65.05               79.15
## 3               93.27               47.53              144.51
## 4               73.80               58.54               63.73
##   EV_familiar_10sec EV_familiar_30sec EV_familiar_60sec
## 1            0.8548             2.120             1.626
## 2            4.2759             3.056             5.377
## 3            3.2805             3.952             3.150
## 4            7.6431             8.835             4.337
##   EV_unfamiliar_10sec EV_unfamiliar_30sec EV_unfamiliar_60sec
## 1               3.540              0.6629               1.695
## 2               5.237              3.5212               5.362
## 3              13.146              7.0058               9.933
## 4              16.433              6.3338               8.116
##   Pmax_familiar_10sec Pmax_familiar_30sec Pmax_familiar_60sec
## 1                 1.1                 2.4                 1.7
## 2                 3.8                 1.9                 3.8
## 3                 4.0                 1.8                 7.2
## 4                12.5                24.0                17.1
##   Pmax_unfamiliar_10sec Pmax_unfamiliar_30sec
## 1                   4.8                   1.0
## 2                   4.1                   3.3
## 3                   8.2                   9.2
## 4                  13.2                   6.6
##   Pmax_unfamiliar_60sec Omax_familiar_10sec Omax_familiar_30sec
## 1                   3.1               15.99               39.42
## 2                   4.0               78.81               55.66
## 3                   3.9               61.18               71.31
```

```
## 4                   7.7              144.38              170.81
##   Omax_familiar_60sec Omax_unfamiliar_10sec
## 1               30.17                 66.34
## 2               98.41                 96.16
## 3               60.43                239.53
## 4               85.28                301.94
##   Omax_unfamiliar_30sec Omax_unfamiliar_60sec R2_familiar_10sec
## 1                 12.02                 32.19            0.9846
## 2                 65.01                 98.27            0.8678
## 3                131.11                178.57            0.8150
## 4                117.46                149.99            0.9679
##   R2_familiar_30sec R2_familiar_60sec R2_unfamiliar_10sec
## 1            0.9715            0.8716              0.8533
## 2            0.7346            0.9828              0.9693
## 3            0.7495            0.9807              0.9803
## 4            0.8255            0.8919              0.9377
##   R2_unfamiliar_30sec R2_unfamiliar_60sec
## 1              0.9021              0.9768
## 2              0.8790              0.9927
## 3              0.8748              0.9796
## 4              0.9110              0.9019
```

```
print(glht_p)  # Linear Contrast Outputs
```

```
## # A tibble: 4 x 4
##   `a: F vs. UnF` `Q0: F vs. UnF` `a: Cond` `Q0: Cond`
##            <dbl>           <dbl>     <dbl>      <dbl>
## 1      1                  1          0.232          1
## 2      1                  1          1              1
## 3      0.0000864          1          1              1
## 4      0.611              0.0449     0.112          1
```

2.2 Group Level Parameter Comparison

The following analyses were conducted to compare the values of demand curve parameters across different conditions (social duration and social familarity) using linear contrasts.

2.2.1 Social Familiarity

```
lc_fmlr <- nlsLM(lq ~ lhs(qf * f + qu * u) * (exp((-exp((af * f +
    au * u))/lhs(qf * f + qu)) * (qf * f + qu) * fr)), data = dat %>%
    select(-cond) %>%
    group_by(pair, fmlr, fr) %>%
    summarise_all(mean), start = list(af = -6, au = -6, qf = 50, qu = 50))
summary(glht(lc_fmlr, linfct = matrix(c(1, -1, 0, 0, 0, 0, 1, -1),
    nc = 4, byrow = T) %>%
    `rownames<-`(c("familiarity: alpha", "familiarity: Q0")), alternative = "two.sided",
    rhs = 0), test = adjusted("none"))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
```

