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 (.pdf) as well as – if necessary – required source codes (.m or .py) into an archive file named according to the following template: HW4_XXXXXXXX.zip
  where XXXXXXXX must be replaced by their student ID.
- ***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 don't use implementation tools when it is asked to solve the problem by hand, otherwise you'll be penalized and lose some points.
- ***Don't bother typing!*** You are free to solve by-hand problems on a paper and include picture of them 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. Don't forget to explain what you have done, and provide enough discussions when it's needed.
- ***Appearance matters!*** In each homework, 5 points (out of a possible 100) belongs to compactness, expressiveness and neatness of your report and codes.
- ***Python is also allowable.*** By default, we assume you implement your codes in MATLAB. If you're using Python, you have to use equivalent functions when it is asked to use specific MATLAB 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 .m file for part b. of question 3. Please name it like p3b.m.
- ***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's where all assignments as well as course announcements are posted on. Homework submissions are also done through Moodle.

- ***Assignment Deadline.*** Please submit your work **before the end of January 25th**.
- ***Delay policy.*** During the semester, students are given 7 free late days which they can use them in their own ways. Afterwards there will be a 25% penalty for every late day, and no more than three 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 don't hesitate to contact me through **ali.the.special@gmail.com**. You may also find me in the pattern recognition and image processing lab, 3rd floor, CEIT building.

1. When Only Maximising Variance is All That Matters (12 Pts.)

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

Principal Component Analysis (**PCA**) is arguably the most popular method of **Dimensionality Reduction**, in which the goal is to represent the variable space with a few orthogonal (uncorrelated) variables that capture most of its variability. Therefore, PCA is somehow just a coordinate transformation which simplifies the complexity in high-dimensional data while retaining trends and patterns.

In this problem, you will carry out a variety of examples intended to get you more familiar with PCA calculations steps. First, assume you are attempting to build a lung cancer detection machine which takes smokers daily records, more specifically "total number of cigarettes smoked" and "total time of physical activity", and determines whether the patient suffers from lung cancer or not. Figure 1 shows the train data where filled circles are those who diagnosed with lung cancer and unfilled circles are healthy individuals. Suppose we already know that the first principal component is $\left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)$. Also assume that centering of the data is not needed.

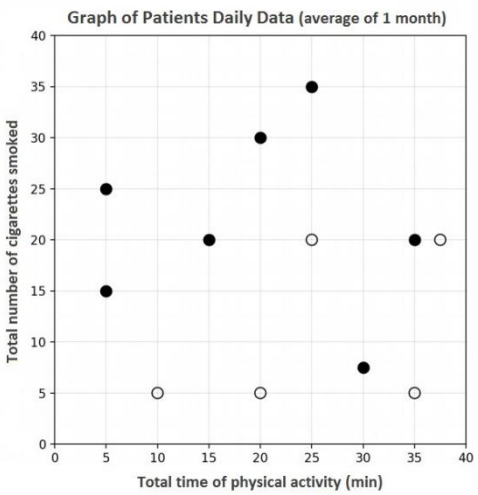

Figure 1 Graph of smokers daily records which is the average of a 1-month study

a. Draw the first principal component as a line on the graph. For your convenience, a copy of Figure 1 is provided in the homework directory.
b. How would a 1-NN classifier classify a patient who smokes an average of 21 cigarettes and spends an average of 5 minutes of physical activity every day?
c. How would a 1-NN classifier classify the same patient after transforming the data to one-dimension? Explain your answer.
d. Are your answers for part b and c the same? Why or Why not?

