Assignment 2

Submission date: 12 /01/2021 , 23:59

General Instructions:

- The solution should be formatted as a report and running code should be included in a digital form.
- The solution can be done in pairs independently to other groups. Identical (or very similar solutions) are not allowed!
- Write your code in Python.
- Please submit the assignment via Moodle.
- The code must be reasonably documented.

Question 1:

In this exercise, we will perform handwritten digit classification. The following method makes use of PCA and KNN classifier.
All the following steps should be formatted as a report and a running code in Python should be included in a digital form.

Step 1:
Download MNIST database from: http://yann.lecun.com/exdb/mnist/ and extract it to your project directory. Load the dataset to your project, you can use the 'mnist' package of python (https://pypi.org/project/python-mnist/).
Plot to your report one image from each class.

Step 2:
The average image for each digit is called centroid and is calculated by averaging the points in each dimension individually. Include in your report a plot of the centroid images of all digits.

Step 3:
In this step we investigate how the centroids differ one from another. We'll measure this difference by taking each pixel (each position in the image) as a dimension in the digit representation space and compute Euclidian distance between each pair of centroid images.

Present the table of distances between all pairs of centroids. What does this table tell us regarding the classification? Give an example of two digits that are easier for classification and two digits that are harder, based on the table of distances.

Step 4:
Not all the pixels (positions in the image) have the same importance in the digit classification. Some positions have exactly the same value in all the images in the dataset (for example, the image borders). These pixels have zero-variance (are constant).
Plot a histogram of the distribution of pixels variances in the dataset. To construct the histogram, calculate the variance for each position in the image over the entire dataset. Then, bin the range of the variances into 10 non-overlapping intervals of equal size. For each interval, count how many positions fall into this interval. Explain your plot.

Step 5:
Remove zero variance pixels from all the images in the dataset. Then, compute PCA using all the training data.Visualize the projection in 2D: project 100 points from each class onto the first two principle components and use scatter plot with different colors per class for visualization. To observe the separable structure, show also a scatter plot with different colors for the 3 classes: 0,1,9.

Step 6:
Explore the variances in the data by plotting the eigenvalues (in descending order).
Next, decide on the range of the dimensions for the feature vector (e.g. by retaining 75-90% of the energy).

Step 7:
We'll use KNN for classification. We need to decide the number of dimensions to be used (n) from the range determined in the previous step and the number of k-Nearest Neighbors (k). Split the dataset into training and validation set and find the best parameters using the validation set. Run KNN with the best parameters using both training and validation sets as the reference set and calculate the accuracy on the test set.

Question 2:

In this exercise you are given 4 toy datasets for binary classification in 2D and you are asked to build the best SVM classifier for each dataset. You need to download `HA_SVM_Student.py` and `svm_2d_data.mat`

Most of the implementation is provided in file `HA_SVM_Student.py`

It loads the datasets, plots the training data, and then plots the resulting decision regions (in color), boundaries (in a solid line), margins (in dotted line) and the support vectors in double circles. It also plots the test set after the classification is done (in faded hue). These tools are provided to analyse your choice of the kernel (among ‘linear’, ‘poly’ (quadratic), and ‘rbf’), slack parameter C and gamma parameter for the width of the RBF kernel when used. Running the provided code will present the training data. You need to uncomment the bottom part of the code and insert the corresponding parameters for getting the plots associated with the classifier. The code also prints out the test accuracy. Try to reason about the type

of the kernel using the shape of the classes and not by enumerating possibilities to maximize the test accuracy.

Question 3:

No programing is required in this question.

Assume the following kernel normalized linear kernel

$$K(x, x') = \frac{x^T x'}{\|x^T\| \|x'\|}$$

3.1 What are the feature vectors corresponding to this kernel?

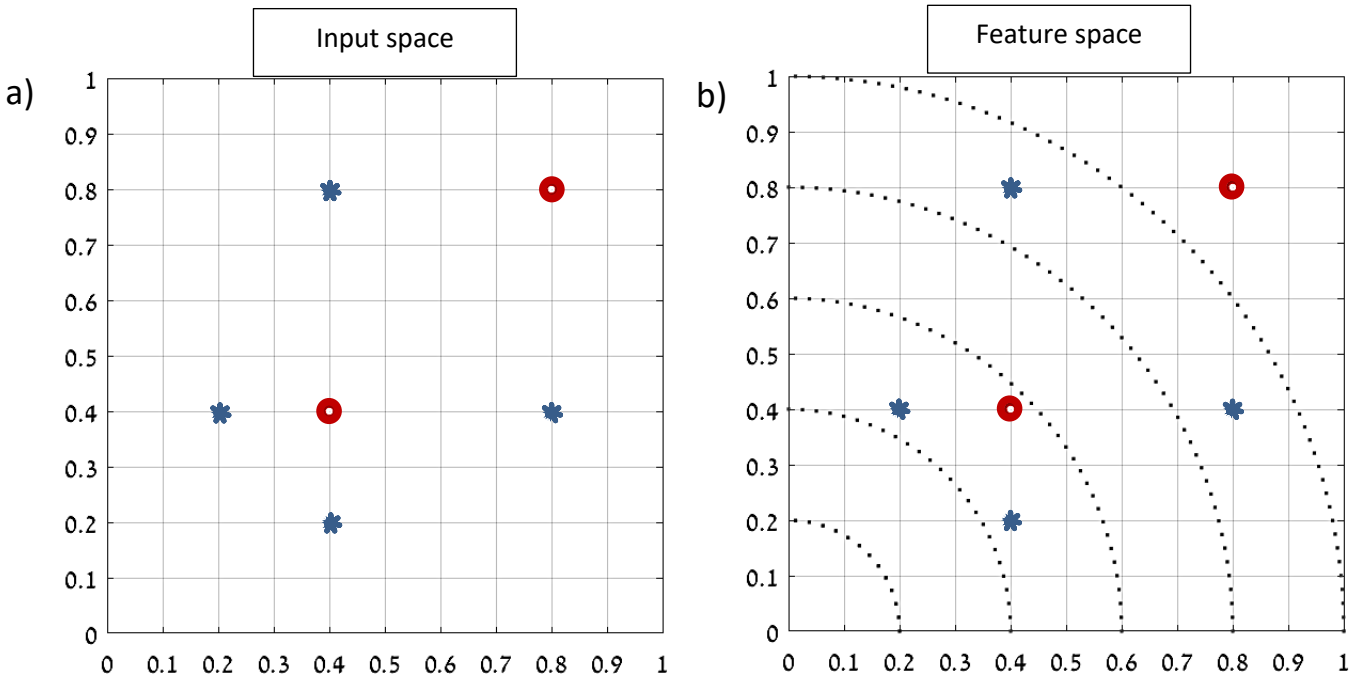

Figure 1

3.2 Using Figure 1b (right) graphically map the points to their new feature representations using the figure as the feature space

3.3 Draw the resulting maximum margin decision boundary in the feature space. Use the same Figure 2b (right).

3.4 Does the value of the discriminant function corresponding to your solution change if we scale any point, i.e., evaluate it at sx instead of x for some $s > 0$?
(Y/N)

3.5 Draw the decision boundary in the original input space resulting from the normalized linear kernel. Use Figure 2a (left).