

Pattern Recognition Homework (4)

Fisher LDA

Instructed by Dr.

Abbas Mehrbanian

[abbas.mrbn@gmail.com](mailto:abbas.mrbn@gmail.com), Stu ID: 40130935

Reza Tahmasebi

saeed77t@gmail.com, Stu ID: 40160957

# Fisher LDA

LDA is a widely used dimensionality reduction technique built on Fisher’s linear discriminant. These concepts are fundamentals of machine learning theory. In this article, I’ll go through an example of a classifier using Fisher’s linear discriminant and derive the optimal solution for Fisher’s criterion. Finally, I compare LDA as a dimensionality reduction technique to PCA.

Fisher’s linear discriminant can be used as a supervised learning classifier. Given labeled data, the classifier can find a set of weights to draw a decision boundary, classifying the data. Fisher’s linear discriminant attempts to find the vector that maximizes the separation between classes of the projected data. Maximizing “separation” can be ambiguous. The criteria that Fisher’s linear discriminant follows to do this is to maximize the distance of the projected means and to minimize the projected within-class variance.

# Fisher LDA implementation

To implement fisher LDA, we used the *fetch\_olivetti\_faces* function to read our data set.

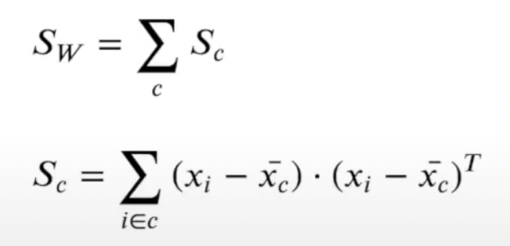
We split our images and classes, labeled them into *x*  and *y,* then passed them to the *CalculateScaters* function to calculate the scatters.

## compute\_scatters function

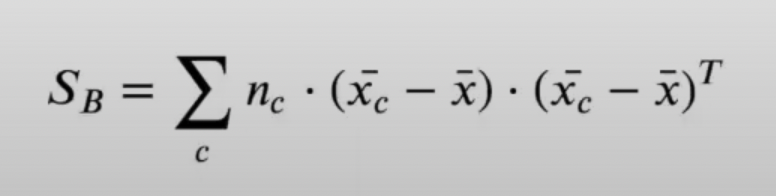
fisher LDA uses two scatters to extract features: 1-within scatter matrix and 2-between scatter matrix.

This method calculates both scatters with the formulas below :

1. Within scatter :



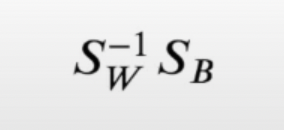
2-between scatter



And this function returns both within and between scatters.

## GeneralSoloution function

This method uses the formula below to solve the Eigenvectors and Eigenvalues of data.



Because our within-scatter matrix is not ranked and can't invert it, we used a unique shrinkage method to inverse the matrix. In this method, we add a small amount to the main diagonal of the scatter matrix. And then it's invisible.

After that *np.linalg.eig* function has been used to solve the eigenvectors and eigenvalues of our data.

We can reshape our eigenvectors and plot them. Like the figure below :



Eigenvectors reshaped to 64x64

## Sorteigens function :

This function gets our eigenvectors and eigenvalues. Then eigenvectors and eigenvalues were sorted in decreasing order.

## Transform function

This method gets x, eigenvectors, and K as arguments and then uses the dot product of the eigenvectors and data to create transformed samples.

And this function returns transformed samples. (feature-reduced models)

