

# How familiarity warps representation in face space — Bootstrapping

*Vassiki Chauhan, Maria Ida Gobbini*

```
require('ggplot2')

## Loading required package: ggplot2

require('dplyr')

## Loading required package: dplyr
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

## Percentage “Identity B” response descriptives

Loading data and computing all combinations of unfamiliar-unfamiliar blocks

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

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

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

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

# dataframe indexing stimulus combinations
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 unfamiliar-unfamiliar block flips and
# appending to main data frame

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)
}

```

Now we create a dataframe that contains summaries of percentage response for all combinations of unfamiliar-unfamiliar pairs of stimuli, and populate the familiar-familiar pairs with repetitions for bootstrapping

```

ids <- sort(unique(data$id))
morphs <- sort(unique(data$morph))
block <- unique(data$block)

o_id <- list()
o_morph <- list()
o_block <- list()
o_cond <- list()
o_res <- list()

index <- 1
for (i in 1:length(ids)){
  for (j in 1:length(morphs)){
    for (k in 1:length(block)){
      for (l in 1:length(conditions$stim1)){
        o_id[index] <- ids[i]
        o_morph[index] <- morphs[j]
        o_block[index] <- as.character(block[k])
        if (block[k] == "fam"){
          #o_cond[index] <- "unflipped"
          o_cond[index] <- conditions$labels[l]
          perc_resp <- filter(data, (id == ids[i] & morph == morphs[j] &
            block == block[k]))$ans
          o_res[index] <- sum(perc_resp == 'perc100')/length(perc_resp)*100
        }else{
          o_cond[index] <- conditions$labels[l]
          perc_resp <- filter(data, (id == ids[i] & morph == morphs[j] &
            (stim == conditions$stim1[l] |

```

```

stim == conditions$stim2[1] |
stim == conditions$stim3[1]))$ans
o_res[index] <- sum(perc_resp == 'perc100')/length(perc_resp)*100
}
index = index + 1
}
}
}
}

o_id <- as.data.frame(unlist(t(o_id)),row.names = NULL)
o_block <- as.data.frame(unlist(t(o_block)),row.names = NULL)
o_morph <- as.data.frame(unlist(t(o_morph)),row.names = NULL)
o_cond <- as.data.frame(unlist(t(o_cond)),row.names = NULL)
o_res <- as.data.frame(unlist(t(o_res)),row.names = NULL)

df <- cbind(o_id,o_block,o_morph,o_cond,o_res)
colnames(df) <- c("id", "block", "morph", "stim", "percentage")

```

Bootstrapping the data...

```

# reading in the responses
#15 subjects, 2 blocks, 9 morphs, 8 stim combinations gives us a list of 2160 elements
x <- arrange(df,morph,desc(block))
ps <- x$percentage
d <- matrix(ps, nrow = 240, byrow = FALSE)
unfam <- d[1:120,]
fam <- d[121:240,]
difference <- unfam-fam
alldata <- cbind(unfam,fam,difference)

bootCI <- matrix(c(rep(0, 27*10000)), ncol=27)
for (j in 1:27) {
  for (i in 1:10000) {
    bootCI[i,j] <- mean(sample(alldata[,j],
                              15, replace=TRUE)) }
}

# find confidence intervals
bootCI2 = matrix(c(rep(0,27*10000)), ncol=27)
# sort bootstrap samples
for (j in 1:27) {
  bootCI2[,j] = sort(bootCI[,j])
}

# make a matrix of means and CIs
means_CIs = matrix(c(rep(0,27*5)), ncol = 5)
for (j in 1:27) {
  means_CIs[j,1] = mean(alldata[,j])
  means_CIs[j,2] = bootCI2[250,j]
  means_CIs[j,3] = bootCI2[9751,j]
  means_CIs[j,4] = bootCI2[100,j]
  means_CIs[j,5] = bootCI2[9901,j]
}