```
## Fit: nlsLM(formula = lq ~ lhs(qf * f + qu * u) * (exp((-exp((af *
##     f + au * u))/lhs(qf * f + qu)) * (qf * f + qu) * fr)), data = dat %>%
##     select(-cond) %>% group_by(pair, fmlr, fr) %>% summarise_all(mean),
##     start = list(af = -6, au = -6, qf = 50, qu = 50), algorithm = "LM",
##     control = list(maxiter = 50, tol = 0.00001, minFactor = 0.0009765625,
##         printEval = FALSE, warnOnly = FALSE, scaleOffset = 0,
##         nDcentral = FALSE), trace = FALSE)
##
## Linear Hypotheses:
##                           Estimate Std. Error z value Pr(>|z|)
## familiarity: alpha == 0  -0.0794     0.2138   -0.37     0.71
## familiarity: Q0 == 0    -12.4775    17.3012   -0.72     0.47
## (Adjusted p values reported -- none method)
```

2.2.2 Social Duration

```
model1 <- nlsLM(lq ~ lhs(q1 * cond_f1 + q3 * cond_f2 + q6 * cond_f3) *
    (exp((-exp((a1 * cond_f1 + a3 * cond_f2 + a6 * cond_f3))/lhs(q1 *
        cond_f1 + q3 * cond_f2 + q6 * cond_f3)) * (q1 * cond_f1 +
        q3 * cond_f2 + q6 * cond_f3) * fr)), data = dat %>%
    select(-fmlr) %>%
    mutate(cond = case_when(cond == "10 Sec" ~ "f1", cond == "30 Sec" ~
        "f2", cond == "60 Sec" ~ "f3")) %>%
    dummy_cols(select_columns = c("cond")) %>%
    group_by(pair, cond, fr) %>%
    summarise_all(mean), start = list(a1 = -6, a3 = -6, a6 = -6, q1 = 50,
    q3 = 50, q6 = 50), control = list(maxfev = 100000, maxiter = 1024))
model2 <- nlsLM(lq ~ lhs(q) * (exp((-exp((a1 * cond_f1 + a3 * cond_f2 +
    a6 * cond_f3))/lhs(q)) * (q) * fr)), data = dat %>%
    select(-fmlr) %>%
    mutate(cond = case_when(cond == "10 Sec" ~ "f1", cond == "30 Sec" ~
        "f2", cond == "60 Sec" ~ "f3")) %>%
    dummy_cols(select_columns = c("cond")) %>%
    group_by(pair, cond, fr) %>%
    summarise_all(mean), start = list(a1 = -6, a3 = -6, a6 = -6, q = 50),
    control = list(maxfev = 100000, maxiter = 1024))
model3 <- nlsLM(lq ~ lhs(q1 * cond_f1 + q3 * cond_f2 + q6 * cond_f3) *
    (exp((-exp((a))/lhs(q1 * cond_f1 + q3 * cond_f2 + q6 * cond_f3)) *
        (q1 * cond_f1 + q3 * cond_f2 + q6 * cond_f3) * fr)), data = dat %>%
    select(-fmlr) %>%
    mutate(cond = case_when(cond == "10 Sec" ~ "f1", cond == "30 Sec" ~
        "f2", cond == "60 Sec" ~ "f3")) %>%
    dummy_cols(select_columns = c("cond")) %>%
    group_by(pair, cond, fr) %>%
    summarise_all(mean), start = list(a = -6, q1 = 50, q3 = 50, q6 = 50),
    control = list(maxfev = 100000, maxiter = 1024))
# Alpha
anova(model1, model3)

## Analysis of Variance Table
##
## Model 1: lq ~ lhs(q1 * cond_f1 + q3 * cond_f2 + q6 * cond_f3) * (exp((-exp((a1 * cond_f1 + a3 * cond_
```

```
## Model 2: lq ~ lhs(q1 * cond_f1 + q3 * cond_f2 + q6 * cond_f3) * (exp((-exp((a))/lhs(q1 * cond_f1 + q3
##   Res.Df Res.Sum Sq Df Sum Sq F value Pr(>F)
## 1     67       5.82
## 2     69       6.08 -2 -0.264    1.52   0.23

