Amirkabir University of Technology
(Tehran Polytechnic)

Assignment 4
Overview of Dimensionality Reduction and Supervised Learning

Homeworks Guidelines and Policies

- ***What you must hand in.*** It is expected that the students submit an assignment report (HW4_[student_id].pdf) as well as required source codes (.m or .py) into an archive file (HW4_[student_id].zip). Please combine all your reports just into a single .pdf file.
- ***Pay attention to problem types.*** Some problems are required to be solved *by hand* (shown by the icon), and some need to be implemented (shown by the icon). Please do not use implementation tools when it is asked to solve the problem by hand, otherwise you will be penalized and lose some points.
- ***Don't bother typing!*** You are free to solve by-hand problems on a paper and include their pictures in your report. Here, cleanness and readability are of high importance. Images should also have appropriate quality.
- ***Reports are critical.*** Your work will be evaluated mostly by the quality of your report. Do not forget to explain your answers clearly, and provide enough discussions when needed.
- ***Appearance matters!*** In each homework, 5 points (out of a possible 100) belong to compactness, expressiveness, and neatness of your report and codes.
- ***MATLAB is also allowable.*** By default, we assume you implement your codes in Python. If you are using MATLAB, you have to use the equivalent functions when it is asked to use specific Python functions.
- ***Be neat and tidy!*** Your codes must be separated for each question, and for each part. For example, you have to create a separate .py file for part b. of question 3, which must be named 'p3b.py'. (or .ipynb)
- ***Use bonus points to improve your score.*** Problems with bonus points are marked by the icon. These problems usually include uncovered related topics, or those that are only mentioned briefly in the class.
- ***Moodle access is essential.*** Make sure you have access to Moodle, because that is where all assignments as well as course announcements are posted. Homework submissions are only made through Moodle.

- ***Assignment Deadline.*** Please submit your work **before the end of January 2nd**.
- ***Delay policy.*** During the semester, students are given only **10 free late days** which they can use them in their own ways. Afterwards, there will be a 20% penalty for every late day, and no more than four late days will be accepted.
- ***Collaboration policy.*** We encourage students to work together, share their findings, and utilize all the resources available. However you are not allowed to share codes/answers or use works from the past semesters. Violators will receive a zero for that particular problem.
- ***Any questions?*** If there is any question, please do not hesitate to contact us through the Telegram group chat or following email addresses: **m.ebadpour@aut.ac.ir** and **atiyeh.moghadam@aut.ac.ir**.

1. Investigating Genes in Lower Dimensions! (22 Pts.)

Keywords: *dimensionality reduction, fisher linear discriminant analysis, LDA, principal component analysis, PCA, curse of dimensionality.*

Pattern recognition plays a pivotal role in advancing biological research by providing a powerful set of tools to unravel complex relationships within biological data. In various biological tasks, including genomics, proteomics, and epidemiology, pattern recognition techniques are instrumental in identifying meaningful patterns and extracting valuable insights.

Dimensionality reduction methods are used in enhancing the analysis of biological tasks, particularly in the realm of genetic studies. In the context of genetic tasks, datasets often suffer from the curse of dimensionality, where the number of variables (genes or features) is much larger than the number of samples. This abundance of features can lead to computational inefficiency, increased noise, and difficulties in identifying relevant patterns. Dimensionality reduction techniques, such as Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), address these challenges by transforming the original high-dimensional data into a lower-dimensional space while retaining the essential information. In this problem, we are going to further familiarize ourselves with these methods.

