***Machine Learning Homework 3***

Topic: PCA and LDA

|  |  |
| --- | --- |
| Name: | **Abu Ayub Ansari Syed** |
| UTA ID: | **1001297330** |
| Course ID: | CSE 6363 |
| Email: | abuayubansari.syed@mavs.uta.edu |
| Due Date: | 12 Dec 2016 |

**Implementation**:

**Formatting the data:**

I have used the previously formatted data i.e. from images to csv file. I have also added the python file that does the same. The complete data is in the *attfull.csv* file.

**Resize:**

I have also added the python file that resize. I have used python library PIL to accomplish the task. The resized data is in the *atthalf.csv* file.

**K Fold:**

I have chosen the eight elements from each class into the train matrix. I have chosen the two elements from each class into a test matrix. These change in each fold so as to use all the data in pair of two in test in each fold. 5 folds in two samples in test complete the data.

**KNN:**

I have implemented 1NN and calculate the accuracy by comparing (Euclidian Distance) the actual test labels with the predicted labels.

**SVM:**

I have used the train/test matrix as a parameter to the *trainSVM* function. This function creates the required parameters for the quadratic programming language i.e. **quadprog**. The solution **α** is used to calculate the w and w0. These forms the 40 linear boundaries to all the classes. The samples are tested one by one, by substituting them to the linear equation. We take the max out of the all the 40 comparisons are classify them to the respective class. We them compare with the actual class and count the correct predictions.

**PCA:**

After centering the data, the covariance of the matrix is found. Collect the top k, Eigen values and vectors from the covariance matrix. Eigen values are plotted and shown. Train data is returned as new subspace but multiplying with the Eigen vectors. Test data is also returned as a new subspace.

**LDA:**

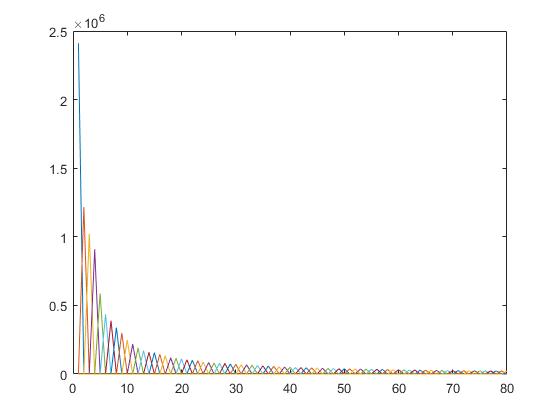
For each class, use the data to create covariance and then Sw. Add all the Sw of all classes. After centering the data, the covariance of the matrix is found. Find Sb, by subtracting Sw from the covariance. Collect the top k, Eigen values and vectors from the **inv(Sw)\*Sb**. Eigen values are plotted and shown. Train data is returned as new subspace but multiplying with the Eigen vectors. Test data is also returned as a new subspace.

**Accuracy:**

The accuracies is calculated in each of the fold. The Final accuracy is the average of the five folds.

**Conclusion:**

**For PCA, Plot the eigenvalues out and select the cut down value by the figure.**



From this figure we can see that, any values from 30 to 80 will give same analysis for PCA. Therefore, I have chosen k=80 for my experiments.

In case of LDA, I have chosen the standard value for K, i.e. No of Classes-1. Therefore, K=39.

**Is KNN or SVM more sensitive to the high dimensionality of data?**

The KNN and SVM performance are recorded as shown below. We can see that the KNN is less sensitive to dimensionality reduction. This can be inferred with task 1 and 2, even when the data dimension is reduced the performance still remains the same. Similarly, in task3 and task 4, When LDA is performed after performing PCA, the performance still remains almost same.

Whereas, SVM has highest accuracy as it uses a kernel. SVM is more sensitive for dimensionality reduction. This can be inferred with task 5 and 6, when PCA is performed the data the accuracy decreases.

Concluding, KNN is more sensitive with high dimensionality as it gets better with lesser feature. SVM is having better performance with high dimension data.

**Execution:**

* Program codes are the task numbered. Ex task1.m, task2.m….
* Keep the data in the same folder as the MATLAB file.
* The data locations are hard coded.
* Coded in MATLAB- R2016b, Linear SVM using optimization tool for Quad Prog function.
* Straight forward execution of the MATLAB file is sufficient.

**Results**:

|  |  |  |
| --- | --- | --- |
| **Task** | **Program** | **Accuracy** |
| 1 | PCA KNN AttFull | 96.75 |
| 2 | PCA KNN AttHalf(Resize) | 97.00 |
| 3 | LDA KNN | 89.00 |
| 4 | PCA LDA KNN | 94.75 |
| 5 | Linear SVM | 98.75 |
| 6 | PCA SVM | 96.75 |

**Files**:

The files in the same folder have few experimental results. The files are as follows:

Imagedatatoxlsheet.py: Python file needed to convert the Images to pixel data in CSV file.

Attfull.csv: Full data of the images into the matrix form.

resize.py: Python file needed to resize and convert the Images to pixel data in CSV file.

Atthalf.csv: Resize data of the images into the matrix form.

AttLabels: Class labels put in CSV file.

Task#.m: tasks of the homework

Pcatrain.m: pca for the training data

Pcatest.m: pca for the testing data

Ldatrain.m: lda for the training data

Ldatest.m: lda for the testing data

Knn.m: 1NN implementation.

LinearSVM.m, trainSVM.m, testSVM.m: Linear SVM with Quad programming (Homework 2)

Att\_faces(Folder): actual images for Dataset.

Plots(Folder): plots of Eigen values for tasks. Names of files are self-explanatory.

Results(Folder): Command line of the execution of the tasks.