COMPSCI 4ML3

Introduction to Machine Learning

Winter 2021

Assignment Two

Tahseen Ahmed

ahmedt26

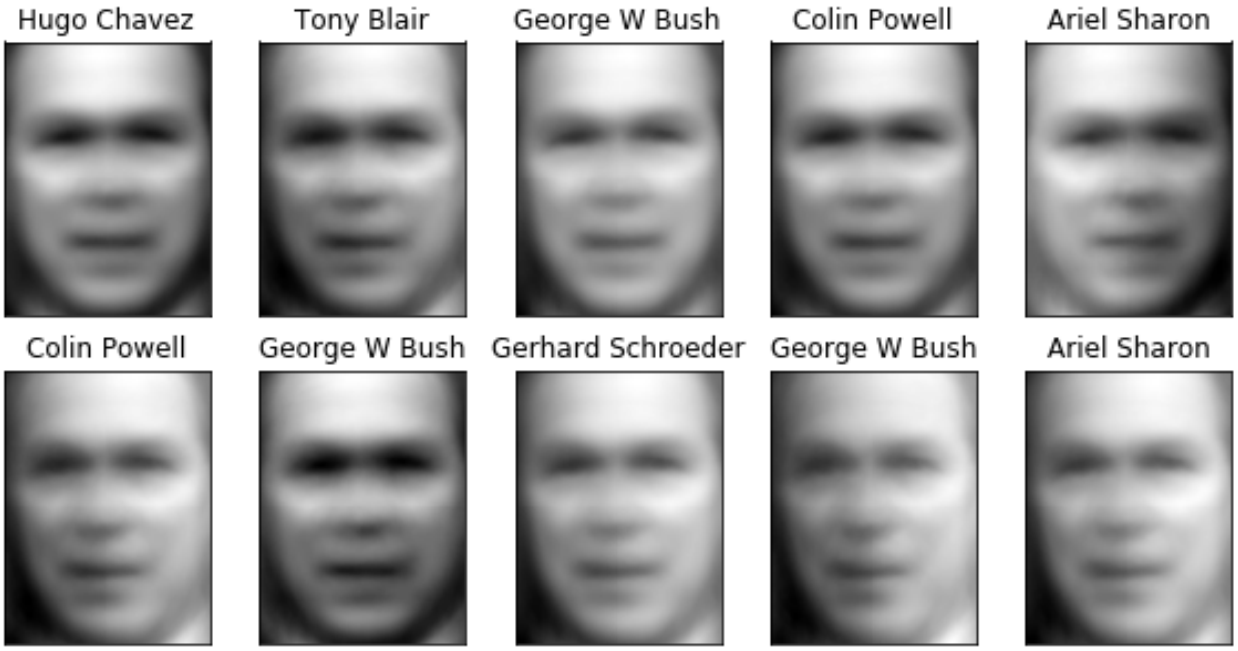
McMaster University

Revised: February 18th, 2020

Hello, and welcome to my submission for assignment two. Below you will find all of my solutions to the requested questions from the assignment instructions.

1. **Programming Component: Task One**

In this task we reconstruct the original images from a Principal Component Analysis (PCA). Here are the first ten images of the dataset using two components only:



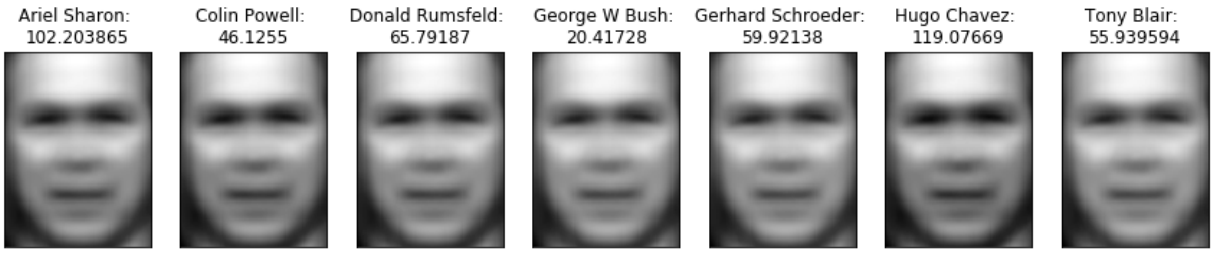
Just like the people in my dreams, these are hard to visualize as a human, and quite terrifying. Using 179 components gives us a better reconstruction. This number of components will show up in Task Two.

We have shown two reconstructions, note that more principal components result in a better reconstruction.