Now let's perform PCA by hand for a simple dataset as follows:

$$X = \begin{bmatrix} -1 & 6 & -1 & 2 & 0 & -1 & -1 & -2 & 5 & 6 \\ 6 & 6 & 5 & 6 & 8 & 5 & 6 & 5 & 6 & 7 \\ 7 & 9 & 7 & 7 & 8 & 9 & 7 & 8 & 8 & 6 \end{bmatrix}$$

where each row corresponds to a gene (gene 1, gene 2 and gene 3) and each column corresponds to an experiment.

e. Find the mean expression value for each gene.
f. Find the variance of the expression values for each gene.
g. Subtract the mean expression value for each gene from matrix entries for that gene. Call this matrix $\tilde{X}$.
h. Find the gene-covariance matrix C using the data in $\tilde{X}$.
i. Calculate the eigenvalues of C.

j. What fraction of the total variance of the data is accounted for by the first principal component of C? Note that the total variance of the data is the sum of the variances of gene 1, 2 and 3 that you calculated in the previous steps.
k. Find the principal component eigenvectors. Order these eigenvectors appropriately.
l. Re-write the gene-experiment matrix $\tilde{X}$ as a principal component-experiment matrix by projecting each data point (column) onto the PCs.

2. Linear Discriminant Analysis: Bringing Labels into Action (10 Pts.)

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

In terms of **Supervised** and **Unsupervised** learning, **PCA** falls into unsupervised techniques as it ignores the class labels and only attempts to maximise the variance of the data in order to find the directions. **Linear Discriminant Analysis** (**LDA**), on the other hand, is another **Dimensionality Reduction** method which takes the class label into consideration by reducing dimensionality while at the same time preserving as much of the class discrimination information as possible. In another word, it maximises the distance between the centroid of each class and simultaneously minimises the variation within each category.

Consider a dataset with two classes, each with the following summary statistics:

$$\bar{X}_1 = \begin{bmatrix} 1.80 \\ 3.89 \end{bmatrix}, \quad \bar{X}_2 = \begin{bmatrix} 5.50 \\ 8.01 \end{bmatrix}, \quad S_1 = \begin{bmatrix} 1.21 & 1.10 \\ 1.10 & 2.00 \end{bmatrix}, \quad S_2 = \begin{bmatrix} 1.06 & 1.62 \\ 1.62 & 2.50 \end{bmatrix}$$

a. Find the linear discriminant function.
b. Classify the point $x = \begin{bmatrix} 1.27 & 4.45 \end{bmatrix}^T$ using the linear discriminant function you obtained in the previous part.
c. Which theoretical assumptions must take into account for this method to be reliable? Are these assumptions (at least partly) valid for the present data?
d. Assume we change S_2 by $S_2' = \begin{bmatrix} 41 & 40 \\ 40 & 41 \end{bmatrix}$. Would it still be reasonable to apply linear discriminant to classify x? If not, then what would you do instead?
e. Perform your proposed solution for part d.

Now consider another dataset with two classes as follows:

$$X_1 = \begin{bmatrix} 3 & 1 & 4 & 2 & 5 \\ 4 & 2 & 3 & 2 & 2 \end{bmatrix}, \quad X_2 = \begin{bmatrix} 8 & 5 & 7 & 9 & 6 \\ 7 & 9 & 6 & 4 & 6 \end{bmatrix}$$

f. Plot the sample points.
g. Find and plot the LDA projection line.
h. Project the sample points onto the obtained subspace. Plot the resultant points.
i. Discuss the discriminability of the classes in the new subspace.

3. PCA Against LDA: Variation vs. Separation (18 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:

$$\mu_1 = \begin{bmatrix} 10 & 10 \end{bmatrix}^T \quad \mu_2 = \begin{bmatrix} 10 & 22 \end{bmatrix}^T$$

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

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.

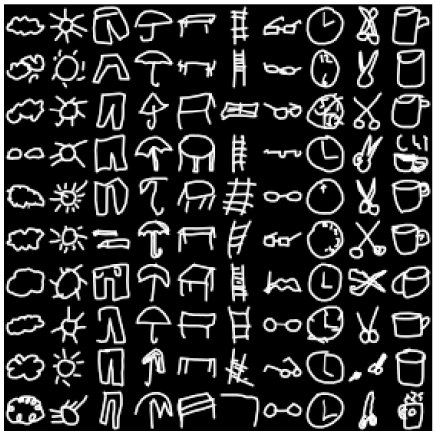

Figure 2 QuickDraw10 is a subset of QuickDraw dataset, which is considered as an alternative for the famous MNIST dataset

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: Using built-in functions is allowed in this problem.

