

Assignment 5 - Final Steps

Joey Heffner
Jae-Young Son
James Wilmott

Replication overview

We are replicating portions of the study ‘Maps of subjective feelings’ by Lauri Nummenmaa, Riitta Hari, Jari K. Hietanen, and Enrico Gleran. This study can be found at: <https://www.pnas.org/content/115/37/9198>. The goal of this study was to elucidate organizing principles of subjective experiences of emotions across individuals, and to uncover whether/how emotions can be grouped together to form broad categories of subjective experience.

Our previous efforts have replicated aspects of Experiment 1 and 2.

Experiment 1 replication

In Experiment 1, the authors’ specific aim was to map the basic dimensions of emotional states. To do so, the authors presented 339 volunteers with 100 words expressing core feelings and bodily processes. The volunteers were then asked to rate these words using five different scales corresponding to the following five basic dimensions:

1. bodily saliency
2. mental saliency
3. emotional intensity
4. controllability
5. lapse (relative frequency of experiencing each emotion)

After collecting the data, the authors investigated whether each basic dimension is related to the others, as visualized in Figure 1. This figure depicts the results of two analyses. First, as shown by the scatterplots, the authors used Spearman’s correlations to assess how strongly each word’s emotion ratings are associated with the corresponding ratings of all other words. Second, as shown by the histograms, the authors assessed the distributional spread of the words’ ratings for each of the five basic dimensions. For these analyses, the authors took all volunteers’ ratings, calculated the median rating for each of the 100 words, then z-scored those ratings.

Figure 1 depicts the results of Experiment 1. This figure has been included in a previous assignment (Assignment 3 - First Steps).

Experiment 2 replication

In Experiment 2, the authors assessed the similarity of emotional and bodily states across individuals. Participants were asked to make similarity judgments between pairwise combinations of 100 feelings. The authors then extracted pairwise similarities between words by determining the (scaled) Euclidean distance between each possible set of words, thereby creating an average similarity matrix across participants for each of the pairwise ratings. These similarities across participants were highly reliable (mean Spearman’s correlation split-half reliability of 0.94 across 5000 random iterations, Figure S3 in Nummenmaa et al. 2018). The mean distance matrix was computed by averaging across participants, and therefore represents the group-level degree of similarity between each of the 100 words.

Using this similarity matrix, the authors set out to determine whether unsupervised learning techniques can reveal underlying clusters of emotions, which represent structure in how people experience emotional states. To do so, they used the DBSCAN clustering algorithm (density based spatial clustering of applications with noise), which resulted in 5 distinct clusters of states, which they termed the following:

1. Positive emotions
2. Negative emotions
3. Illnesses
4. Homeostasis
5. Cognition

Other words that were not assigned to these clusters were labeled ‘Unspecific’.

