



# Technical University of Denmark

Course #: 02458

---

## Report #2:

## Decoding internal representations

---

*Code: <https://drive.google.com/file/d/1Y-uHNy7w0ir8vnqqnmLBwp9DEjHAu7Ci/view?usp=sharing>*

*Students:*

Tamás Rudokász, s202522  
Edison von Matt, s223534

October 16, 2022

## Contents

<b>Exercises</b>	<b>1</b>
2. Select the images that you want to use...	1
What is the feature of interest?	1
Describe in detail how you selected the images based on the feature of interest and on features of no interest.	1
4. Data pre-processing	1
Explain every step of the conversion of the raw data to the normalised pooled data.	1
Verify that normalisation worked by plotting histograms of ratings for each test person.	1
5. PCA and feature selection	2
Provide a figure of the scaled PCs added to the mean. What do the PCs represent?	2
How much variance do they encode?	3
6. Select a subset of relevant PCs.	3
Explain how your feature selection method works (your own or a routine from a toolbox).	3
8. Generate samples from the model.	4
Question: Explain in detail how you fit the model and how you use it to generate synthetic faces.	4
Explain in detail how you created the continua and show them.	5
9. Set up a second experiment	6
Analyse your results using ROC curves: Choose a baseline stimulus and fit/-plot ROC curves for all the other stimuli with respect to the baseline stimulus.	6
Do the ROC curves appear like you would expect?	7
Analyse your results using psychometric functions: fit/plot psychometric functions for each response criterion.	7
Do the psychometric functions appear like you would expect?	8

## Exercises

### 2. Select the images that you want to use...

**What is the feature of interest?**

We have chosen the strength of a smile as our feature of interest.

**Describe in detail how you selected the images based on the feature of interest and on features of no interest.**

Since we are a group of two, we decided to use the AR data set as it reduces the complexity of our project and fits well with our feature of interest.

### 4. Data pre-processing

**Explain every step of the conversion of the raw data to the normalised pooled data.**

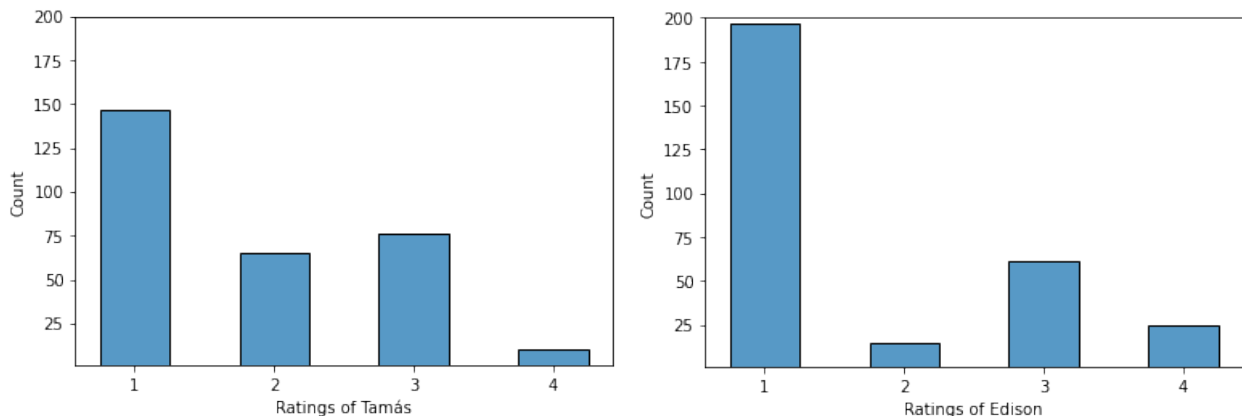
We have agreed on a scale ranging from 1 to 4, 1 being the absence of a smile, while a 4 is a full smile.