4. Let's Get Our Hands Dirty: Designing Practical Neural Nets (16 Pts.)

Keywords: *Binary Classification, Supervised Learning, Neural Networks, Perceptron Rule, Linearly Separable*

In machine learning, a **Perceptron** is a single layer neural network and a multi-layer perceptron is called **Neural Networks**. It takes inputs, associates a set of **Weights** to them along with a **Bias**, aggregates them through its **Hidden Layers** (if any) and **Activation Function**, and returns 1 only if the aggregated sum is more than some threshold, else returns 0. A single perceptron is only capable of implementing **Linearly Separable** patterns.

Now let's practice! Consider a binary classification problem with inputs x_1 and x_2, each take values between 0 and 1. In each of the following part, design a neural network capable of performing the required classifications. You must specify network architecture as well as the parameters. Use the smallest number of units possible. Make your own assumption if necessary.

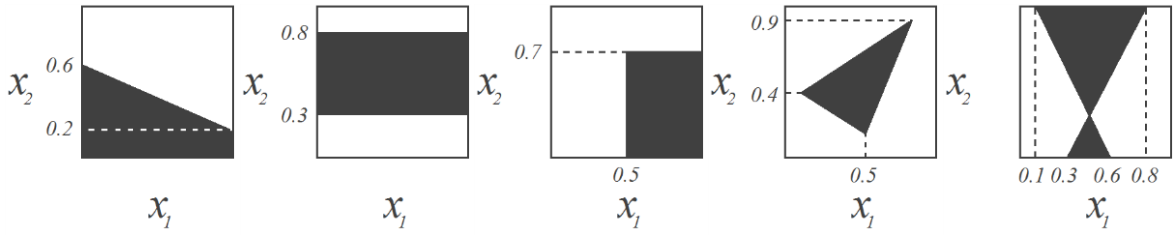

Figure 3 Different classification scenarios in 2-D feature space

In another scenario, we're going to deal with a real world problem. The International Rhino Foundation (IRF) is a charity foundation dedicated to the conservation of the five species of rhinoceros, the *White Rhinoceros* and *Black Rhinoceros* in Africa, the *Indian Rhinoceros*, *Javan Rhinoceros* and *Sumatran Rhinoceros* in Asia.

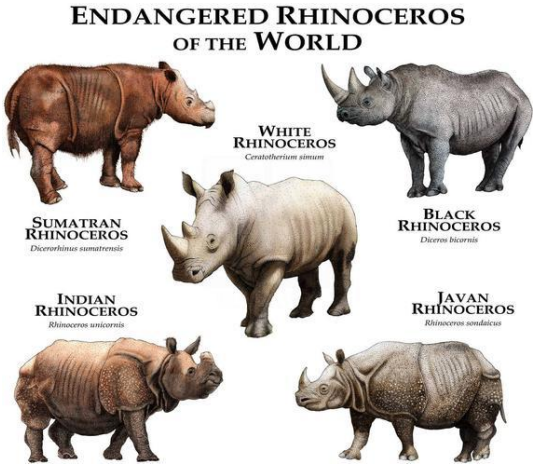

Figure 4 Five different endangered rhinoceros species. The goal is to apply ANN in order to classify their images based on rhinoceros physical characteristics

In one of their recent survival researches, IRF has decided to separate these five species in images using their physical characteristics. The program which is considered to perform this task applies neural networks. It takes five measurements; body height, body length, eye diameter, horn length and leg length. A database of a few hundred labeled images of individuals of these species is also available.

You are to help them design such a network.

f. How many input units should the network have?

g. How many output units should the network have?
h. Should the network use hidden units or not?
i. Should the network use feedforward connections, recurrent connections, both, or neither?
j. What activation function(s) should the neurons use?
k. What learning mechanism(s) should the network use?

5. Wrestling with Support Vector Machines in MATLAB (15 Pts.)

Keywords: *Classification Problem, Supervised Learning, Support Vector Machine, Kernel Trick*

Support Vector Machine (**SVM**) is a linear model which can solve linear and non-linear problems with a simple idea; it creates a line or a hyperplane which separates the data into classes. There are many possible choices for this line or hyperplane, and SVM attempts to find a plane that has the maximum **Margin**, which is the distance between data points of both classes. Data points that are closer to this hyperplane and influence on the position and orientation of the hyperplane are called **Support Vectors**.

