# Final Project: The least squares face-off

Tejas Patel

December 2019

## 1 Abstract

**Out of all the final projects that sounded interested me, there was one that caught my eye in particular, "The least squares face-off". My goal was to write a function**

$$function[s, conf] = identiface(A, subj, z)$$

**using least squares regression in MATLAB. The purpose of this project was to compare a particular subject's face to a different set of group, and to test my understanding of the different methods for least square problems. The function returns two different outputs, the first output s, is the subject number that is judged to be the most likely match for z, and the second output conf is a number between 0 (lowest) and 1 (highest) that indicates the level of confidence in the identification.**

## 2 Data

**My first approach was to prepare the data. I was given a directory that contains a set of faces taken between April 1992 and April 1994 at the Olivetti Research Laboratory in Cambridge, UK. There are 10 different images of 40 distinct subjects. The files are in PGM format and has a size of 92x112, 8-bit grey levels. The images are organized in 40 directories (one for each subject) named as:**

$$sX$$

**where X indicates the subject number (between 1 and 40). In each directory there are 10 different images of the selected subject named as:**

$$Y.pgm$$

**where Y indicates which image for the specific subject (between 1 and 10).Two other functions were also given to me: loadface, where**

it took in the face for the given subject number (integer, 1 to 40) and the given pose (integer, 1 to 10). This function returns the corresponding image in column form. The other function: showfaces, takes a matrix of image vectors and displays them in a grid. With these two functions, and the directory, I now have all the necessary tools to complete my function: identiface.

# 3 Method

identiface takes in three parameters, A,subj, and z. A is a library of n identically sized pq images as columns of a single pqn matrix. subj is an n-vector that records the subject number whose image appears in the corresponding column. Lastly z, is an image in vector form, being compared to the library A.

Once the data was suitable for analysis, I wanted to solve

$$Ax = z$$

for x using linear least squares to see if z is a well represented linear combination of images in the library. I first began solving this linear squares by QR factorization

$$A = QR$$

for the matrix A. I solved the upper triangular linear system by performing the backward substitution method.

After calculating if z is a well represented as a linear combination of images in the library, I wanted to return the index where the maximum number lied in the linear combination. If this max number was indicated as a one, then the same image as z. If not one, this would indicate which image in A is most similar to z. Then I would then compare this index to the index to the vector, subj. Now that I returned the subject number, I still must return a number indicating the level of confidence in the identification.

To calculate the level of confidence in the identification I must compute a linear regression between two vectors. The first of which is the vector that the function predicts what the subject number will be. The second is simply just the input z. After calculating how similar the two images are, using linear regression, the function will finally return the last output, conf.

# 4 Results

**After writing this function, I wanted to see how my algorithm performs, so I wrote a few test cases in the file ProjectTest.m. So I began by storing a few of the different subject's images in different variables, such as:**

$$a1 = loadface(1, 4)$$

**I did this with the all 40 subjects and chose a random pose for them. I then stored just the first 20 of these variables in the matrix A. For my subj vector, it simply contains the numbers one to twenty. For the image I want to compare, it will be stored in my z vector. The first five z's are images in the matrix A and the next five for not in the matrix. If my function is done properly, The first five times the function is called, it should return exactly the subject number and a confidence number of one. For the next five times, the function should return the closest match to the corresponding z vector and return a relatively high confieced number. The following below is what was return for the subj and conf respectively**

| | | |
|---|---|---|
| identiface(A,subj,z1) | 3 | 1 |
| identiface(A,subj,z2); | 1 | 1 |
| identiface(A,subj,z3); | 15 | 1 |
| identiface(A,subj,z4); | 4 | 1 |
| identiface(A,subj,z5); | 18 | 1 |
| identiface(A,subj,z6); | 19 | 0.9598 |
| identiface(A,subj,z7); | 8 | 0.8311 |
| identiface(A,subj,z8); | 1 | 0.8024 |
| identiface(A,subj,z9); | 14 | 0.8794 |
| identiface(A,subj,z10); | 6 | 0.8846 |

**Also please view projectTest.m to see the results in more detail.**

# 5 Conclusion

**After viewing the results, I can confidently say that, my function works almost perfectly all the time. I have yet to run into a case where the function fails. Overall there was a problem and my approach was to solve it by solving linear squares by QR factorization. I would finally like to give credit to credit to the Olivetti Research Laboratory for letting me use their images.**

# 6 Sources

F. Samaria and A. Harter
"Parameterisation of a stochastic model for human face identification"
2nd IEEE Workshop on Applications of Computer Vision
December 1994, Sarasota (Florida).