# Q0
anova(model1, model2)

## Analysis of Variance Table
##
## Model 1: lq ~ lhs(q1 * cond_f1 + q3 * cond_f2 + q6 * cond_f3) * (exp((-exp((a1 * cond_f1 + a3 * cond_
## Model 2: lq ~ lhs(q) * (exp((-exp((a1 * cond_f1 + a3 * cond_f2 + a6 * cond_f3))/lhs(q)) * (q) * fr))
##   Res.Df Res.Sum Sq Df  Sum Sq F value Pr(>F)
## 1     67       5.82
## 2     69       5.86 -2 -0.0378    0.22   0.81
```

```r
# Get system details.
S <- benchmarkme::get_sys_details()
```

```
## Loading required package:  benchmarkme
```

```r
GB <- memuse::Sys.meminfo()
```

The current machine uses the following CPU: Apple M1, with 8 cores and 16.000 GiB of RAM.

```r
sessionInfo()
```

```
## R version 4.3.3 (2024-02-29)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Sonoma 14.3
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;  LAPACK ve
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/Chicago
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods
## [7] base
##
## other attached packages:
##  [1] benchmarkme_1.0.8 here_1.0.1        lubridate_1.9.3
##  [4] forcats_1.0.0     stringr_1.5.1     dplyr_1.1.4
##  [7] purrr_1.0.2       tidyr_1.3.1       tibble_3.2.1
## [10] tidyverse_2.0.0   ggpubr_0.6.0      ggplot2_3.5.0
## [13] modelr_0.1.11     broom_1.0.5       multcomp_1.4-25
## [16] TH.data_1.1-2     MASS_7.3-60.0.1   survival_3.5-8
## [19] mvtnorm_1.2-4     fastDummies_1.7.3 lemon_0.4.9
## [22] nlstools_2.1-0    scales_1.3.0      nlme_3.1-164
## [25] readr_2.1.5       minpack.lm_1.2-4  knitr_1.45
##
## loaded via a namespace (and not attached):
##  [1] benchmarkmeData_1.0.4 gtable_0.3.4
##  [3] xfun_0.42             rstatix_0.7.2
##  [5] lattice_0.22-6        tzdb_0.4.0
##  [7] vctrs_0.6.5           tools_4.3.3
##  [9] generics_0.1.3        parallel_4.3.3
## [11] sandwich_3.1-0        fansi_1.0.6
## [13] highr_0.10            pkgconfig_2.0.3
## [15] Matrix_1.6-5          data.table_1.15.2
## [17] lifecycle_1.0.4       compiler_4.3.3
## [19] munsell_0.5.0         codetools_0.2-19
## [21] carData_3.0-5         crayon_1.5.2
## [23] pillar_1.9.0          car_3.1-2
## [25] iterators_1.0.14      foreach_1.5.2
```

```
## [27] abind_1.4-5          tidyselect_1.2.1
## [29] stringi_1.8.3        splines_4.3.3
## [31] rprojroot_2.0.4      grid_4.3.3
## [33] colorspace_2.1-0     cli_3.6.2
## [35] magrittr_2.0.3       utf8_1.2.4
## [37] withr_3.0.0          backports_1.4.1
## [39] bit64_4.0.5          timechange_0.3.0
## [41] httr_1.4.7           bit_4.0.5
## [43] gridExtra_2.3        ggsignif_0.6.4
## [45] zoo_1.8-12           hms_1.1.3
## [47] memuse_4.2-3         evaluate_0.23
## [49] doParallel_1.0.17    rlang_1.1.3
## [51] Rcpp_1.0.12          glue_1.7.0
## [53] formatR_1.14         rstudioapi_0.15.0
## [55] vroom_1.6.5          R6_2.5.1
## [57] plyr_1.8.9
```

```r
Sys.time() - how_long
```

```
## Time difference of 1.984 secs
```