2. **Programming Component: Task Two**

The question asks us to get the average reconstruction error (MSE) of each face. We do this for a two-component PCA:

```
============================================================
PCA of 2 Components, MSE of each face (Original PCA):
Components Used: 2
```

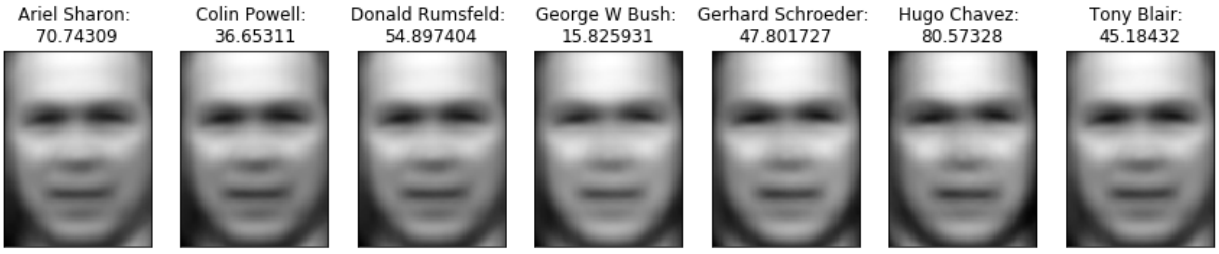


Hugo Chavez's images resulted in the great reconstruction error. The entire dataset has an MSE of 1043.5609.

The second part of this task wants the MSE to be less than 100 "for all (and each)" class. To be certain we provide the desired outcome, we give two image sets.

The first is a reconstruction using a four-component PCA, which gives us an MSE less than 100 for each class. This is the minimum number of required components for this. Here are the images:

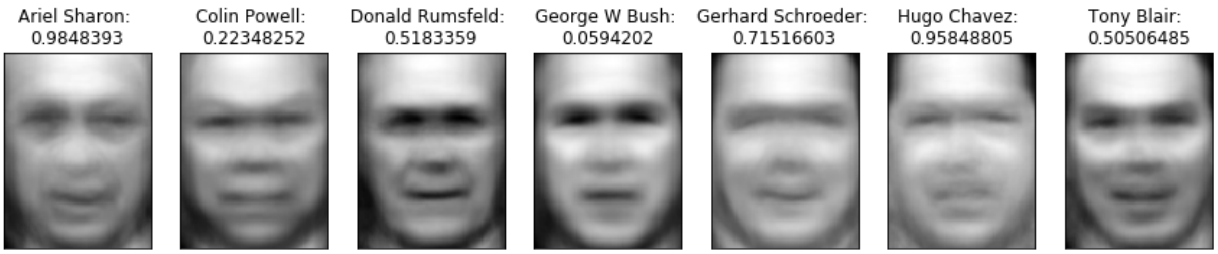
```
PCA of 4 Components, MSE of each face (PCA where each class' MSE < 100):
Components Used: 4
```



The entire dataset with a four-component PCA reconstruction has an MSE of 855.35834.

For the next set of images, we used a minimum of 179 images to get the entire set's MSE to be less than 100 (99.467766):

```
============================================================
PCA of 179 Components, MSE of each face (PCA where entire set's MSE < 100):
Components Used: 179
```



The reconstructions (as we know) get better with more componnets, and equally creepy.

3. **Programming Component: Task Three**

We take the first one-hundred images from the dataset to make a training dataset. We use this to create a PCA mapping. We then use this mapping to transform the training and test dataset (the test dataset is all the other images in the remaining set) and then reconstruct those transformations. We get the following output from my code:

```
Test and Train PCA with 2 components:
Training Error (2 components): 1012.4919
 Testing Error (2 components): 1084.1616
```

The training error is a little smaller because we applied PCA on that training dataset. The testing dataset has images different than the training so during remapping it will not perfectly realign.

For the second part of this question (using 100 components) we get this output:

```
Test and Train PCA with 100 components:
Training Error (100 components): 1.1718704e-09
 Testing Error (100 components): 370.51874
```

The training and testing error differ vastly. There is essentially no training error. This is a case of overfitting: not a lot of training data, and a lot of components (high dimensions). If we reverse the training and testing datasets (make testing training, training testing) we get this output:

```
Test and Train PCA with 100 components, using larger training set:
Training Error (Reversed) (100 components): 170.0796
 Testing Error (Reversed) (100 components): 215.44899
```

The increased number of data points gave the PCA method more room to work with, which removed the problem of overfitting. The increased number of components (as in task two) improved MSE values. Furthermore, more training points improved the testing MSE considerably as well.

In this question we are determining if drivers are high. We consider X to be the random variable for this. Note that like the coin flips, this is a Bernoulli random variable.

$$P(x = 0) = 1 - \alpha,\ P(x = 1) = \alpha$$

The test itself is also a Bernoulli random variable. We can consider random variable Y for this case.We say 1 if the test is correct, and 0 if the test is incorrect.

$$P(y = 0) = \beta,\ P(y = 1) = 1 - \beta$$

1. **Question 2A**

   We are looking at the joint probability of independent events. The formula for this is:

   $$P(A, B) = P(A) \cdot P(B)$$

   The probabiltiy table also gives us the probabilities of Type I and Type II errors caused by this test.