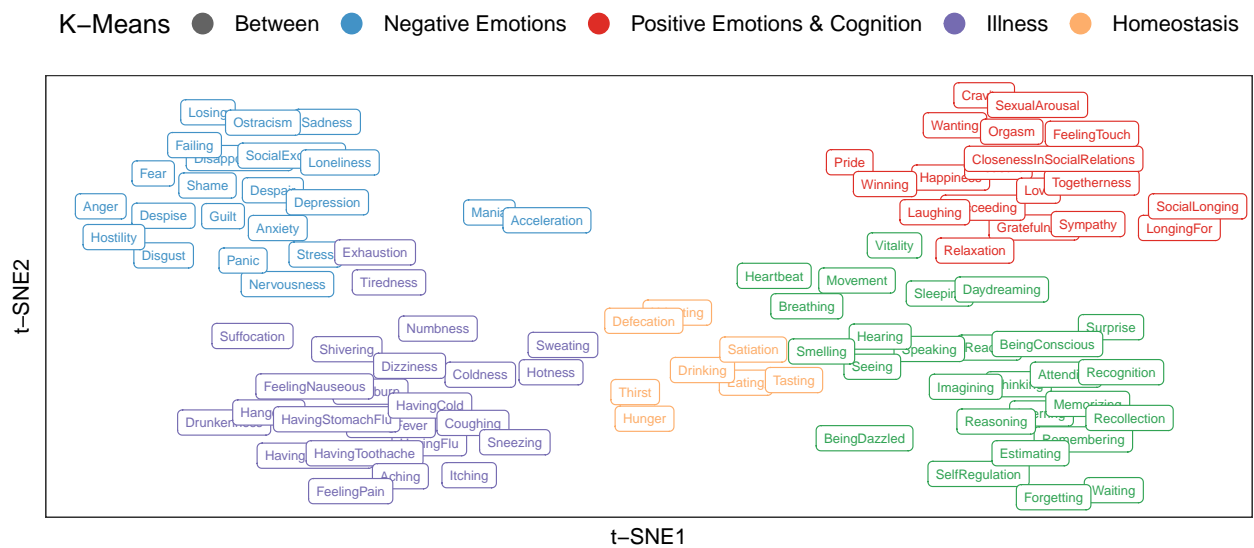
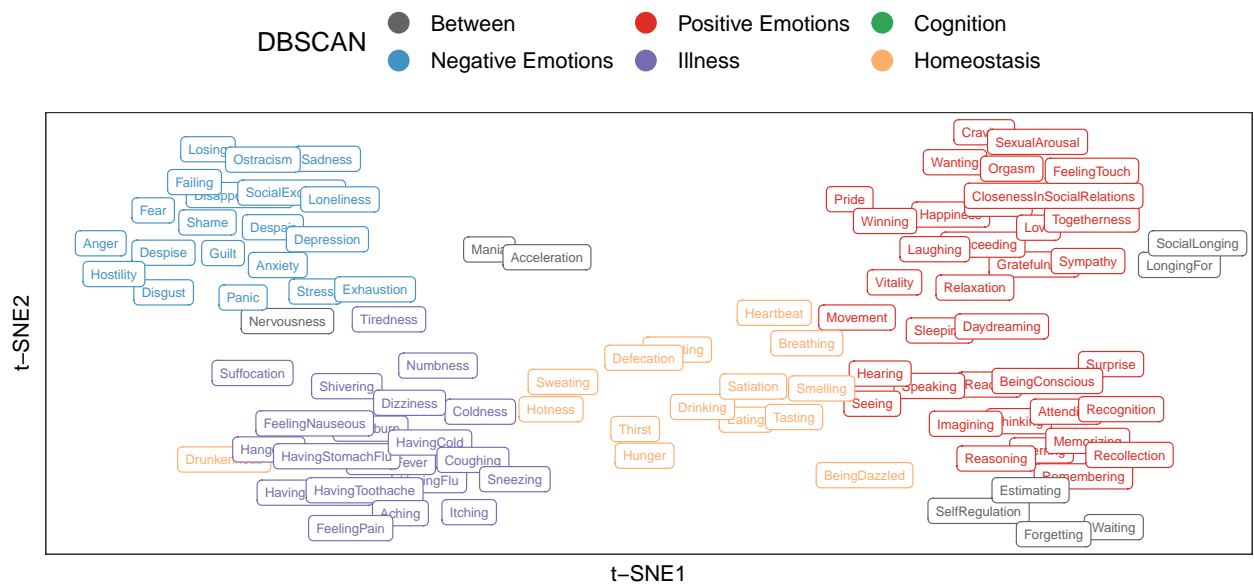
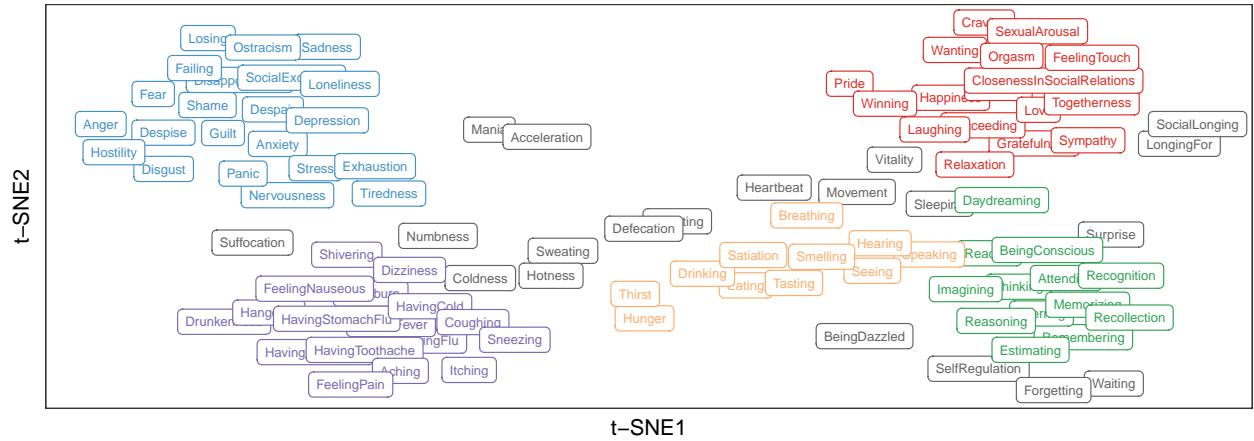
To graphically communicate how the 100 words clustered together, the authors used a dimensionality reduction technique known as t-SNE (t-distributed stochastic neighbor embedding), which is *conceptually* akin to a nonlinear PCA (principal components analysis). Briefly, the goal of t-SNE is to represent similarity/dissimilarity in high-dimensional space by placing points closer together/further apart in 2D space. Importantly, the use of t-SNE in this experiment was *not* an analytical technique, but instead a convenient method for visualizing similarity/dissimilarity in high-dimensional space. The authors then colored the points according to the clusters identified by the DBSCAN algorithm. The ultimate result of this visualization is Figure 2. Therefore, Figure 2 is a two-dimensional representation of “feeling space”, which is essentially a map between feelings arranged by t-SNE and clustered with DBSCAN.