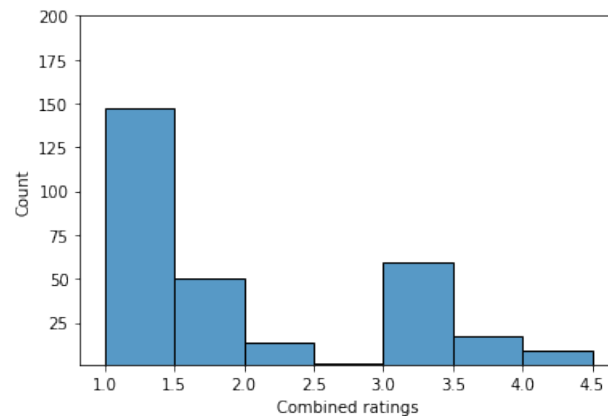
We merged our ratings by taking the average rating of each image and plotted the distribution below.

Min-max normalization was not necessary, as we have both used the full scale to rate the images - we can see, that are ratings heavily skew towards the no smile (1) rating.

**Verify that normalisation worked by plotting histograms of ratings for each test person.**

Looking at the distribution of ratings, we can see, that both of us, still perceived the images to be either just smiling or not smiling, not really capturing an even scale. In the case of Tamás the ratio of 1 and non-1 ratings is almost 1:1, whereas in the case of Edison, the ratings skew towards the rating 1.



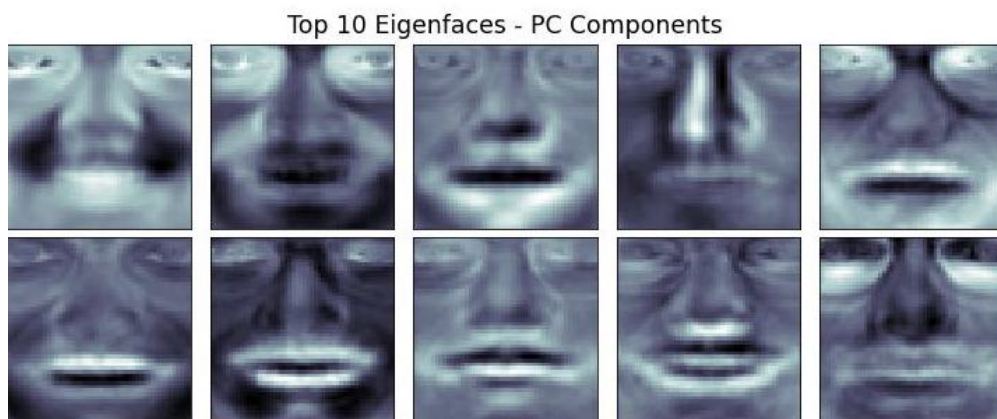


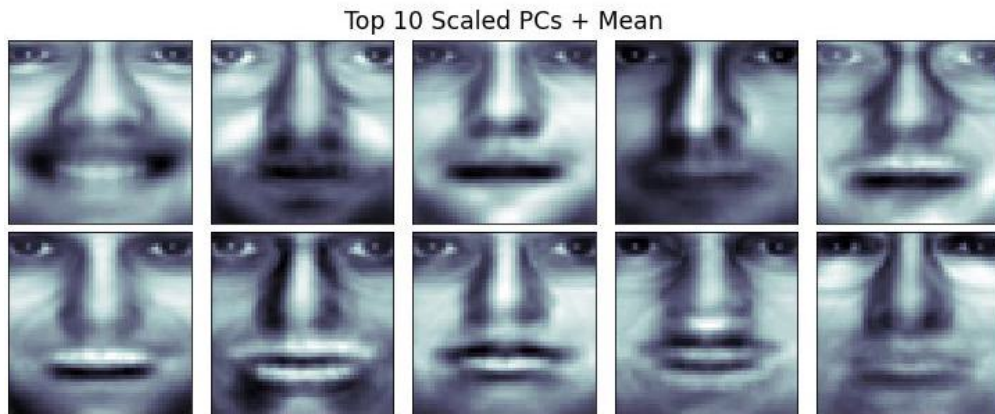
Note: bars are not centered on the values, but each bar corresponds to the value on it's lower end

## 5. PCA and feature selection

**Provide a figure of the scaled PCs added to the mean. What do the PCs represent?**

Below are plots of the top 10 raw PC components and the top 10 scaled PC components added to the mean. We can see, that the 1. PC component mainly captures the pixel variance around the teeth and eye area. In general, it captures variance in most face areas despite the nasolabial crease. The 2. PC captures the variance of the eyes and the upper nasolabial crease. The 3. PC mainly captures the variance under the nose and around the chin. The 4. PC captures the variance on the nose while the 5. PC captures the variance in the eyes, the upper lip and around the chin. The next 4 PC mainly capture the variance around the mouth area and lastly, the 10th PC captures the variance under the eyes.