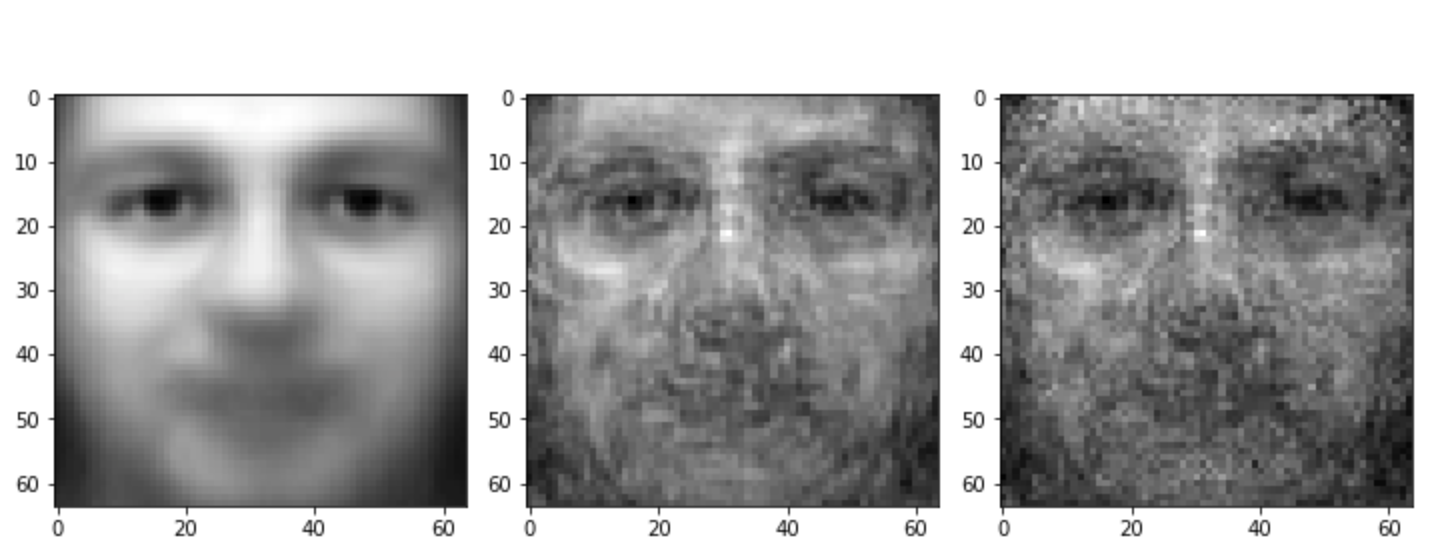
## Reconstruct method

This method tries to reconstruct our original data from transformed values.

This function reconstructs original data with the dot product of eigenvectors and transformed data.

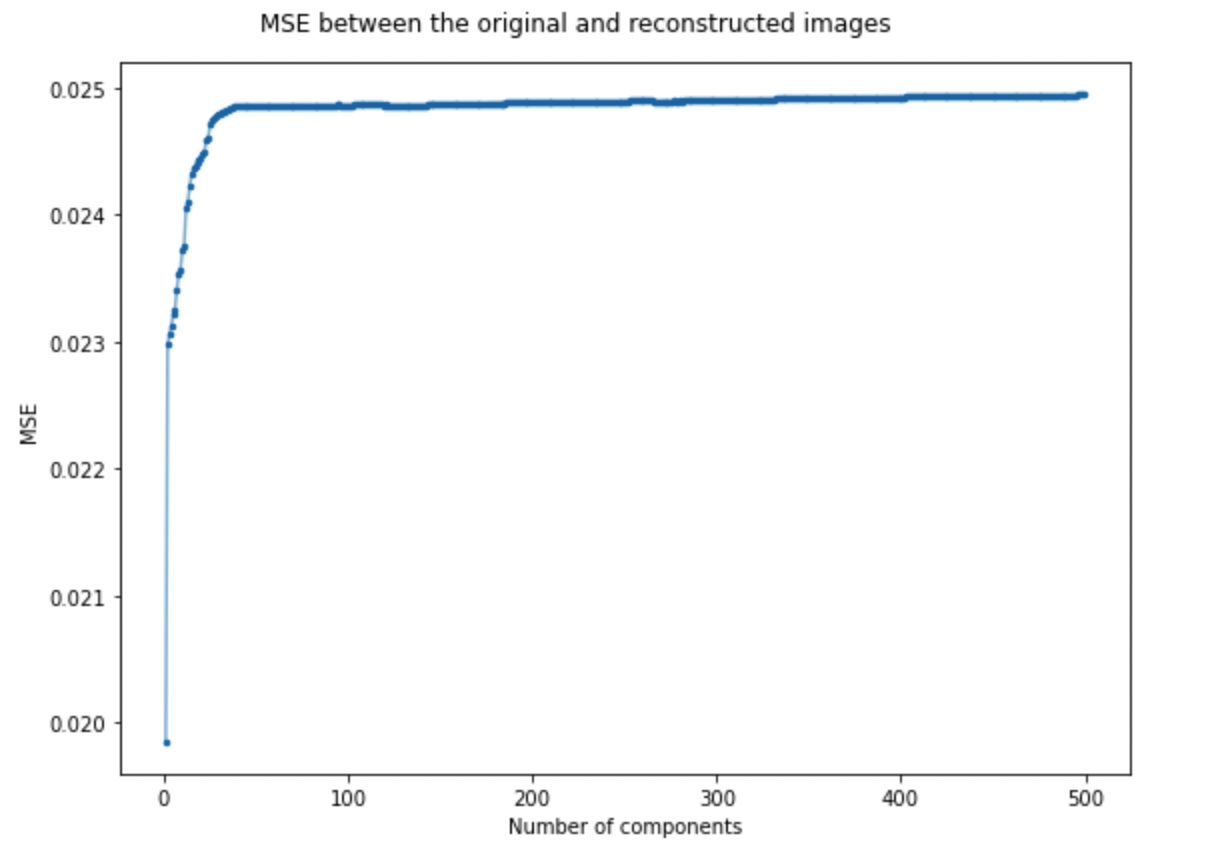
Reconstructed data could be seen:

K=1 K = 60 K=40



So after k = 40, the pictures won't get any better, but they also become noisier.