Figure 2 depicts the results of the authors’ DBSCAN clustering analysis and t-SNE visualization. This figure has been included in a previous assignment (Assignment 4 - Second Steps).

Extension of Experiment 2

In this assignment, we are attempting to extend the analyses conducted by Nummenmaa et al. (2018) in Experiment 2. Their analysis centered around the use of a specific clustering method, DBSCAN, in determining five distinct clusters of states (listed in more detail in the above section). We sought to determine how sensitive their results are to the specific choice of analysis method by assessing how the clustering of words change when using alternative clustering techniques. Specifically, we will cluster words using two methods: K-means clustering and hierarchical clustering.

In their open access data set, Nummenmaa et al. (2018) provided clustering results from running their own K-means clustering and hierarchical clustering analyses. These analyses were not included in their published paper or supplementary materials. We use the output of these analyses to plot alternative clustering structures below.



Code Appendix

```
knitr::opts_chunk$set(echo = FALSE)

# Initialize environment
libraryBooks <- c("knitr", "tidyverse", "cowplot")
invisible(lapply(libraryBooks, require, character.only = TRUE)); rm(libraryBooks)

# Path to data
# scriptPath <- dirname(rstudioapi::getSourceEditorContext())$path
scriptPath <- getwd() # WHY does knitr complain about the interactive rstudio calls

# Graph aesthetics
pnas_theme = theme_bw(base_size = 10) +
  theme(text = element_text(size = 10), # Increase the font size
        panel.grid = element_blank(),
        axis.ticks = element_blank()) # remove x & y ticks

# Read in classification data
d1 <- read.csv(paste0(scriptPath, "/Data/Exp2Classifications.csv"),
              header = TRUE, stringsAsFactors=F) %>%
  select(-DBSCAN_type) %>%
  mutate(sensations = replace(sensations, sensations=="Being conscious", "BeingConscious"),
         sensations = replace(sensations, sensations=="Being dazzled", "BeingDazzled"),
         sensations = replace(sensations, sensations=="Closeness (in social relations)",
                              "ClosenessInSocialRelations"),
         sensations = replace(sensations, sensations=="Feeling nauseous", "FeelingNauseous"),
         sensations = replace(sensations, sensations=="Feeling pain", "FeelingPain"),
         sensations = replace(sensations, sensations=="Feeling touch", "FeelingTouch"),
         sensations = replace(sensations, sensations=="Having cold", "HavingCold"),
         sensations = replace(sensations, sensations=="Having fever", "HavingFever"),
         sensations = replace(sensations, sensations=="Having flu", "HavingFlu"),
         sensations = replace(sensations, sensations=="Having headache", "HavingHeadache"),
         sensations = replace(sensations, sensations=="Having stomach flu", "HavingStomachFlu"),
         sensations = replace(sensations, sensations=="Having toothache", "HavingToothache"),
         sensations = replace(sensations, sensations=="Longing for", "LongingFor"),
         sensations = replace(sensations, sensations=="Self-regulation", "SelfRegulation"),
         sensations = replace(sensations, sensations=="Sexual arousal", "SexualArousal"),
         sensations = replace(sensations, sensations=="Social exclusion", "SocialExclusion"),
         sensations = replace(sensations, sensations=="Social longing", "SocialLonging"))

# Read in tSNE coordinate data
d2 <- read_csv(paste0(scriptPath, "/Data/Fig2_tSNE_coords.csv")) %>%
  rename(xcoord = Var1, ycoord = Var2, sensations = Row)

# Combine classification and tSNE coordinate data
d0 <- inner_join(d1, d2) %>%
  select(-DBSCAN_labels) %>%
  rename(dbscan = DBSCAN_class,
         kmeans = KMEANS_class,
         hc = HC_class) %>%