```

```
# make a data frame with column names and row names
means_CIs_2 = data.frame(means_CIs)
colnames(means_CIs_2) = c("mean", "CI95_lower", "CI95_upper",
                          "CI99_lower", "CI99_upper")
rownames(means_CIs_2) = c("10% Stranger", "20% Stranger", "30% Stranger", "40% Stranger",
                          "50% Stranger", "60% Stranger", "70% Stranger", "80% Stranger", "90% Stranger",
                          "10% Friend", "20% Friend", "30% Friend", "40% Friend",
                          "50% Friend", "60% Friend", "70% Friend", "80% Friend", "90% Friend",
                          "10% Stranger - Friend", "20% Stranger - Friend", "30% Stranger - Friend",
                          "40% Stranger - Friend", "50% Stranger - Friend", "60% Stranger - Friend",
                          "70% Stranger - Friend", "80% Stranger - Friend", "90% Stranger - Friend")
```

Plotting the data

```
ids <- means_CIs_2
ids$block <- c(rep("Stranger",9),rep("Friend",9),rep("Stranger-Friend",9))
# Don't want to plot this condition
ids <- filter(ids,block != "Stranger-Friend")

ids$morph <- c(rep(c(10,20,30,40,50,60,70,80,90),2))
ids$Condition <- ifelse(ids$block == "Friend",'Unfamiliar A to Familiar B','Unfamiliar A to Unfamiliar B')

pd = position_dodge(width=-1.5)
plot <- ggplot(ids, aes(x=morph, y=mean, col = Condition)) +
  geom_errorbar(aes(ymin=CI95_lower, ymax=CI95_upper), width=2, position=pd) +
  geom_point(size = 3, position=pd) + ylab('% "Identity B" responses') + xlab('Morph Percentage')

p1 <- plot + scale_color_brewer(palette='Set1') + theme_bw()

p2 <- p1 + theme(axis.title = element_text(size = 15, face = "bold")) +
  theme(axis.text.x = element_text(size = 12)) +
  theme(axis.text.y = element_text(size = 12, face = "bold")) +
  theme(strip.text = element_text(size = 15, face = "bold"))

p2
```

```
## Warning: position_dodge requires non-overlapping x intervals
```

