How familiarity warps representation in face space — Analysis

Vassiki Chauhan, Maria Ida Gobbini

Experimental paradigm:

Fixation cross for either 500 or 700 ms Morphed face (10% to 90% in steps of 10%) for 1000 ms Original faces for 2AFC displayed until response

Counterbalancing parameters:

6 blocks total. 3 blocks from familiar faces, 3 blocks from unfamiliar faces

108 trials per block, each identity at each morph percentage presented 4 times in a block, 12 times over the course of the experiment

Consecutive trials from different identities, block order counterbalanced across subjects

Loading Libraries...

```
## Loading required package: ggplot2
## Loading required package: lme4
## Loading required package: Matrix
## Loading required package: car
## Loading required package: dplyr
##
## Attaching package: 'dplyr'
## The following object is masked from 'package:car':
##
##     recode
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
##
## Attaching package: 'lmerTest'
## The following object is masked from 'package:lme4':
##
##     lmer
## The following object is masked from 'package:stats':
##
##     step
```

Reaction Times

```
data <- read.csv('~/Desktop/CatPercep/Data/data.csv')

data$cond <- as.character(data$cond)

# removing outliers from reaction times
data <- filter(data, rt > 0.15 & rt < 5)

# indexing separate stimulus conditions
for (x in 1:length(data$id)){
  data$stim[x] <- paste(data$cond[x],'_',data$block[x],sep="")
}

# defining labels to flip unfamiliar-unfamiliar blocks
stim_conditions <- c("f2_unfam","m2_unfam","m1_unfam")
flip_conditions <- paste(stim_conditions, "_flip", sep="")
conditions <- data.frame(c(stim_conditions[1],stim_conditions[2],stim_conditions[3]),
                         c(stim_conditions[1],flip_conditions[2],stim_conditions[3]),
                         c(stim_conditions[1],flip_conditions[2],flip_conditions[3]),
                         c(flip_conditions[1],stim_conditions[2],flip_conditions[3]),
                         c(flip_conditions[1],flip_conditions[2],stim_conditions[3]),
                         c(flip_conditions[1],flip_conditions[2],flip_conditions[3]),
                         c(stim_conditions[1],stim_conditions[2],flip_conditions[3]),
                         c(flip_conditions[1],stim_conditions[2],stim_conditions[3]),
                         fix.empty.names = FALSE   )
conditions <- as.data.frame(t(conditions))
colnames(conditions) <- c("stim1","stim2","stim3")
rep_labels <- c("c1","c2","c3","c4","c5","c6","c7","c8")
conditions$labels <- rep_labels

# computing all possible combinations of flipped unfamiliar blocks
for (i in 1:length(stim_conditions)){
  flip_label <- flip_conditions[i]
  data_to_flip <- filter(data,stim == stim_conditions[i])
  for (j in 1:length(data_to_flip$id)){
    if (data_to_flip$ans[j] == 'perc100'){
      #data_to_flip$ans[j] = 'perc0'
      data_to_flip$ans[j] = 'perc0'
    }else{
      data_to_flip$ans[j] = 'perc100'
    }
    data_to_flip$morph[j] = 100-data_to_flip$morph[j]
    data_to_flip$famfaceloc[j] <- ifelse(data_to_flip$famfaceloc[j] == 'left','right','left')
    data_to_flip$stim[j] <- flip_label
  }
  # adding flipped trials to main data frame
  data <- bind_rows(data,data_to_flip)
}

data$acc <- ifelse((data$morph < 50 & data$famfaceloc != data$resp) |
                     (data$morph > 50 & data$famfaceloc == data$resp),1, 0)
```

```r
# subsetting correct trials
data_correct <- filter(data,acc == 1)
```