	Not Actually High	Actually High
Incorrect Result	False Positive: $\beta \cdot (1 - \alpha)$	False Positive: $\beta \cdot \alpha$
Correct Result	True Negative: $(1 - \beta) \cdot (1 - \alpha)$	True Positive: $(1 - \beta) \cdot \alpha$

   We apply the Sum Rule for probability to find $P(\text{positive result})$

   $$P(\text{positive result}) = P(y = 0|x = 0) + P(y = 1|x = 1) = \beta \cdot (1 - \alpha) + (1 - \beta) \cdot \alpha$$
   $$= \alpha + \beta - 2\alpha\beta$$

   We let $P(Z) = \gamma = \alpha + \beta - 2\alpha\beta$ to simplify some upcoming mathematics. We can say $Z = 1$ when there is a positive result, and $Z = 0$ when there is a negative result.

2. **Question 2B**

   The PMF for this Binomial distribution is:

   $$P(n, \gamma) = \binom{n}{y}\gamma^y(1 - \gamma)^{n-y};\ y = 1, ..., n$$

   We are looking for $y = n_+$:

   $$P(n, \gamma) = \binom{n}{n_+}\gamma^{n_+}(1 - \gamma)^{n-n_+}$$

   $$P(n, \gamma) = \frac{n!}{n_+!(n - n_+)!}(\alpha + \beta - 2\alpha\beta)^{n_+}(1 - (\alpha + \beta - 2\alpha\beta))^{n-n_+}$$

3. **Question 2C**

In this question, we know β and we treat it like a constant. We compute and simplify whatever we get from the MLE method, then we isolate and solve for α.

$$\alpha^{ML} = \mathsf{argmax}_\alpha[P(Z \mid \alpha)] = \mathsf{argmin}_\alpha[-\sum_i \mathsf{log}(P(z^i \mid \alpha))]$$

$$\mathsf{argmin}_\alpha[-\sum_i \mathsf{log}(P(z = z^i \mid \alpha))]$$

$$\mathsf{argmin}_\alpha[-\sum_{i|z^i=0} \mathsf{log}(P(z = z^i \mid \alpha)) - \sum_{i|z^i=1} \mathsf{log}(P(z = z^i \mid \alpha))]$$

$$\mathsf{argmin}_\alpha[-\sum_{i|z^i=0} \mathsf{log}((1-\gamma)) - \sum_{i|z^i=1} \mathsf{log}(\gamma)]$$

$$\mathsf{argmin}_\alpha[-n_- \mathsf{log}(1-\gamma) - n_+ \mathsf{log}(\gamma)]$$

We have discovered $f(\gamma) = -n_- \mathsf{log}(1-\gamma) - n_+ \mathsf{log}(\gamma)$. We start to minimize it:

$$\frac{\partial f(\gamma)}{\partial \alpha} = -n_-(\frac{1}{1-\gamma} \cdot -1) - n_+(\frac{1}{\gamma}) = 0$$

$$\Rightarrow \frac{n_-}{1-\gamma} = (\frac{n_+}{\gamma})$$

$$\Rightarrow \gamma = \frac{n_+}{n_- + n_+}$$

We know that $\gamma = \alpha + \beta - 2\alpha\beta$, so we substitute that in to find α:

$$\gamma = \frac{n_+}{n_- + n_+}$$

$$\alpha + \beta - 2\alpha\beta = \frac{n_+}{n_- + n_+}$$

$$\alpha - 2\alpha\beta = \frac{n_+}{n_- + n_+} - \beta$$

$$\alpha(1 - 2\beta) = \frac{n_+}{n_- + n_+} - \beta$$

$$\alpha = \frac{n_+}{(n_- + n_+) \cdot (1 - 2\beta)} - \frac{\beta}{(1 - 2\beta)}$$

$$\alpha = \frac{n_+ - \beta(n_- + n_+)}{(n_- + n_+) \cdot (1 - 2\beta)}$$

4. **Question 2D**

In case i, $\beta = 0$, the test perfecty determines whether the driver is high or not. This means that

$$\alpha = \frac{n_+ - \beta(n_- + n_+)}{(n_- + n_+) \cdot (1 - 2\beta)} \Rightarrow \frac{n_+}{n_- + n_+} = \frac{n_+}{n}$$

which is just the proportion of positive results (high drivers) against the total. This is the same as finding the bias of a coin since both are Bernoulli random variables.

For case *ii* we have a problem:

$$\alpha = \frac{n_+ - \beta(n_- + n_+)}{(n_- + n_+) \cdot (1 - 2\beta)}$$

$$\alpha = \frac{n_+ - 0.5(n_- + n_+)}{(n_- + n_+) \cdot (1 - 2 \cdot 0.5)} \Rightarrow \alpha = \frac{n_+ - 0.5(n_- + n_+)}{0} = \infty$$

The probability for α is indeterminant. It cannot be computed since the second test find the correct results exactly half of the time. As $n \to \infty$, half of the negative results and positive results are incorrect, and there is no way to tell which from which. And so, it is impossible to determine α.