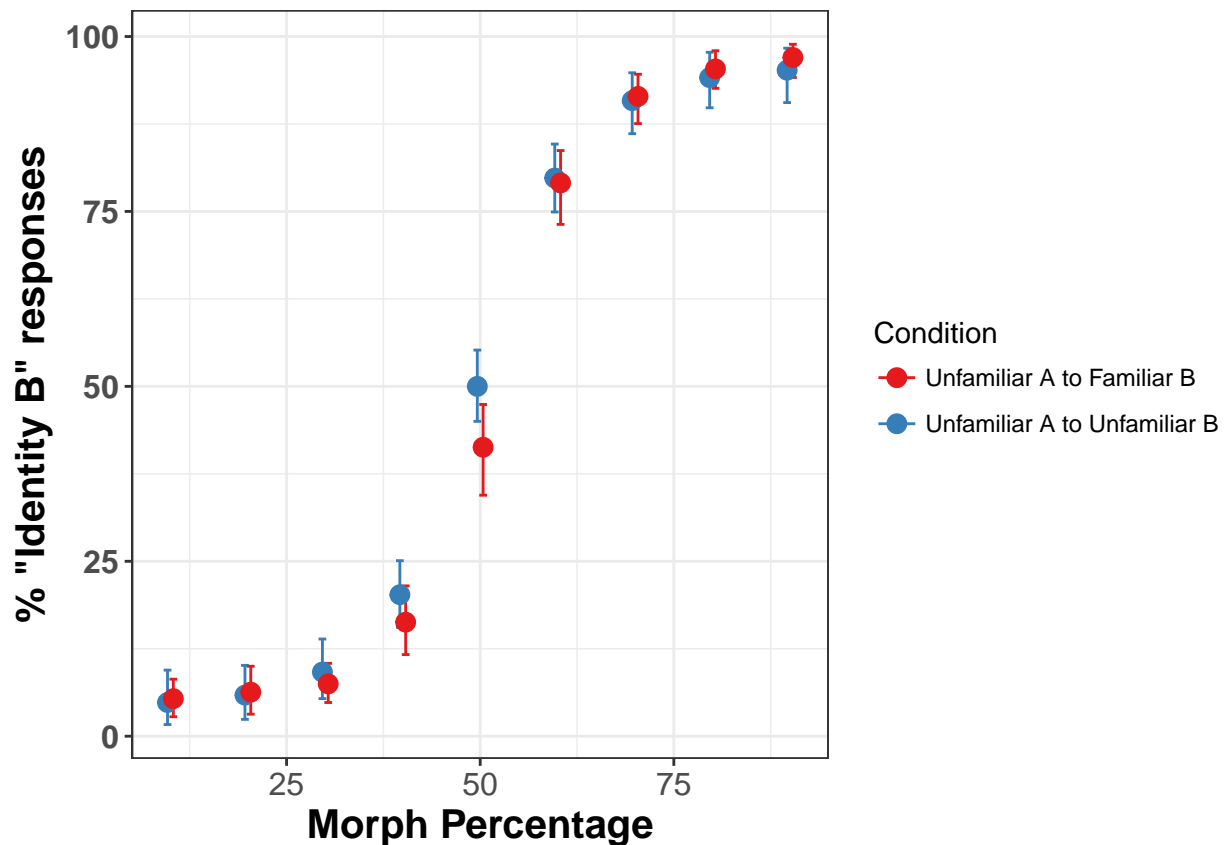


Figure 2: This figure shows morph percentage on the x axis and the mean percentage “Identity B” response on the y axis. Morph levels varied from 10% Identity B to 90% Identity B in steps of 10%. In unfamiliar-familiar blocks, “Identity B” corresponds to the familiar face. In unfamiliar-unfamiliar blocks, “Identity B” is arbitrary, and the data points were calculated by bootstrapping across all possible combinations of unfamiliar-unfamiliar morph continua. The error bars represent bootstrapped 95% confidence intervals around the means.

## Reaction Times

```
# Clear the workspace!!
rm(list=ls())
data <- read.csv('~\\Desktop\\CatPercep\\Data\\data.csv')

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

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

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$ans[j] = 'perc0'
    }else{
      data_to_flip$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
  }
  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)

data <- filter(data,acc == 1)

ids <- sort(unique(data$id))
morphs <- sort(unique(data$morph))
block <- unique(data$block)

o_id <- list()
o_morph <- list()
o_block <- list()
o_cond <- list()
o_res <- list()

index <- 1
for (i in 1:length(ids)){
  for (j in 1:length(morphs)){
    for (k in 1:length(block)){
      for (l in 1:length(conditions$stim1)){
        o_id[index] <- ids[i]
        o_morph[index] <- morphs[j]
        o_block[index] <- as.character(block[k])
        if (block[k] == "fam"){
          o_cond[index] <- "unflipped"
          # read the df, not the resp!!

```

```

    perc_resp <- filter(data, (id == ids[i] & morph == morphs[j] &
                             block == block[k]))
    o_res[index] <- mean(perc_resp$rt)
  }else{
    o_cond[index] <- conditions$labels[l]
    perc_resp <- filter(data, (id == ids[i] & morph == morphs[j] &
                              (stim == conditions$stim1[l] |
                               stim == conditions$stim2[l] |
                               stim == conditions$stim3[l])))
    o_res[index] <- mean(perc_resp$rt)
  }
  index = index + 1
}
}
}
}

o_id <- as.data.frame(unlist(t(o_id)),row.names = NULL)
o_block <- as.data.frame(unlist(t(o_block)),row.names = NULL)
o_morph <- as.data.frame(unlist(t(o_morph)),row.names = NULL)
o_cond <- as.data.frame(unlist(t(o_cond)),row.names = NULL)
o_res <- as.data.frame(unlist(t(o_res)),row.names = NULL)

df_rt <- cbind(o_id,o_block,o_morph,o_cond,o_res)
colnames(df_rt) <- c("id", "block", "morph", "stim", "rt")
# quickly convert rts to ms
df_rt$rt <- df_rt$rt*1000