The goal of this problem is to study SVM in practice. But don't worry, you don't have to write any code. Instead, a straightforward implementation of a support vector machine is given to you and you are expected to complete its missing parts properly. Note that the code is written in MATLAB.

a. The function `classify_grid()` applies a learned SVM classifier which is needed to test a classifier. Fill the line 41 of this function with an appropriate statement.
b. Fill the line 47 of the function `trainsvm()`.
c. Now fill the line 77 of the function `trainsvm()`. Solve the SVM optimisation problem by using the quadprog solver in MATLAB.
d. Load the file "test1.dat" and run the following command:
```
[SV,alpha,b] = trainsvm(D,inf,@kernel_linear,[]);
```
You should end up seeing figures like those in Figure 5. Explain the parameters that the function returns.

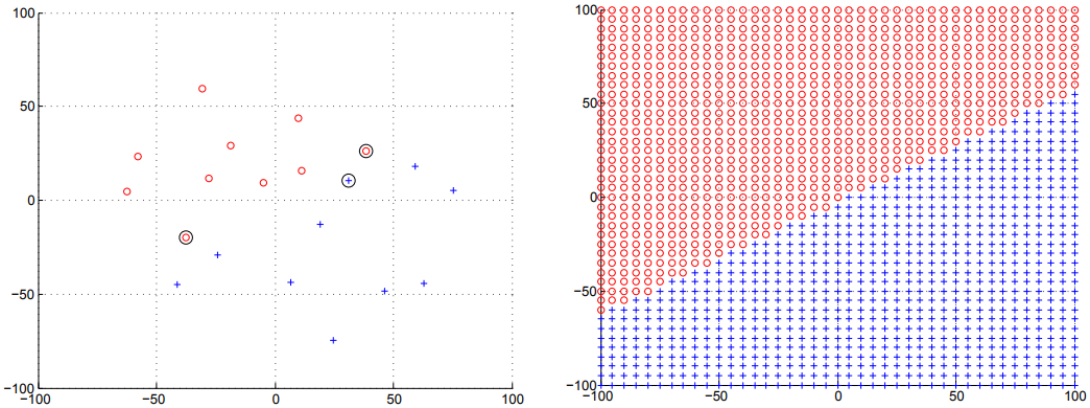

Figure 5 You are expected to obtain these two figures after filling the missing parts in the given code

e. What are the circled points in Figure 5? Explain.
f. What is the `inf` parameter in the inputs of the function `trainsvm()`?
g. Implement a modified kernel function that performs a quadratic kernel. Load the file "nonlin1.dat" and run the SVM. Plot and explain the output.
h. Now load the file "noise1.dat" and run the code. Set the second parameter to `inf`, 10, 1 and 0.1 and plot the obtained results. What's your observations? Specify the role of this parameter.

6. Performing Image Compression using PCA (16 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 6 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".

Recommended MATLAB Functions: `cov()`, `eig()`, `reshape()`, `imagesc()`, `colormap()`, `subplot()`

7. Some Explanatory Questions (8 Pts.)

Please answer the following questions as clear as possible:

a. Does it make any sense to do PCA with number of principle components greater than dimensions? Justify your answer.
b. Describe the meaning of “label switching” in the context of principal component analysis.
c. SVD and PCA are said to be synonymous. Can they really produce the same projection result? If yes, under what circumstances? If no, explain why.
d. When dealing with data of high dimensionality like images, PCA suffers from a numerical computational issue. Explain what it is and how it can be avoided.
e. Image denoising is the process of reducing noise from a noisy image, so as to restore the true image. Explain how PCA can be applied in such a problem.
f. Which topological properties must a network fulfill in order to be able to be trained using backpropagation algorithm?
g. Is it possible to use the logistic sigmoid in the hidden layers of a neural network? Why or why not?
h. Can SVM handle regression problem? If not, explain why. If yes, explain how.

Good Luck!
Ali Abbasi