The MSE plot is :



# Answer to asked questions :

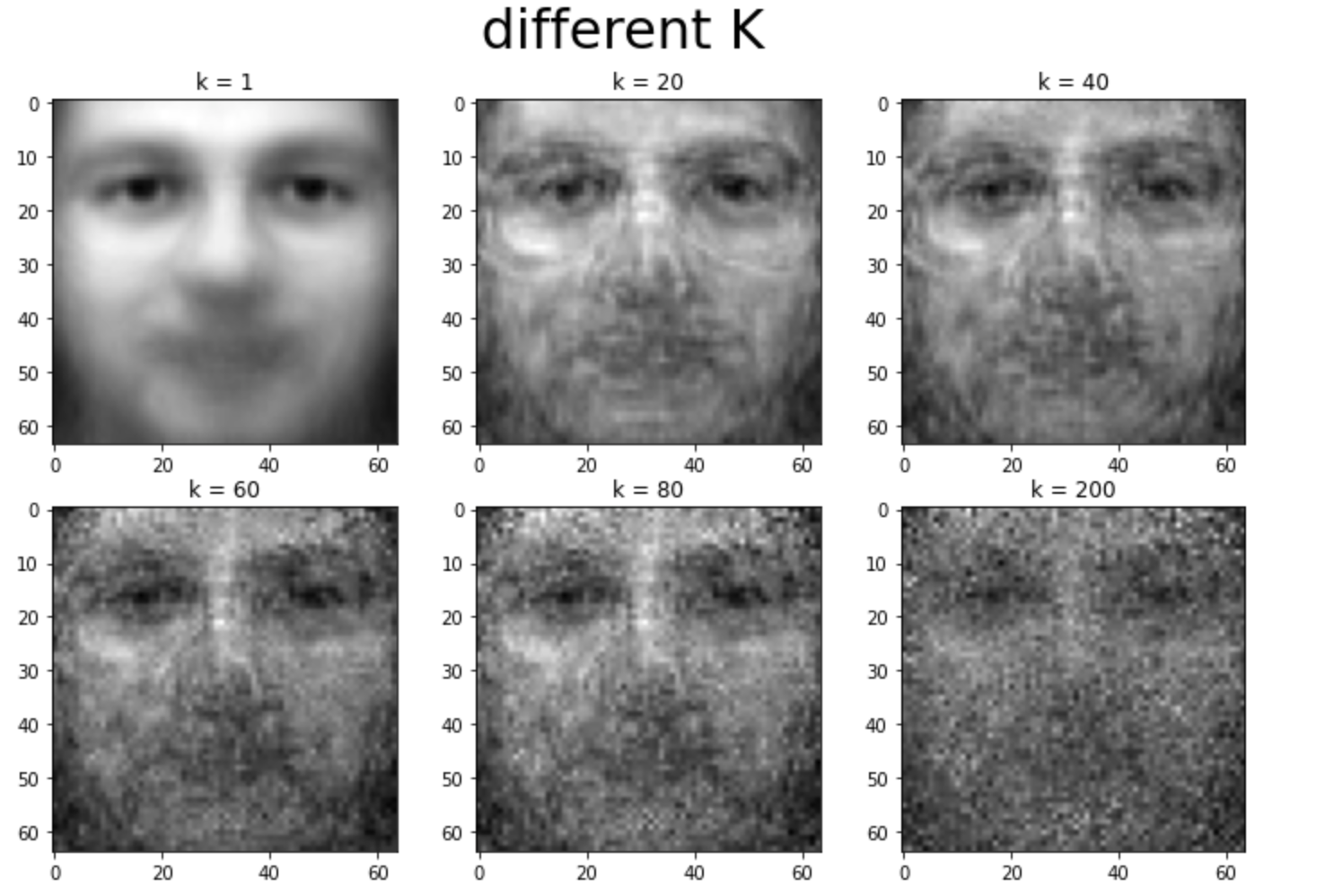
* What is the problem with applying Fisher LDA to the dataset?

After using Fisher LDA, the original data can hardly be reconstructed.

The within-scatter matrix is not positive definite, so it can not be inverted.

* What would happen if we had many outliers in the dataset?

Because we are using the mean of our data to calculate the within-scatter and between-scatter matrixes, our error gets high if we have many outliers in our dataset. It affects our projection(feature reduction), and also it involves our reconstruction.



As illustrated in the picture, if we increase the K after getting to the number of classes, in this case, 40, the image won't be any better, but it becomes noisier.