```

Bootstrapping reaction times...

```

x <- arrange(df_rt,morph,desc(block))
ps <- x$rt
d <- matrix(ps, nrow = 240, byrow = FALSE)
unfam <- d[1:120,]
fam <- d[121:240,]
difference <- fam - unfam
alldata <- cbind(unfam,fam,difference)

# smaller dataframes because we don't have rts for 50% morphs - "correct" trials are ill-defined
# for this condition

bootCI <- matrix(c(rep(0, 24*10000)), ncol=24)
for (j in 1:24) {
  for (i in 1:10000) {
    bootCI[i,j] <- mean(sample(alldata[,j],
                              15, replace=TRUE)) }
}

# find confidence intervals
bootCI2 = matrix(c(rep(0,24*10000)), ncol=24)
for (j in 1:24) {
  bootCI2[,j] = sort(bootCI[,j])
}

```

```

# make a matrix of means and CIs
means_CIs = matrix(c(rep(0,24*5)), ncol = 5)
for (j in 1:24) {
  means_CIs[j,1] = mean(alldata[,j])
  means_CIs[j,2] = bootCI2[250,j]
  means_CIs[j,3] = bootCI2[9751,j]
  means_CIs[j,4] = bootCI2[100,j]
  means_CIs[j,5] = bootCI2[9901,j]
}

# make a data frame with column names and row names
means_CIs_2 = data.frame(means_CIs)

colnames(means_CIs_2) = c("mean", "CI95_lower", "CI95_upper",
                          "CI99_lower", "CI99_upper")
rownames(means_CIs_2) = c("10% Stranger", "20% Stranger", "30% Stranger", "40% Stranger",
                          "60% Stranger", "70% Stranger", "80% Stranger", "90% Stranger",
                          "10% Friend", "20% Friend", "30% Friend", "40% Friend",
                          "60% Friend", "70% Friend", "80% Friend", "90% Friend",
                          "10% Friend - Stranger", "20% Friend - Stranger", "30% Friend - Stranger",
                          "40% Friend - Stranger", "60% Friend - Stranger",
                          "70% Friend - Stranger", "80% Friend - Stranger", "90% Friend - Stranger")

```

Now plotting the reaction times for correct trials

```

ids <- means_CIs_2
ids$block <- c(rep("Stranger",8),rep("Friend",8),rep("Friend-Stranger",8))
ids <- filter(ids,block != "Friend-Stranger")

ids$morph <- c(rep(c(10,20,30,40,60,70,80,90),2))
ids$Condition <- ifelse(ids$block == "Friend", 'Unfamiliar A to Familiar B', 'Unfamiliar A to Unfamiliar B')

pd = position_dodge(width=1.5)
plot <- ggplot(ids, aes(x=morph, y=mean, col = Condition)) +
  geom_errorbar(aes(ymin=CI95_lower, ymax=CI95_upper), width=2, position=pd) +
  geom_point(size = 3, position=pd) + ylab('Reaction Times (ms)') + xlab('Morph Percentage')

p1 <- plot + scale_color_brewer(palette='Set1') + theme_bw()

p2 <- p1 + theme(axis.title = element_text(size = 15, face = "bold")) +
  theme(axis.text.x = element_text(size = 12)) +
  theme(axis.text.y = element_text(size = 12, face = "bold")) +
  theme(strip.text = element_text(size = 15, face = "bold"))

p2

