Superficial face judgement

Necdet Gürkan & Yangyang Yu

Last Knitted: 2023-09-18

### Vignette Setup:

### Project/Data Title:

Deep models of superficial trait inferences

Data provided by: Jordan W. Suchow

### Project/Data Description:

The diversity of human faces and the contexts in which they appear gives rise to an expansive stimulus space over which people infer psychological traits (e.g., trustworthiness or alertness) and other attributes (e.g., age or adiposity). Machine learning methods, in particular deep neural networks, provide expressive feature representations of face stimuli, but the correspondence between these representations and various human attribute inferences is difficult to determine because the former are high-dimensional vectors produced via black box optimization algorithms. In this paper, we combined deep generative image models with over 1 million judgments to model inferences of more than 30 attributes over a comprehensive latent face space. The predictive accuracy of the model approached human interrater reliability, which simulations suggest would not have been possible with fewer faces, fewer judgments, or lower-dimensional feature representations. The model can be used to predict and manipulate inferences with respect to arbitrary face photographs or to generate synthetic photorealistic face stimuli that evoke impressions tuned along the modeled attributes.

In sum, the dataset contains 1.14 million ratings across 1000 items and 34 traits by 5,000 participants. NOTE: The trait trustworthy in the dataset was collected twice, so the trait column has 35 traits.

### Methods Description:

For the attribute model studies, we used a between-subjects design where participants evaluated faces with respect to each attribute. Participants first consented. Then they completed a preinstruction agreement to answer open-ended questions at the end of the study. In the instructions, participants were given 25 examples of face images in order to provide a sense of the diversity they would encounter during the experiment. Participants were instructed to rate a series of faces on a continuous slider scale where extremes were bipolar descriptors such as “trustworthy” to “not trustworthy.” We did not supply definitions of each attribute to participants and instead relied on participants’ intuitive notions for each.

Each participant then completed 120 trials with the single attribute to which they were assigned. One hundred of these trials displayed images randomly selected (without replacement) from the full set; the remaining 20 trials were repeats of earlier trials, selected randomly from the 100 unique trials, which we used to assess intrarater reliability. Each stimulus in the full set was judged by at least 30 unique participants.

At the end of the experiment, participants were given a survey that queried what participants believed we were assessing and asked for a self-assessment of their performance and feedback on any potential points of confusion, as well as demographic information such as age, race, and gender. Participants were given 30 min to complete the entire experiment, but most completed it in under 20 min. Each participant was paid $1.50.

### Data Location:

<https://github.com/jcpeterson/omi>

## Please set the work directory to the folder containing the scripts and data  
face\_data <- import("suchow\_data.csv")  
str(face\_data)

## 'data.frame': 1139300 obs. of 5 variables:  
## $ participant: int 1256 1256 1256 1256 1256 1256 1256 1256 1256 1256 ...  
## $ stimulus : int 63 75 73 64 54 46 23 18 74 49 ...  
## $ trait : int 1 1 1 1 1 1 1 1 1 1 ...  
## $ response : int 77 99 0 58 69 58 54 47 24 71 ...  
## $ rt : int 4413 3518 5248 4167 3703 6304 4774 4480 2974 3359 ...

### Date Published:

2022-04-15

### Dataset Citation:

Peterson, J. C., Uddenberg, S., Griffiths, T., Todorov, A., & Suchow, J. W. (2022). Deep models of superficial face judgments. Proceedings of the National Academy of Sciences (PNAS).

### Keywords:

first impressions, social perception, face perception

### Use License:

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

### Geographic Description - City/State/Country of Participants:

For the attribute model studies, we used Amazon Mechanical Turk to recruit a total of 4,157 participants across 10,974 sessions, of which 10,633 (≈ 97%) met our criteria for inclusion. Participants identified their gender as female (2,065) or male (2,053), preferred not to say (21), or did not have their gender listed as an option (18). The mean age was ∼39 y old. Participants identified their race/ethnicity as either White (2,935), Black/African American (458), Latinx/a/o or Hispanic (158), East Asian (174), Southeast Asian (71), South Asian (70), Native American/American Indian (31), Middle Eastern (12), Native Hawaiian or Other Pacific Islander (3), or some combination of two or more races/ethnicities (215). The remaining participants either preferred not to say (22) or did not have their race/ethnicity listed as an option (8). Participants were recruited from the United States.

### Column Metadata:

metadata <- import("suchow\_metadata.xlsx")  
  
flextable(metadata) %>% autofit()

| Variable Name | Variable Description | Type (numeric, character, logical, etc.) |
| --- | --- | --- |
| Participant | Unique number assigned to each participant | Numeric |
| Stimulus | Face 1 to 1004 | Numeric |
| Trait | Trait 1 to 35 | Numeric |
| Response | Rating for corresponding rating | Numeric |

### AIPE Analysis:

#### Stopping Rule

When pilot data is this large, it is important to sample a smaller subset based on what the participant might actually do in the study. We will pick 50 faces rated on 10 traits - and then select the highest and lowest variance to estimate from. This choice is somewhat arbitrary - in a real study, you could choose to use only the variables you were interested in and pick the most conservative values or simply average together estimates from all variables.