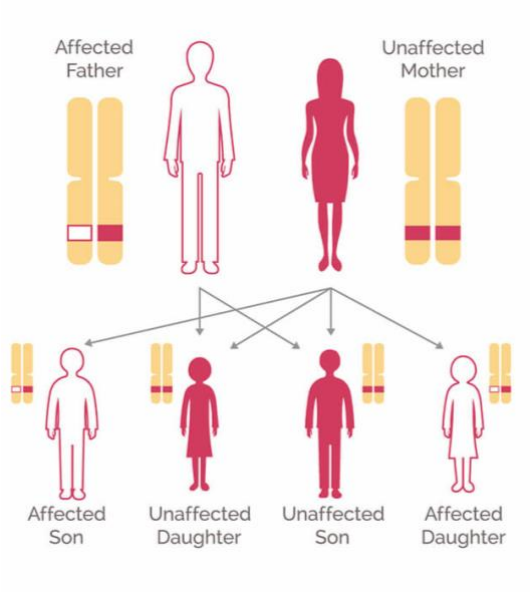
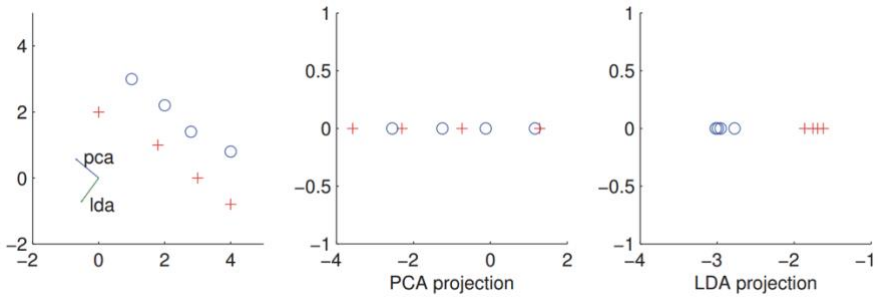


Figure 1: Pattern recognition methods help classify different subtypes of cancers based on their genetic profiles. This can aid in personalized treatment strategies and prognosis prediction.

In the figure below, we see a synthetic two-dimensional data where LDA does a better job than PCA in terms of class separation. Draw a similar dataset where PCA and LDA find the same good direction. Draw another where neither PCA nor LDA find a good direction.



Now, Given the following dataset for a 2-class problem with 2-D features:

$$c_1 = \left\{ \begin{bmatrix} 2 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 3 \end{bmatrix} \right\} \quad c_2 = \left\{ \begin{bmatrix} 4 \\ 3 \end{bmatrix}, \begin{bmatrix} 5 \\ 3 \end{bmatrix}, \begin{bmatrix} 6 \\ 4 \end{bmatrix} \right\}$$

a. Using PCA, find the best direction onto which the data will be projected on. Express the equation of the line along this direction passing through the samples mean in the following form: $w^T x + w_0 = 0$, where:

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

and w_0 is a scalar known as the bias. Plot that line along with all the sample points.

b. Project and reconstruct all the sample points. Plot the recon- structed points and verify that they fall along the line plotted earlier.

c. Find the total mean square error MSE for all sample points (between the original points and the reconstructed points.) Also find the Fisher Ratio for this projection defined by:

$$FR = \frac{(m_1 - m_2)^2}{\sigma_1^2 + \sigma_2^2},$$

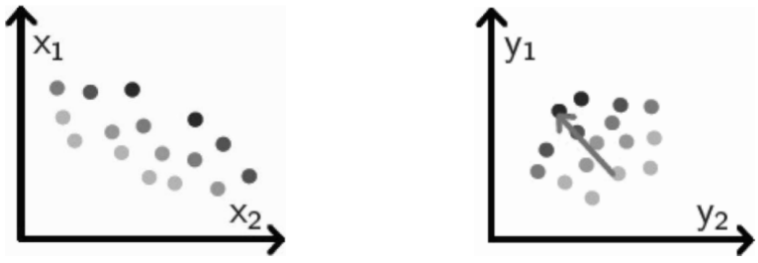
where m_i is the mean of the projected samples of class i, and σ_i^2 is the equivalent variance. You can compute the FR on the projected 1-D points (rather than the reconstructed points which are 2-D vectors).

d. Using Fisher Linear Discriminant Analysis (LDA), determine the best one-dimensional space onto which the above data should be projected. Repeat part (a) and (b) for this projection.

e. Find the total MSE for all sample points (between the original points and the reconstructed points). Compare that to the total MSE found in the first question.

f. Find the Fisher Ratio for this projection. You can compute the FR on the projected points (rather than the reconstructed points which are 2-D vectors). Compare that to the FR found in part 1. Interpret your result.