How much variance do they encode?

PC	Explained variance ratio
1	0.1456
2	0.0961
3	0.0786
4	0.0613
5	0.0442
6	0.0360
7	0.0322
8	0.0262
9	0.0255
10	0.0231

In the table above, we can see how much variance of the data set is explained by each principle component. The first component explains 14.56% of the variance, the second one explains 9.61% and the third one 7.86% and so on ...

It is to note that around 117 PCs are needed to explain as much as 95% of the variance in the data set.

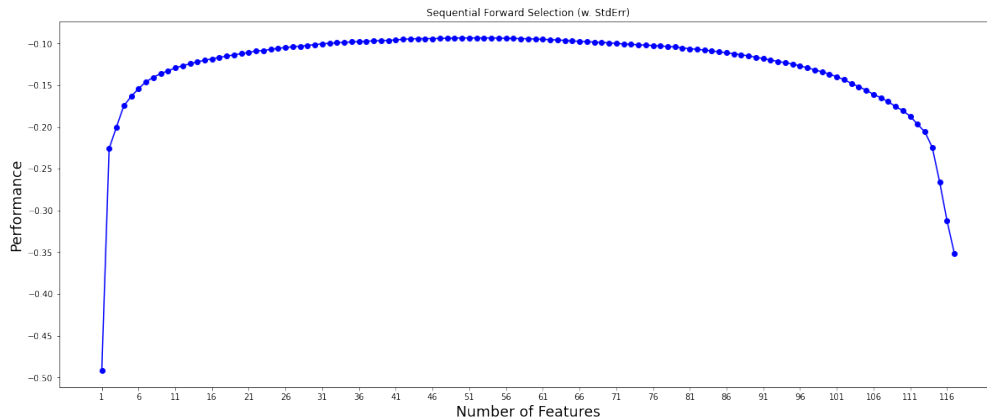
## 6. Select a subset of relevant PCs.

**Explain how your feature selection method works (your own or a routine from a toolbox).**

In order to select a relevant subset of the PCs we used the *SequentialFeatureSelector* function from the *mlxtend.feature\_selection* library in python.

We first inspected what happens if we span all features, as we can see, we get a significant performance increase if we start including handful of features, however, after a dozen or so,

we see diminishing returns. After about half of the features, we see a steady performance decline.



Because of this, we have used 5 as the number of features to select, as we tested out different combinations and this number of features gave us the best visual results to create the synthetic images. We figured that if we choose a higher number of features we run at risk of over-fitting the model.

Apart from the number of features to select, we used

- *direction* = *forward*
- *scoring* = *neg\_mean\_absolute\_error*
- *cv* = 5

as parameters for the function

The forward selection incrementally adds features to the subset of already integrated features in our linear regression model that maximize the regression performance until the feature subset has the size of the desired feature number.

The forward selection algorithm selected a subset of the following feature indexes: [0, 1, 2, 12, 16] reflecting the PC indexes that we should integrate our model to optimize the smile strength regression performance according to the collected data.

## 8. Generate samples from the model.

**Question:** Explain in detail how you fit the model and how you use it to generate synthetic faces.

Subsequently, we used the PCs with the respective indexes [x, x, x, x, x] to fit our linear regression model. For this we used our complete data set as the training data. To create and fit the model we used the *LinearRegression* library from the *sklearn.linear\_model* package in Python.