# pick random faces  
faces <- unique(face\_data$stimulus)[sample(unique(face\_data$stimulus), size = 50)]  
# pick random traits  
traits <- unique(face\_data$trait)[sample(unique(face\_data$trait), size = 10)]  
  
face\_data <- face\_data %>%   
 filter(trait %in% traits) %>%   
 filter(stimulus %in% faces)

# all SEs   
SE\_full <- tapply(face\_data$response, face\_data$trait, function (x) { sd(x)/sqrt(length(x)) })  
SE\_full

## 3 11 15 17 18 20 25 26   
## 0.5302480 0.6052025 0.6057533 0.7151215 0.5899025 0.5936983 0.8063501 0.8173728   
## 29 33   
## 0.7510497 0.6380199

## smallest variance is trait 4  
face\_data\_trait4\_sub <- subset(face\_data, trait == names(which.min(SE\_full)))  
  
## largest is trait 30  
face\_data\_trait30\_sub <- subset(face\_data, trait == names(which.max(SE\_full)))

# individual SEs for 4 trait   
SE1 <- tapply(face\_data\_trait4\_sub$response, face\_data\_trait4\_sub$stimulus, function (x) { sd(x)/sqrt(length(x)) })  
quantile(SE1, probs = .4)

## 40%   
## 3.230473

# individual SEs for 30 trait  
SE2 <- tapply(face\_data\_trait30\_sub$response, face\_data\_trait30\_sub$stimulus, function (x) { sd(x)/sqrt(length(x)) })  
  
quantile(SE2, probs = .4)

## 40%   
## 4.120649

#### Minimum Sample Size

How large does the sample have to be for 80% and 95% of the items to be below our stopping SE rule?

# sequence of sample sizes to try  
samplesize\_values <- seq(25, 100, 5)  
  
# create a blank table for us to save the values in   
sim\_table <- matrix(NA,   
 nrow = length(samplesize\_values),   
 ncol = length(unique(face\_data\_trait4\_sub$stimulus)))  
# make it a data frame  
sim\_table <- as.data.frame(sim\_table)  
  
# add a place for sample size values   
sim\_table$sample\_size <- NA  
sim\_table$var <- "response"  
  
# make a second table for the second variable  
sim\_table2 <- matrix(NA,   
 nrow = length(samplesize\_values),   
 ncol = length(unique(face\_data\_trait30\_sub$stimulus)))  
  
# make it a data frame  
sim\_table2 <- as.data.frame(sim\_table2)  
  
# add a place for sample size values   
sim\_table2$sample\_size <- NA  
sim\_table2$var <- "response"  
  
# loop over sample sizes for age and outdoor trait  
for (i in 1:length(samplesize\_values)){  
   
 # temp dataframe for age trait that samples and summarizes  
 temp7 <- face\_data\_trait4\_sub %>%   
 dplyr::group\_by(stimulus) %>%   
 dplyr::sample\_n(samplesize\_values[i], replace = T) %>%   
 dplyr::summarize(se1 = sd(response)/sqrt(length(response)))   
   
 #   
 colnames(sim\_table)[1:length(unique(face\_data\_trait4\_sub$stimulus))] <- temp7$stimulus  
 sim\_table[i, 1:length(unique(face\_data\_trait4\_sub$stimulus))] <- temp7$se1  
 sim\_table[i, "sample\_size"] <- samplesize\_values[i]  
   
 # temp dataframe for outdoor trait that samples and summarizes  
 temp35 <-face\_data\_trait30\_sub %>%   
 dplyr::group\_by(stimulus) %>%   
 dplyr::sample\_n(samplesize\_values[i], replace = T) %>%   
 dplyr::summarize(se2 = sd(response)/sqrt(length(response)))   
   
 #   
 colnames(sim\_table2)[1:length(unique(face\_data\_trait30\_sub$stimulus))] <- temp35$stimulus  
 sim\_table2[i, 1:length(unique(face\_data\_trait30\_sub$stimulus))] <- temp35$se2  
 sim\_table2[i, "sample\_size"] <- samplesize\_values[i]  
  
}

Calculate the cutoff score with information necessary for correction.

cutoff\_trait4 <- calculate\_cutoff(population = face\_data\_trait4\_sub,   
 grouping\_items = "stimulus",  
 score = "response",   
 minimum = min(face\_data\_trait4\_sub$response),  
 maximum = max(face\_data\_trait4\_sub$response))  
  
