

Perceived power in punching sounds

2023-03-15

Datasets

Power.df

Columns: Participant, Rating (numeric [0,1]), Comment (string), Sound (factor), SoundGroup (factor), sID (factor, redundant with Participant), Media (factor, redundant with SoundGroup)

z.Power.all.indices.df

Columns: Sound (factor), z.RatingEst (numeric), columns 2 -> z-transformed audio features as described in the paper

Power.dataset.df

Combined dataset containing Ratings and audio features

Audio examples

The audio examples in this repository are not LUFS-normalized but preserve the original loudness differences. For the experiment, sounds were loudness-normalized according to LUFS as described in the paper.

```
# Load packages
library(readr)
library(dplyr)
library(brms)
library(ggplot2)
library(tidyr)
library(tidyverse)
library(knitr)
library(stringr)
library(rstan)
library(brms)

# Load datasets
load("zPower.all.indices.df.RData")
load("Power.df.RData")
load("Power.dataset.df.RData")

# plot raw data:
ggplot(Power.df, aes(x = reorder(Sound, -Rating, median), y = Rating, fill = Media)) + geom_boxplot() +
  xlab("") + scale_y_discrete(name="Ratings") +
  ylim(0, 1) +
  theme_set(theme_bw(14)) +
  scale_fill_brewer(type="seq", palette=8, direction = -1) +
  theme(axis.text.x = element_text(angle = 90)) +
  theme(text=element_text(size=16)) +
  #scale_x_discrete(label=abbreviate) +
  scale_x_discrete(label = function(x) stringr::str_trunc(x, 8, side="center")) +
  theme(legend.position="top")

# Gaussian model with nested Sound/SoundGroup grouping (eq. 1 in ms):
fit.media4 <- brm(Rating ~ 1 + (1|Media/Sound) + (1|sID), data=Power.df)
summary(fit.media4)
fit.media4 <- add_criterion(fit.media4, "loo") #add loo criterium to model object
Media.eff <- data.frame(coef(fit.media4)$Media) # effect of Media
Media.eff$Media <- rownames(Media.eff)
colnames(Media.eff) <- c("Estimate" , "Est.Error" , "Q2.5" , "Q97.5" , "Media")
rownames(Media.eff) <- NULL

ggplot(Media.eff, aes(x = reorder(Media, -Estimate, mean), y = Estimate)) +
  geom_point() +
  geom_errorbar(aes(ymin = Q2.5, ymax = Q97.5), width = 0.2) +
  #ggtitle("Perceived power by Media") +
  xlab("Media") +
  theme_set(theme_bw(18)) +
  theme(text=element_text(size=18)) +
  scale_x_discrete(label = function(x) stringr::str_trunc(x, 8, side="right")) +
  theme(axis.text.x = element_text(angle = 90))

# ZOIB model (eq. 2 in ms)
zoib_model2 <- bf(
  Rating ~ 1 + Sound + (1|sID),
  phi ~ Sound,
  zoi ~ 1,
  coi ~ 1,
  family = zero_one_inflated_beta()
)

zoib.fit2 <- brm(
  formula = zoib_model2,
  data = Power.df,
  cores = 4,
  file = "Power_zoib_fit2"
)

# Conditional effects
ce.zoib.fit2 <- conditional_effects(zoib.fit2, dpar = "mu")
ce.zoib2 <- ce.zoib.fit2$Sound
ce.zoib2 <- ce.zoib2 %>% mutate(Sound=fct_reorder(Sound, estimate__, .desc=TRUE))
ggplot(data=ce.zoib2, aes(x= Sound, y=estimate__)) + geom_point() +
  geom_errorbar(aes(ymin = lower__, ymax = upper__), width = 0.2) +
  #ggtitle("Perceived power") +
  xlab("Sound") +
  ylab("Rating") +
  scale_y_discrete(name="Ratings") +
  ylim(0.25, 0.75) +
  theme_set(theme_bw(14)) +
  theme(axis.text.x = element_text(angle = 90)) +
  theme(text= element_text(size=14)) +
  scale_x_discrete(label = function(x) stringr::str_trunc(x, 10, side="center"))

#hypothesis testing (example):
h1 <- c("Kung_fu_hustle_scene_1_04 - SoundKung_fu_hustle_scene_1_08" = "plogis(Intercept + SoundKung_fu_hustle_sc
ene_1_04) = plogis(Intercept + SoundKung_fu_hustle_scene_1_08)")
hypothesis(zoib.fit2, h1) #not credibly non-zero

# ZOIB model with audio features (eq. 3 in ms)
zoib_model17 <- bf(
  Rating ~ 1 + z.decay.time + z.m.spectSlope.aw + z.mfcc10_kurtosis + z.delta12_variance + (1|sID),
  phi ~ 1,
  zoi ~ 1,
  coi ~ 1,
  family = zero_one_inflated_beta()
)

zoib.fit17 <- brm(
  formula = zoib_model17,
  data = Power.dataset.df,
  cores = 4,
  file = "Power_zoib_fit17"
)

summary(zoib.fit17)

# conditional effects (fig. 5 (bottom panels in ms)):
p2 <- plot(ce.zoib.fit17,
  points = FALSE,
  point_args = list(width = .03, shape = 1, size=0.5),
  theme=theme(axis.text.x = element_text(angle = 90, size=28), text = element_text(size=28)))

posterior_samples(zoib.fit17, pars = "b_")[,1:3] %>%
  mutate_at(c("b_phi_Intercept"), exp) %>%
  mutate_at(vars(-"b_phi_Intercept"), plogis) %>%
  posterior_summary() %>%
  as.data.frame() %>%
  rownames_to_column("Parameter") %>%
  kable(digits = 2)

# Posterior samples transformed back to original scale:
Power.post.samples <- posterior_samples(zoib.fit17, pars = "b_")[,1:4] %>%
  mutate_at(c("b_phi_Intercept"), exp) %>%
  mutate_at(vars(-"b_phi_Intercept"), plogis) %>%
  posterior_summary() %>%
  as.data.frame() %>%
  rownames_to_column("Parameter") %>%
  kable(digits = 2)

# Posterior predictions
pr <- predict(zoib.fit17, summary = FALSE)
posterior_table(pr)

pp_check(zoib.fit17)

pp <- predict(zoib.fit17)
head(pp)

#mcmc-draws (fig. 5 (top) in ms)
mcmc_plot(zoib.fit17, pars = c("b_z.decay.time", "b_z.m.spectSlope.aw", "b_z.mfcc10_kurtosis", "b_z.delta12_varia
nce"), type = "hist")
#
```