Testing different models for reaction times

```r
# we need to scale the variable morph and treat it as a continuous variable
data_correct$morph <- as.factor(data_correct$morph)
data_correct$morph_sc <- as.numeric(as.character(data_correct$morph))
morph_scaled <- scale(data_correct$morph_sc)
data_correct$morph_scale <- as.numeric(morph_scaled)

contrasts(data$block) <- contr.sum

model_1_rt <- lmer(log(rt) ~ block*morph_scale +
               (1 + block | id) +
               (1 + block | stim),
             data = data_correct, REML = F)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : unable to evaluate scaled gradient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control
## $checkConv, : Model failed to converge: degenerate Hessian with 1 negative
## eigenvalues
```

```r
model_2_rt <- lmer(log(rt)~block*morph_scale + (1+block|id) +
                   (1|stim), data = data_correct, REML = F)

anova(model_1_rt,model_2_rt)
```

```
## Data: data_correct
## Models:
## ..1: log(rt) ~ block * morph_scale + (1 + block | id) + (1 | stim)
## object: log(rt) ~ block * morph_scale + (1 + block | id) + (1 + block |
## object:     stim)
##        Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## ..1     9 2160.9 2227.2 -1071.5   2142.9
## object 11 2164.3 2245.3 -1071.2   2142.3 0.6265      2     0.7311
```

Random intercept of block with stimulus condition does not improve model fit, and our first model fails to converge.

```r
model_3_rt <- lmer(log(rt)~block*morph_scale + (1|id) +
                   (1|stim), data = data_correct, REML = F)

anova(model_2_rt,model_3_rt)
```

```
## Data: data_correct
## Models:
## ..1: log(rt) ~ block * morph_scale + (1 | id) + (1 | stim)
## object: log(rt) ~ block * morph_scale + (1 + block | id) + (1 | stim)
##        Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## ..1     7 2263.7 2315.3 -1124.9   2249.7
## object  9 2160.9 2227.2 -1071.5   2142.9 106.82      2  < 2.2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

However, the random effect of stimulus condition clearly improves our model. This will be the model we report statistics on! Refitting the model with restricted maximum likelihood and looking at which of the fixed effects are significant.

Model Output

```
model_rt <- update(model_2_rt, REML=T)
Anova(model_rt,type = 3)
```

```
## Analysis of Deviance Table (Type III Wald chisquare tests)
##
## Response: log(rt)
##                     Chisq Df Pr(>Chisq)
## (Intercept)       43.900  1  3.456e-11 ***
## block             10.495  1   0.001197 **
## morph_scale       33.094  1  8.778e-09 ***
## block:morph_scale 22.018  1  2.701e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Percentage "Identity B" Responses

```
# please clean your workspace and re enter the data!!
rm(list=ls())

data <- read.csv('~/Desktop/CatPercep/Data/data.csv')

data$cond <- as.character(data$cond)

# removing outliers again
data <- filter(data, rt > 0.15 & rt < 5)

for (x in 1:length(data$id)){
  data$stim[x] <- paste(data$cond[x],'_',data$block[x],sep="")
}

stim_conditions <- c("f2_unfam","m2_unfam","m1_unfam")
flip_conditions <- paste(stim_conditions, "_flip", sep="")
conditions <- data.frame(c(stim_conditions[1],stim_conditions[2],stim_conditions[3]),
                         c(stim_conditions[1],flip_conditions[2],stim_conditions[3]),
                         c(stim_conditions[1],flip_conditions[2],flip_conditions[3]),
                         c(flip_conditions[1],stim_conditions[2],flip_conditions[3]),
                         c(flip_conditions[1],flip_conditions[2],stim_conditions[3]),
                         c(flip_conditions[1],flip_conditions[2],flip_conditions[3]),
                         c(stim_conditions[1],stim_conditions[2],flip_conditions[3]),
                         c(flip_conditions[1],stim_conditions[2],stim_conditions[3]),
                         fix.empty.names = FALSE   )
conditions <- as.data.frame(t(conditions))
colnames(conditions) <- c("stim1","stim2","stim3")
```

```
rep_labels <- c("c1","c2","c3","c4","c5","c6","c7","c8")
conditions$labels <- rep_labels

# flipping labels
for (i in 1:length(stim_conditions)){
  flip_label <- flip_conditions[i]
  data_to_flip <- filter(data,stim == stim_conditions[i])
  for (j in 1:length(data_to_flip$id)){
    if (data_to_flip$ans[j] == 'perc100'){
      #data_to_flip$ans[j] = 'perc0'
      data_to_flip$ans[j] = 'perc0'
    }else{
      data_to_flip$ans[j] = 'perc100'
    }
    data_to_flip$morph[j] = 100-data_to_flip$morph[j]
    data_to_flip$stim[j] <- flip_label
  }
  data <- bind_rows(data,data_to_flip)
}
```