**Explain in detail how you created the continua and show them.**

As we had an initial rating scale of [1, 2, 3, 4], we used the ratings of [1, 1.5, 2, 2.5, 3, 3.5, 4] for the respective 7-step continuum. To create the images based our linear regression model we had to calculate the respective  $x$ -sample-vector given by the following equations:

$$x = \alpha\beta, \forall \alpha \in R$$

, where

$$\alpha = \frac{y_0 - \delta}{\|\beta\|^2}$$

, where  $y_0$  denotes the respective rating value we want the picture to have,  $\delta$  denotes the intercept and  $\beta$  denotes the slope of the linear regression model.

After computing the respective  $x$  vector for the synthetic image sample, we need to project it back into the original feature (pixel) space. We did this by using the *inverse\_transform(x)* function of the regression model optimized with the *SequentialFeatureSelection* from *sklearn*, which transform the sample from the feature space of the 5 selected PC subset back into the original PCA space with all 117 computed PCs. Subsequently, we need to use the *inverse\_transform(x)* again on the computed *PCA* object of *sklearn* to project the sample from the PCA space back into the original 50x50 pixel space. We repeated this process 7 times for each rating stemming from the 7 scale continuum to create the respective 7 synthetic images shown below:

**generated synthetic faces for [1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0]**



We also created synthetic faces out of the rating scale that are shown below:

**generated synthetic faces for [-1.0, 0.0, 5.0, 6.0, 12.0]**

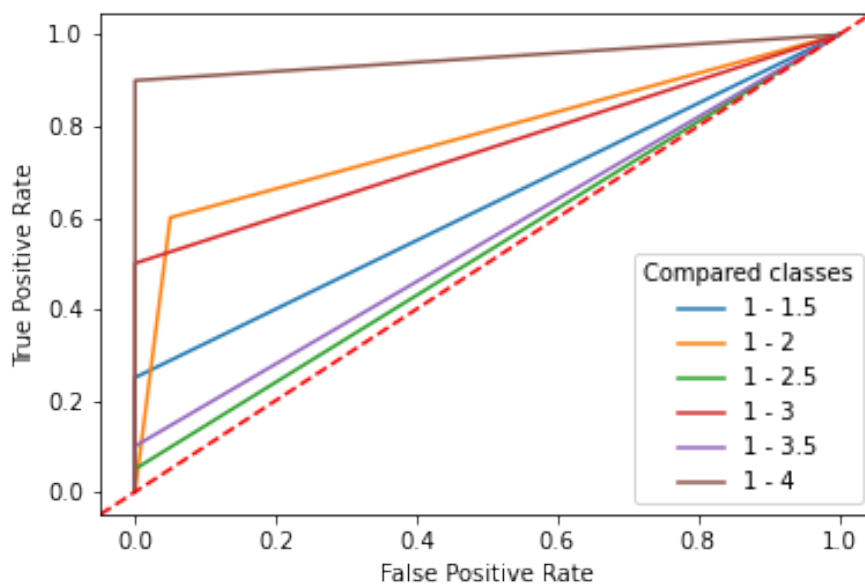


## 9. Set up a second experiment

**Analyse your results using ROC curves: Choose a baseline stimulus and fit/plot ROC curves for all the other stimuli with respect to the baseline stimulus.**

For the second experiment, we asked two friends to rate our synthetic images. We showed them each image ten times in a random order. Initially, we had a rating scale of [1, 2, 3, 4], which means that our images in the seven-step synthetic have the ratings [1, 1.5, 2, 2.5, 3, 3.5, 4] which then were the selectable rating options in our experiment.

After collecting the related data we tried to create the different ROC curves. As described in the project we fitted the ROC curves for all the other stimuli with respect to the baseline stimulus. We chose the rating [1 - no smile present] as the baseline stimulus to which we compare the results of every other rating class [1.5, 2.0, 2.5, 3, 3.5, 4]. The ROC curves are plotted below.



It is important to note that the ROC curves above are 1-threshold (1-point) ROC curves which lack important information as ROC curves with few thresholds significantly underestimate the true area under the curve. We were confused about how we should create multiple-point ROC curves given our experiment data, which consists of the binary predictions of the observer (which we retrieve by isolating a specific class stimuli with respect to the baseline stimulus) and the true binary score values of the images.