# same as above  
cutoff\_trait4$cutoff

## 40%   
## 3.230473

cutoff\_trait30 <- calculate\_cutoff(population = face\_data\_trait30\_sub,   
 grouping\_items = "stimulus",  
 score = "response",   
 minimum = min(face\_data\_trait30\_sub$response),  
 maximum = max(face\_data\_trait30\_sub$response))  
  
cutoff\_trait30$cutoff

## 40%   
## 4.120649

Trait 4 Results:

cutoff <- quantile(SE1, probs = .4)  
final\_sample <-   
 sim\_table %>%  
 pivot\_longer(cols = -c(sample\_size, var)) %>%   
 dplyr::rename(item = name, se = value) %>%   
 dplyr::group\_by(sample\_size, var) %>%   
 dplyr::summarize(percent\_below = sum(se <= cutoff)/length(unique(face\_data\_trait4\_sub$stimulus))) %>%   
 dplyr::filter(percent\_below >= .80) %>%   
 dplyr::arrange(percent\_below) %>%   
 ungroup()

## `summarise()` has grouped output by 'sample\_size'. You can override using the  
## `.groups` argument.

flextable(final\_sample %>% head()) %>% autofit()

| sample\_size | var | percent\_below |
| --- | --- | --- |
| 55 | response | 0.80 |
| 60 | response | 0.90 |
| 65 | response | 0.92 |
| 70 | response | 0.98 |
| 75 | response | 0.98 |
| 80 | response | 0.98 |

Calculate the final corrected scores:

final\_scores <- calculate\_correction(proportion\_summary = final\_sample,  
 pilot\_sample\_size = face\_data\_trait4\_sub %>%   
 group\_by(stimulus) %>%   
 summarize(sample\_size = n()) %>%   
 ungroup() %>%   
 summarize(avg\_sample = mean(sample\_size)) %>%   
 pull(avg\_sample),  
 proportion\_variability = cutoff\_trait4$prop\_var)  
  
flextable(final\_scores) %>% autofit()

| percent\_below | sample\_size | corrected\_sample\_size |
| --- | --- | --- |
| 80 | 55 | 45.70328 |
| 90 | 60 | 50.63981 |
| 90 | 60 | 50.63981 |
| 98 | 70 | 61.02163 |

Trait 30 Results:

cutoff <- quantile(SE2, probs = .4)   
final\_sample2 <-   
 sim\_table2 %>%  
 pivot\_longer(cols = -c(sample\_size, var)) %>%   
 dplyr::rename(item = name, se = value) %>%   
 dplyr::group\_by(sample\_size, var) %>%   
 dplyr::summarize(percent\_below = sum(se <= cutoff)/length(unique(face\_data\_trait30\_sub$stimulus))) %>%   
 dplyr::filter(percent\_below >= .80) %>%   
 dplyr::arrange(percent\_below) %>%   
 ungroup()

## `summarise()` has grouped output by 'sample\_size'. You can override using the  
## `.groups` argument.

flextable(final\_sample2 %>% head()) %>% autofit()

| sample\_size | var | percent\_below |
| --- | --- | --- |
| 50 | response | 0.84 |
| 55 | response | 0.90 |
| 60 | response | 0.92 |
| 65 | response | 0.98 |
| 70 | response | 1.00 |
| 75 | response | 1.00 |

Calculate the final corrected scores:

final\_scores2 <- calculate\_correction(proportion\_summary = final\_sample2,  
 pilot\_sample\_size = face\_data\_trait30\_sub %>%   
 group\_by(stimulus) %>%   
 summarize(sample\_size = n()) %>%   
 ungroup() %>%   
 summarize(avg\_sample = mean(sample\_size)) %>%   
 pull(avg\_sample),  
 proportion\_variability = cutoff\_trait30$prop\_var)  
  
flextable(final\_scores2) %>% autofit()

| percent\_below | sample\_size | corrected\_sample\_size |
| --- | --- | --- |
| 84 | 50 | 45.93775 |
| 90 | 55 | 51.50835 |
| 90 | 55 | 51.50835 |
| 98 | 65 | 62.40286 |

Based on these simulations, we can decide our minimum sample size for 80% is likely close to 46 for the trait 4 trials or 46 for the trait 30 trials, depending on rounding. We can consider only the most variant trait for power analysis since it would satisfy other traits in the dataset as well.

#### Maximum Sample Size

In this example, we could set our maximum sample size for 95% items below the criterion, which would equate to 61 for the trait 4 trials or 62 for trait 30 trials.

#### Final Sample Size

Considering both estimated traits and a smaller proportion of things to rate, we found that we could likely use samples from 40 to 60 people. Other considerations could include fatigue on the number of ratings each person has to complete.