OP08 Pattern Recognition – Machine Learning

2theWork

Job Type:***Individual***

Delivery date:***Sunday21/05/2023, 23:55 (No extension will be granted)***

Delivery method:***Exclusively through himeclass***

Total Points:**100**(15% of the final course grade)

Work is**individual**and consists of 2 questions. It is highly recommended that you take the time to understand the logic behind the questions in the paper and avoid looking for ready-made solutions on the internet. However, if you consult and/or use any material and/or code available on the Internet, you must properly cite the source and/or link to the website from which you obtained the information. In any case, copying part or all of the work is not acceptable and in the event that copying is found, all parties involved will be zeroed out of the course.

You should submit**a single fileIPython notebook (Jupiter notebook) via the eclass tasks tool**, following the following naming convention for your file: ***Surname_Registration Number.ipynb***

You can use heading cells to further organize your document.**Important:**The IPython notebook you hand in should make sure it opens and runs in google colab.

[Question 1: Face recognition]*(70 points)*

In this query you will apply the Eigenfaces method (ie a combination of PCA for feature extraction and nearest neighbor classifier for face recognition). You will use face images from the Yale B face database which contains 10 faces photographed under 64 different lighting conditions. Using your implementation, you will evaluate the ability of the Eigenfaces algorithm to handle lighting conditions of the test set images that differ from those of the training set images.

The data is available in the faces.zip file in the Documents directory in eclass.

The Eigenfaces method for face recognition includes 3 main steps:

Step 1: Each dimension image50 x 50 pixels of the training set is converted into a vector of dimension 2500 elements and stored as a column in the training dataset X. Then we apply principal component analysis (PCA) to the training dataset and extract the d principal components. The d eigenvectors when transformed and visualized as images are called Eigenfaces.

Step 2: We display the images of the training and control sets in spaced-dimensional and in this way we extract low-dimensional features (ddimensional features). The low-dimensional space d is called the eigenspace.

Step 3: Face recognition is done in the eigenspace using a (one) nearest neighbor classifier with Euclidean distance as metric.

From the Yale B Person Dataset you will use the following subsets:

Set_1: person*_01.png to person*_07.png (ie the first 7 images of each person) Set_2: person*_08.png to person*_19.png
Set_3: person*_20.png to person*_31.png
Set_4: person*_32.png to person*_45.png
Set_5: person*_46.png to person*_64.png

Wanted:

I. Write a function loadImages(path, set_number) which takes as input the path in which the folder of images is located, e.g. loadImages("C:/images", "Set_1"), reads the images and returns an array of data according to set_number, where each image is represented as a column vector. The function also returns the categories (labels) to which the different images belong encoded with integers (eg 0 for photos belonging to person_0, 1 for photos belonging to person_1 etc.).

II. Train the Eigenfaces method with d = 9 and d = 30 using all images in Set_1 (70 images) and recognize the faces in Set_1 to Set_5.
For each Set and each value of the dimension d report the classification accuracy. For Set_1 we expect 100% classification accuracy as it was used to train the Eigenfaces method. Comment on the possibility of generalizing the method to the different Sets.

III. Visualize (in image form) the 9 top eigenvectors obtained after training the Eigenfaces method on Set_1. What do you notice? What could we say the different eigenvectors represent?

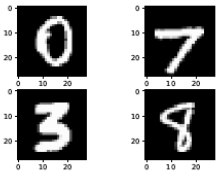
IV. Use d = 9 and d = 30 Eigenfaces found from Set_1, to reconstruct a random image from each of the 5 Sets. Plot both the original and reconstructed images for different values of d.
Comment on the reconstruction quality of each image.

V. Plot the 9 principal singular vectors that result after applying SVD to the data matrix of Set_1. Are singular vectors different from the corresponding eigenvectors? If so, why?

Note: You can use either ready-made implementations of itPCA or implement it using modal analysis (eig-type functions) in the covariance matrix. It is suggested to pre-process each image by subtracting its mean value and dividing by the standard deviation of its values.

[Question 2: Image classification using SVMs] *(30 points)*

In this question, you are asked to investigate the performance of support vector machines on the problem of handwritten digit recognition. For this purpose you will use the MNIST dataset and algorithm implementations of the scikit-learn library. The MNIST dataset consists of 70000 images of handwritten digits and is typically divided into three subsets: training set (50000 images), validation set (10000 images), test set (10000 images). Each image is 28 x 28 pixels and depicts a handwritten digit. Examples of such images are depicted in the figure below:



- You are prompted to load the set dataMNIST and convert each image into a vector format of dimension 28 x 28 = 784. Then normalize the data to the interval [0,1].
- AtSVMs have several options that can affect their performance in classification problems. Examples of such options are the kernel type and the values of various (hyper)parameters. You are asked to examine the performance of SVMs for linear SVMs and RBF kernel and different parameter values to determine the parameter/kernel combination that leads to the highest classification accuracy. For

this experiment to use 60000 images for training (training) and 10000 examples for tests (test). Report the parameter values, ie kernel type, values of C and gamma that lead to the best performance in both the training set and the test set.

- Then applyPCA the data by choosing 3 different values for the retained variance and for each variance value run the SVM method again using the parameters that led to the best performance in the query above. For each run, report the number of components retained as well as the classification accuracy. Also, record the execution times of each experiment and draw conclusions about a possible trade-off between classification accuracy, dimensionality reduction, and algorithm execution time.