```



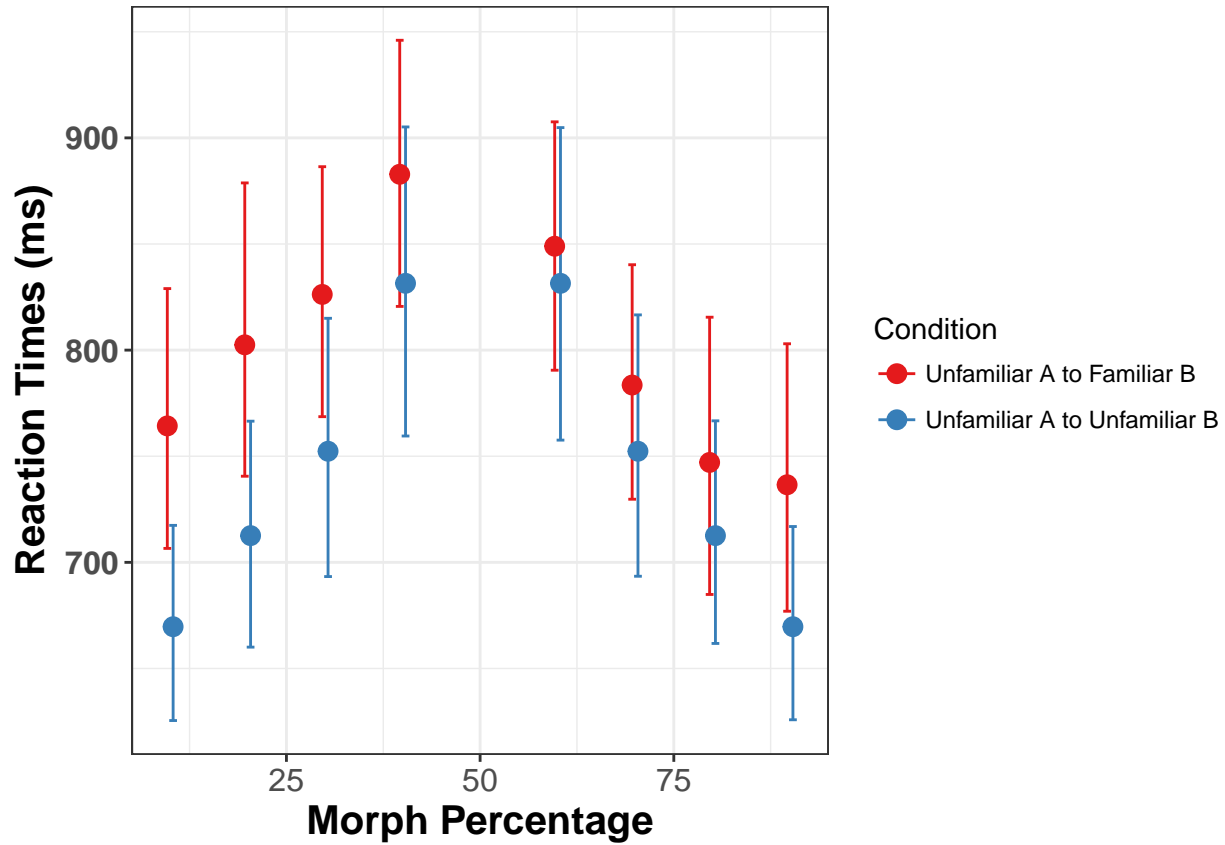


Figure 3: This figure shows morph percentage on the x axis and the mean reaction times (ms) on the y axis. Morph levels varied from 10% Identity B to 90% Identity B in steps of 10%. In unfamiliar-familiar blocks, “Identity B” corresponds to the familiar face. In unfamiliar-unfamiliar blocks, “Identity B” is arbitrary, and the data points were calculated by bootstrapping across all possible combinations of unfamiliar-unfamiliar morph continua. The error bars represent bootstrapped 95% confidence intervals around the means.