Canonical correlation analysis (CCA) handles the situation that each data point (i.e., each object) has two representations (i.e., two sets of features), e.g., a web page can be represented by the text on that page, and can also be represented by other pages linked to that page. Now suppose each data point has two representations x and y, each of which is a 2-dimensional feature vector (i.e., $x = [x_1, x_2]^T$ and y= $[y_1, y_2]^T$). Given a set of data points, CCA finds a pair of projection directions (u, v) to maximize the sample correlation $corr(u^T x)(v^T y)$ along the directions u and v.



g. Now we can see data points shown in the Figure above, where each data point has two representations $x = [x_1, x_2]^T$ and y= $[y_1, y_2]^T$. In the right figure we've given one CCA projection direction v, draw the other CCA projection direction u in the left figure.

2. PCA and LDA in the Face Recognition Problem: Eigenfaces vs. Fisherfaces (28+10 Pts.)

Keywords: *Classification Problem, Face Recognition, Principal Component Analysis (PCA), Linear Discriminant Functions (LDA), K-Nearest Neighbor Classifier, Fisherfaces, Eigenfaces*

Matthew Turk and *Alex Pentland* were the first to apply **Principal Component Analysis** (**PCS**) in **Face Recognition** problem. Their approach – introduced in 1991 and known as **Eigenfaces** – is considered as the first successful face recognition algorithm. **Fisherfaces**, on the other hand, was an enhancement of the Eigenfaces method, which were proposed by *Joao Hespanha* in 1997. As can be guessed, it applies **Fisher's Linear Discriminant Analysis** (**FLDA** or **LDA**) for the **Dimensionality Reduction**.

(a)

(b)

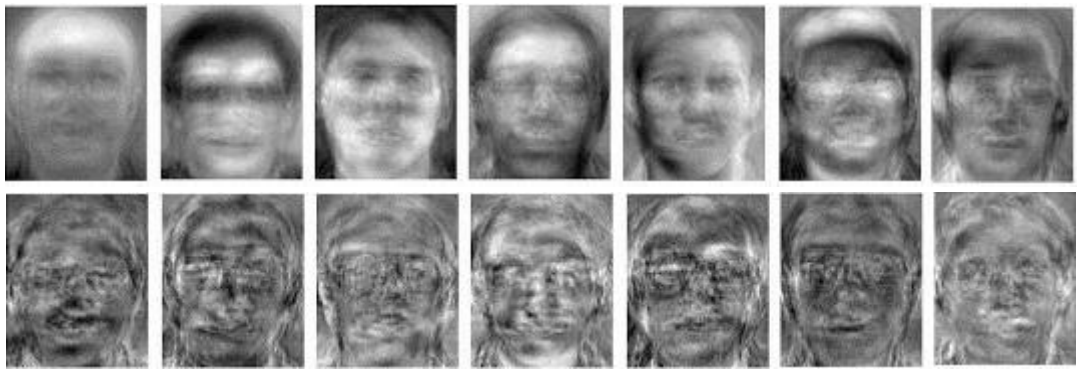


Figure 2 Examples of Eigenfaces and Fisherfaces obtained in a certain dataset (a) First 7 Eigenfaces (b) First 7 Fisherfaces

In this problem, the goal is to implement Eigenfaces and Fisherfaces for recognising faces. You will be using AT&T dataset, which contains 400 images of the size 112×92. There are 40 distinct subjects, 10 images per each person. Here, 360 images are selected as the training set (9 per each subject) and 40 as the test set (one per each subject). Training set images and their corresponding labels are stored in 'trainset_imgs.csv' and 'trainset_labels.csv' respectively, while test set images and their labels are in 'testset_imgs.csv' and 'testset_labels.csv' respectively, all attached to 'inputs/P7' folder of this homework.

Figure 3 AT&T face images for a specific subject. As can be seen, the main variation in the face images is in their pose

First, let's see how Eigenfaces algorithm deals with this dataset.

a. Write a function `[W,mu]=eigenfaces_train(trainset,v)` which takes $n \times d$ train set images matrix `trainset`, where $n = 360$ is the number of training images, and $d = 10304$ is the number of pixels in each image. This function performs Principal Component Analysis on the data and computes the top `v` eigenvectors. It return the $v \times d$ matrix of eigenvectors `W`, and a d-dimensional vector `mu` representing the mean of the training faces. Compute the top 50 eigenvectors.

b. Reshape each of the top 50 eigenvectors obtained in the previous part, and display them by appending them together into a 1120×460 image (a grid of images containing 5 rows and 10 columns).

c. Select 10 random images from the train set, and show what each of these images would look like when using only the top v eigenvectors to reconstruct them, where $v = 1, ..., 10$.