Testing different models for percentage "Identity B responses"

```
data$morph <- as.factor(data$morph)
data$block <- as.factor(data$block)

contrasts(data$block) <- contr.sum

data$morph_sc <- as.numeric(as.character(data$morph))
morph_scaled <- scale(data$morph_sc)
data$morph_scale <- as.numeric(morph_scaled)

md1 <- glmer(ans ~ block*morph_scale +
               (1 + block | id) +
               (1 + block | stim),
             family=binomial,
             data = data,
             control=glmerControl(optimizer="bobyqa"))

md2 <- glmer(ans ~ block*morph_scale +
               (1 + block | id) +
               (1 | stim),
             family=binomial,
             data = data,
             control=glmerControl(optimizer="bobyqa"))

anova(md1,md2)
```

```
## Data: data
## Models:
## md2: ans ~ block * morph_scale + (1 + block | id) + (1 | stim)
## md1: ans ~ block * morph_scale + (1 + block | id) + (1 + block | stim)
##     Df   AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
```

```
## md2  8 10189 10249 -5086.3    10173
## md1 10 10193 10268 -5086.3    10173 0.0546      2     0.9731
```

Again, the random intercept due to block does not improve model fit.

```
md3 <- glmer(ans ~ block*morph_scale +
               (1 + block | id),
             family=binomial,
             data = data,
             control=glmerControl(optimizer="bobyqa"))

anova(md2,md3)
```

```
## Data: data
## Models:
## md3: ans ~ block * morph_scale + (1 + block | id)
## md2: ans ~ block * morph_scale + (1 + block | id) + (1 | stim)
##     Df   AIC   BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
## md3  7 10221 10274 -5103.5    10207
## md2  8 10189 10249 -5086.3    10173 34.309      1  4.701e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Clearly, `md2` will be our final statistical model

Model Output

```
Anova(md2, type=3)
```

```
## Analysis of Deviance Table (Type III Wald chisquare tests)
##
## Response: ans
##                       Chisq Df Pr(>Chisq)
## (Intercept)          0.4833  1    0.48693
## block                0.4810  1    0.48799
## morph_scale       3596.5424  1    < 2e-16 ***
## block:morph_scale    4.9010  1    0.02684 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(md2)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: ans ~ block * morph_scale + (1 + block | id) + (1 | stim)
##    Data: data
## Control: glmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##  10188.7  10249.4  -5086.3  10172.7    14560
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -10.0746  -0.3503  -0.0939   0.3503  12.4854
```

```
##
## Random effects:
##  Groups Name        Variance Std.Dev. Corr
##  id     (Intercept) 0.02335  0.1528
##         block1      0.02335  0.1528   1.00
##  stim   (Intercept) 0.03323  0.1823
## Number of obs: 14568, groups:  id, 15; stim, 9
##
## Fixed effects:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -0.05580    0.08027   -0.70   0.4869
## block1             -0.05567    0.08027   -0.69   0.4880
## morph_scale         2.63320    0.04391   59.97   <2e-16 ***
## block1:morph_scale  0.09709    0.04386    2.21   0.0268 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) block1 mrph_s
## block1       0.498
## morph_scale -0.014 -0.014
## blck1:mrph_ -0.014 -0.014  0.409
```