02458 Cognitive modelling E19

4th Assignment

Adrian Atienza (s192780)

Arianna Taormina ([arita@fysik.dtu.dk](mailto:arita@fysik.dtu.dk))

Marie Claire Capolei ([macca@elektro.dtu.dk](mailto:macca@elektro.dtu.dk))

¶The experiment in brief

During the experiment, the participant looks at a sequence of photos of faces, and for each they have to score them as Yes of No according to a binary variable that will be chosen (e.g., is the face smiling?). Several participants look at the same faces, and their binary answers are converted to a continuous scale combining the answer with the reaction time. These scores are the dependent variable in a linear model where the images themselves (or rather, their Principal Components’ scores) are the predictors.

## Setup

We choose this face database as stimuli : <https://susanqq.github.io/UTKFace/> (“Aligned&Cropped” version).

We choose to model the variable: smiling vs not smiling.

Given a wide range of images, we selected 507 of them, trying to balance the choice per gender, age, skin color and moreover excluding the ones with very different cut in prospective or not clear vision of the face.

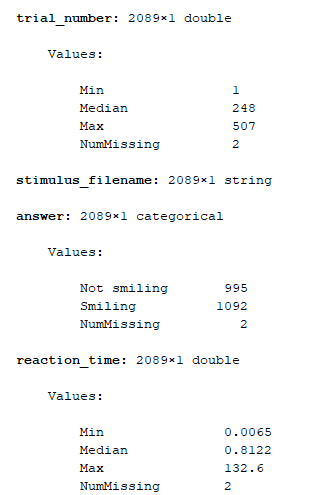
We already could see some bias, being the dataset composed by pictures downloaded from internet, it was much more common so see woman and girls smiling, while there were mainly old men on the grumpy and very grumpy side.

## Run the experiment

Each one of us performed the task of marking images as “smiley or not” twice, with different random seed, and with some time in between trials. We tried to make the choice without too much over thinking and we all committed some mistakes on the way, or sometime it took longer to decide. Finally we gathered the results into one file “”results\_tot.

## Data Pre Processing

We decided to use Matlab for the Task, loading all the data in the results\_tot as a Table.



We decided to clean the data removing outliers with reaction times below 200 milliseconds and above 2 seconds.

We want to convert the binary answers over a continuous scale. In order to do so we decided to take the variable of reaction\_time as reference. Also we decide to mark the categorical variable answer as –1 for “Not smiling” and +1 per “smiling”. This way quick answer will be close to the extremes (-1,+1), while slower answers will be more around zero.

First of all we normalize the reaction\_time per subject\_id, since each person can have different general approach to the task and we want to use all the results.

We use grpstats function to group the images in the table by name, getting a count of how many answers are valid (some could have been excluded as outliers), and to get one score out as a mean of all the peoples scores.

|  |
| --- |
|  |
| **Table. 3.1.** |

Now we have the average score for all the 507 observations, which gives us the average reaction time(slow or fast) and the “average” type of answer (smiling, not smiling)

## PCA and Features Selection

Now we want to reduce the dimensionality of the problem without losing too much information.

First we reshape the pictures to 300x300 pixels and we load each picture as a long vector (90000) in a matrix S (507x90000).

Then we apply PCA on S to get the components that retain the most of the variance.

We see that 118 first principal components are responsible for the 95% of the variance.

|  |
| --- |
|  |
| **Figure 4.1. Principal components variance** |

|  |
| --- |
|  |
| **Figure 4.2 The 118 pca describing the 95% of the variance.** |

Lets look at the first 3 reconstructed images:

|  |
| --- |
|  |
| **Figure 4.3. Image reconstruction with 118 principal components.** |

|  |
| --- |
|  |
| **Figure 4.4.** |

## Selection of a subset of relevant components

<https://se.mathworks.com/help/stats/examples/selecting-features-for-classifying-high-dimensional-data.html>

The performance on the training data (performance) is not a good estimate for a model's performance on an independent test set. To predict the performance of a selected model, we assess its performance on another data set that was not used to build the model. Here, we use cvpartition to divide data into a training set of size xxx and a test set of size of size xxx.

Without first reducing the number of features, some classification algorithms would fail since the number of features is much larger than the number of observations.

we use forward sequential feature selection to find important features. More specifically, since the typical goal of classification is to minimize the mean squared error (MSE), the feature selection procedure performs a sequential search using the MSE of the learning algorithm on each candidate feature subset as the performance indicator for that subset.

|  |
| --- |
|  |
| **Figure 5.1. The 33 selected components** |

|  |
| --- |
|  |
| **Figure 7.1. Image reconstruction with 33 principal components.** |

## Linear Model

**Write something about trained model and k components**

From the trained model we obtain the slope β (with dimension k), the intercept *d*, such that the new model y,

y= βx + d + ymean,

Where ymean is the mean face, x is the PCA score for a specific y0,

x = αβ,

where α,

α = (y0 - d)/(|| β ||2)

## Generate Samples from the Model

Figure 7.1. show the linear model output for y0 from –1 (not smiling ) to 1 (smiling).

|  |
| --- |
|  |
| **Figure 7.1. Samples generated with the linear model: from not smiling (left) to smiling (right).** |