  mutate(dbscan = as.character(dbscan),
         dbscan = recode(dbscan,
```

```

      "-1" = "Between",
      "1" = "Negative Emotions",
      "2" = "Positive Emotions",
      "3" = "Illness",
      "4" = "Cognition",
      "5" = "Homeostasis"),
  dbscan = factor(dbscan, levels=c("Between",
                                   "Negative Emotions",
                                   "Positive Emotions",
                                   "Illness",
                                   "Cognition",
                                   "Homeostasis"))) %>%

mutate(kmeans = as.character(kmeans),
       kmeans = recode(kmeans,
                        "1" = "Positive Emotions & Cognition",
                        "2" = "Illness",
                        "3" = "Negative Emotions",
                        "4" = "Between",
                        "5" = "Homeostasis"),
       kmeans = factor(kmeans, levels=c("Between",
                                         "Negative Emotions",
                                         "Positive Emotions & Cognition",
                                         "Illness",
                                         "Homeostasis"))) %>%

mutate(hc = as.character(hc),
       hc = recode(hc,
                    "1" = "Homeostasis",
                    "2" = "Cognition",
                    "3" = "Illness",
                    "4" = "Negative Emotions",
                    "5" = "Positive Emotions"),
       hc = factor(hc, levels=c("Negative Emotions",
                                 "Positive Emotions",
                                 "Illness",
                                 "Cognition",
                                 "Homeostasis")))

# Define color schemes for each clustering method
feelcolors.dbscan <- c("#636363", "#4292c6", "#de2d26", "#756bb1", "#31a354", "#fdae6b")
feelcolors.kmeans <- c("#636363", "#4292c6", "#de2d26", "#756bb1", "#fdae6b")
feelcolors.hc <- c("#4292c6", "#de2d26", "#756bb1", "#31a354", "#fdae6b")

# Plot DBSCAN
plot.dbscan <- ggplot(d0, aes(x=xcoord, y=ycoord, group=dbscan, color=dbscan)) +
  geom_point(size = 2) +
  geom_label(label=d0$sensations, nudge_x=0.25, nudge_y=0.2,
             size=2, show.legend=FALSE) +
  scale_color_manual(name="DBSCAN",
                     values=feelcolors.dbscan) +
  xlab("t-SNE1") +
  ylab("t-SNE2") +
  pnas_theme +
  guides(color = guide_legend(override.aes = list(size=4))) +

```

```

theme(legend.title = element_text(size = 12),
      legend.text = element_text(size = 10),
      axis.text=element_blank(),
      legend.position="bottom")

# Plot K-MEANS
plot.kmeans <- ggplot(d0, aes(x=xcoord, y=ycoord, group=kmeans, color=kmeans)) +
  geom_point(size = 2) +
  geom_label(label=d0$sensations, nudge_x=0.25, nudge_y=0.2,
            size=2, show.legend=FALSE) +
  scale_color_manual(name="K-Means",
                    values=feelcolors.kmeans) +
  xlab("t-SNE1") +
  ylab("t-SNE2") +
  pnas_theme +
  guides(color = guide_legend(override.aes = list(size=4))) +
  theme(legend.title = element_text(size = 12),
        legend.text = element_text(size = 10),
        axis.text=element_blank(),
        legend.position="bottom")

# Plot HIERARCHICAL CLUSTERING
plot.hc <- ggplot(d0, aes(x=xcoord, y=ycoord, group=hc, color=hc)) +
  geom_point(size = 2) +
  geom_label(label=d0$sensations, nudge_x=0.25, nudge_y=0.2,
            size=2, show.legend=FALSE) +
  scale_color_manual(name="Hierarchical Clustering",
                    values=feelcolors.hc) +
  xlab("t-SNE1") +
  ylab("t-SNE2") +
  pnas_theme +
  guides(color = guide_legend(override.aes = list(size=4))) +
  theme(legend.title = element_text(size = 12),
        legend.text = element_text(size = 10),
        axis.text=element_blank(),
        legend.position="bottom")

# Combine plots
plot_grid(plot.dbscan, plot.kmeans, plot.hc, ncol=1)

```