   Hint: This reconstruction procedure should project each image into a v-dimensional space, project that v-dimensional space back into a 10304 dimensional space, and finally reshape that 10304 vector into a 112×92 image.

d. Now let's test the Eigenfaces method performance. Implement a function with the header `testset_labels=eigenfaces_test(trainset,trainlabels,testset,W,mu,v)` which takes the train set matrix `trainset` as well as their labels vector `trainlabels`, test set matrix `testset` and outputs of PCA, `W` and `mu`, and the number of eigenvectors to use `v`. Your function must first project each image from `trainset` and `testset` onto the space spanned by the first `v` eigenvectors. Then it must classify each test image using the nearest neighbor (1-NN) classifier by considering L_2 distance in the lower dimensional space. It must return an m-dimensional vector `testset_labels` encoding the predicted class label for each test face, where $m = 40$ is the number of test images. Evaluate `eigenfaces_test` on the test set images for values $v = 1, ..., 50$, and plot the error rates as a function of `v`.

e. Repeat the previous part, but throw away the first 5 eigenvectors. In other words, use `v` eigenvectors starting with the 6th eigenvector. Produce a plot similar to the one in the previous step, and comment on the result. How do you explain the difference in recognition performance in comparison with the previous part?

Next, it's time to examine Fisherfaces.

f. Implement a function `[W,mu]=fisherfaces_train(trainset,trainlabels,c)` that takes the same $n \times d$ train images matrix `trainset` as well as their corresponding labels vector `trainlabels`, and the number of classes $c = 40$. Your function must do the following:

   - Compute the mean `mu` of the training data, and use PCA to compute the first $n - c$ principal components. Let this be W_{PCA}.
   - Use the obtained W_{PCA} to project the training images into a $n - c$ dimensional space.

- Compute the between-class scatter matrix S_B and the within-class scatter matrix S_W on the $n-c$ dimensional space from the previous space.
- Compute W_{FLD} by solving for the generalised eigenvectors of the $c-1$ largest generalised eigenvalues for the problem $S_B w_i = \lambda_i S_W w_i$.

  Hint: In MATLAB, you can use the function `eig()` to solve for the generalised eigenvalues of S_B and S_W.
- The fisher bases will be a $W = W_{FLD} W_{PCA}$, where W is $(c-1)\times d$ dimensional, W_{FLD} is $(c-1)\times(n-c)$ dimensional, and W_{PCA} is $(n-c)\times d$ dimensional.

g. As in the Eigenfaces method, reshape the top 50 Fisher bases you obtained in the previous part into images of the size 112×92 and stack them into one big 1120×460 image.

h. Perform recognition on the test set with Fisherfaces. As in the Eigenfaces exercise, use a L_2 nearest neighbor classifier (1-NN), and evaluate results for each test images for values $v = 1,...,50$. Plot the error rates as a function of `v`.

Note I: Images of the given dataset are stored in `double` format.

Note II: You should implement asked functions and are not allowed to use built-in functions.

Useful Sources: [1], [2], [3]

3. Performing Image Compression using PCA (22+5 Pts.)

Keywords: *Principal Component Analysis, Image Compression, Feature Selection*

Although **Principal Component Analysis** (**PCA**) is considered as a classical approach, yet it has numerous applications in the field of machine learning. In this problem, you are going to test PCA performance in an image processing application, i.e. **Image Compression**.

The approach considered here is based on the fact that subimages of an image can be presented using relatively few features. Therefore, instead of storing the original image, the values of the features in the subimages can be restored and then used for reconstruction.

a. First, you are working with the image "sad_days_gray.jpg" as the input. Implement a function `patch_extract()` which takes an input image and returns a matrix of vectorised blocks of the image (known as patch) as its columns. More specifically, this function scans the input image and extracts non-overlapping 8×8 pixel blocks. Then it converts them into column vectors and construct a matrix. Here, since the image size is 240×360, the output would be a 64×1350 matrix of image blocks.