To our understanding, in order to create a multiple-point ROC curve we need to have the predicted probability values that are non-thresholded in order to yield multiple true positive rate and a false positive rate given different thresholds/criterias. But we have the binary



predictions, which have already been thresholded. Sadly, it was too late to ask about this in the lecture.

src:

<https://stats.stackexchange.com/questions/473226/how-do-i-plot-roc-curves-with-binary-predictions>

<https://stackoverflow.com/questions/66842711/plot-roc-curve-with-sklearn-for-hard-multi-class-predictions>

<https://stackoverflow.com/questions/67500856/can-you-plot-a-roc-curve-with-only-predicted-class-labels-instead-of-probabilities>

### Do the ROC curves appear like you would expect?

Disregarding the fact, that our ROC-Curves significantly underestimate the true area underneath as they only had the values for one threshold (leading to a one point curve as noted above), we can see that the classes 1 and 4 are most distinct/separated from each other, which is expected as these classes share the most different pictures in regards to smiling leading to a high true positive rate and a low false positive rate for a given threshold.

Apart from that, some ROC-Curves are indicating unexpected results. According to our ROC-curves the rating classes 1 and 1.5 seem to be more separated than the classes 1 and 3.5 and the classes 1 and 2.5, which is odd as these should share less similar looking faces (in regards to smile strength) which should lead to a higher true positive rate and a lower false positive rate.

### Analyse your results using psychometric functions: fit/plot psychometric functions for each response criterion.

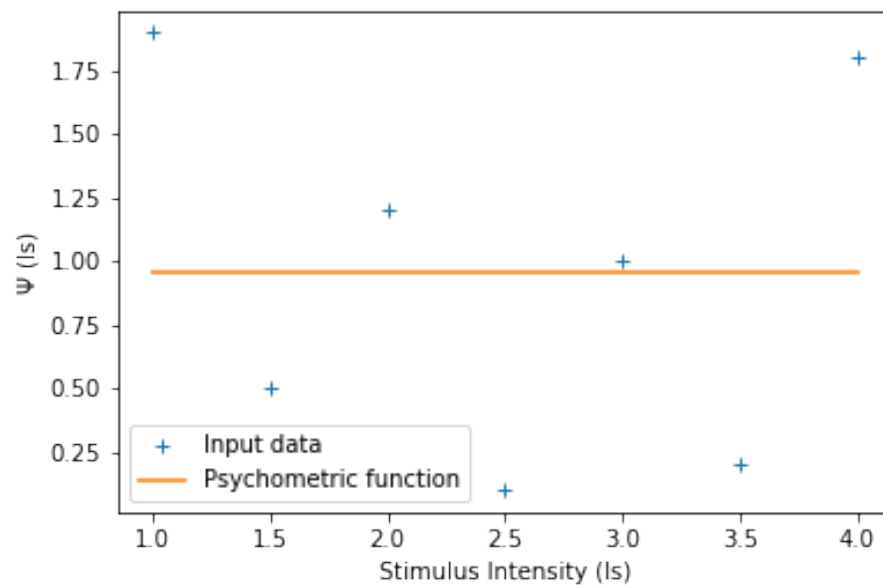
The number of correct responses for each rating class are summarized in the table below:

Stimulus intensity	1	1.5	2	2.5	3	3.5	4
Number of correct responses	19	5	12	1	10	2	18

The table makes sense, given that it should be the easiest to classify if no smile is present at all or if an extreme smile is present, while it should be hard to distinguish between a medium and a medium-strong smile (e.g. classes 2.5 and 3) leading to higher numbers of correct responses for the stimulus intensity of 1 and 4.

Subsequently we fitted the psychometric function, by minimizing the negative log likelihood for the high threshold model to retrieve the optimal  $\sigma$  and the optimal criterion  $c$  as we did in the last submission.

The results were  $\sigma = 4.07301072e + 14$  and  $c = -6.18999662e + 14$ . These extremely high values are probably wrong but we could not find out where we did a mistake as it worked in the submission before.



Do the psychometric functions appear like you would expect?

No, something went very wrong.