(a) **(b)**

Figure 4 Two images are intended to be compressed here, one in grayscale and the other in RGB space

b. Perform PCA analysis of the covariance matrix of the output obtained by the function `patch_extract()`. Find and report the first 20 largest eigenvalues and their corresponding eigenvectors. Also display the mean image and plot the eigenvectors (or "eigenimages") which correspond to the 8 largest eigenvalues.

c. Compress the subimages using the mean vector and the eigenvectors corresponding to the $k=2,5,10,20$ largest eigenvalues.

d. Now try to reconstruct each subimage using its low-dimensional representation. Remember to include the mean value.

e. Now implement a function `patch_reconstruct()` to merge the reconstructed subimages and create a 240×360 pixel image again. Display the reconstructed and original images together and comment on the results. Also discuss on the effect of k.

f. Repeat the previous parts for the color image "sad_days_rgb.jpg".

Note: You should implement asked functions and are not allowed to use built-in functions.

4. PCA Against LDA: Variation vs. Separation (23 Pts.)

Keywords: *Dimensionality Reduction, Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA)*

Up until now, you've encountered some basic **PCA** and **LDA** problems. Now let's study their behavior more intuitively. Here's a simple problem for warm-up. Consider a two-category problem where each class follows the statistics below:

$$\mu_1 = [10 \quad 10]^T \qquad \mu_2 = [10 \quad 22]^T$$

$$\Sigma_1 = \Sigma_2 = \begin{bmatrix} 9 & 4 \\ 4 & 4 \end{bmatrix}$$

a. Generate 1000 samples for each class. Plot these samples and highlight each class with different colors.
b. Find the line on which PCA projects the samples. Draw this line on the previous figure.
c. Project all sample points onto the obtained PCA line and display the results.
d. Explain your observation. Does it match with what you expected?
e. Reconstruct the initial samples in 2-D space, and find the reconstruction error.
f. Now find the line on which LDA projects the samples. Draw this line on the same figure.
g. Project all sample points onto the obtained LDA line and display the result.
h. Explain your observation. Does it match with what you expected?

Now we will deal with a more challenging problem. Download the dataset QuickDraw10, which is a MNIST-like dataset obtained by users drawing different objects in Google "Quick, Draw!" online game.

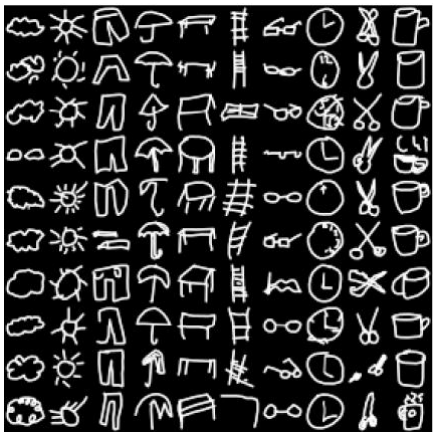


Figure 5 QuickDraw10 is a subset of QuickDraw dataset, which is considered as an alternative for the famous MNIST dataset

i. Considering only the training set, implement PCA and plot the data points using the first 2 principal components. Distinguish between the 10 classes with different colors.
j. Now consider only two classes, "pants" (2) and "eyeglasses" (6). Apply PCA to the training samples and project them onto the first 2 principal components. Then apply LDA to project them onto 1 dimension. Determine the separator line and report the confusion matrix for both the training and test sets separately.
k. Draw the plot of the first two principal components for just "pants" and "eyeglasses" as well as the separating line you obtained in the previous part.
l. Display the image for the "most incorrectly" classified examples. That is, show the image for the "pants" test sample whose LDA projections lies the most in the "eyeglasses" direction and the image for the "eyeglasses" test sample whose LDA projections lies the most in the "pants" direction.

m. Repeat the previous LDA classification, however try to use enough principal components to capture 90% of the variance in the training data. You can determine the required number of components separately and just report the number.

n. Implement the K-NN algorithm and apply it to the same dataset as in the previous problem, using the PCA needed to capture 90% of the variance. Set $k = 1, 3, 5, 7, 9$ and find the k which yields the lowest error rate on the training set. Report the resultant confusion matrices for both the training and test set using that best k. Also display the images for your choice of two misclassified test samples, one for a misclassified "pants" and one for a misclassified "eyeglasses".

Note: You should implement asked functions and are not allowed to use built-in functions(like K-NN).

Good Luck!

Mohsen Ebadpour, Atiyeh *Moghadam, Romina Zakerian*