Homework 1 | Linear Clasification and Subspaces

Course: CS7616
Due: Sun, Feb 7, 11:55pm

Problem

In the lectures we discussed linear classification and use of subspaces. Now we are going to write and train a set of classifiers and apply it to datasets below. We also discussed normal distributions and you are going to learn normal distributions from training data through Parametric Density Estimation and apply it to the rest.

- You must write PCA, LDA, and a Naive Bayes linear classifier. You will use PCA for data compression and LDA as a linear classifier. You are allowed to use functions that do Eigen decomposition and the SVD but you MAY NOT use any functions that implement PCA, LDA, or linear classification for you. You may use all basic vector and matrix math functions.

For this Problem Set you would be required to do the following for each of the two datasets listed below.

For Wine Dataset

1. Take a training set of **five** randomly chosen examples of each class and test on the rest of the data.

- Take a training set of **fifty** randomly chosen examples of each class and test on the rest of the data.
- Do **10-fold** cross-validation on the complete given data.

For MNIST Dataset

1. Use the test/train split given by the Kaggle link. You may wish to take a subset of the training data for validation as it is improper to tune your parameters on the test set.

For Naive Bayes, we are going to build a classifier by comparing the posterior probability for every class. For a class C_k and data sample x, we have

$$p(C_k|x) = \frac{p(x|C_k)p(C_k)}{p(x)}$$

For this problem set we are going to assume a normal form for the class conditional.

$$p(C_k|x) = \mathcal{N}(x|\mu_k, \Sigma_k)$$

Lets say,

$$\theta = (\mu_k, \Sigma_k)$$

Given a set of training data D the parameter estimation problem entails finding the "best" parameter θ given D. The method for Parameter Estimation, that we'll be using is:

- **Bayesian** Estimation (BE) - estimate the distribution $p(\theta|D)$

Refer to slides and textbook for more details on these estimation techniques. What we did not discuss in class how to obtain the prior $p(\theta)$ without domain knowledge. One way in which you can do it for this PS is by estimating a normal distribution over the complete dataset irrespective of class. So, you might keep the prior on θ the same for every class.

Datasets

Dataset	Description	Data
Wine Data Set	[Description]	[Data]
MNIST	[Description]	[Data]

Submit

Submit a your code and a PDF report with results in a zip file. You can use any programming language, but we will prefer `Python` or `Matlab`. The code should be easily runnable. Please include a README.md file that describes how the TAs are to run your code.

We expect the report to have two parts, each one corresponding to each dataset (approximately 1-1.5 pages each). The Report should contain the following for each of the three datasets.

- A concise description of the methods you used that seemed to work.
- For PCA/LDA, display the most important eigenvector and the 20th eigenvector (show these as an image for MNIST).
- The accuracy you achieved for each of the cases described organized in tables and figures (if that helps).
- For PCA/LDA, plot the sorted cumulative sum of eigenvalues (divide each one by the total). Pick how many eigenvalues to retain for your compression. Explain how you picked that number (you may want to reference the former graph).
- For PCA/LDA, reconstruct a test example and show the reconstruction error. For MNIST, visualize this.
- A confusion matrix on the results of your algorithm on the test data
- Mandatory is an accuracy table that has three rows (for each training case) and two columns (Linear & LDA)
- A short description of the results.
- A succinct explanation of the why you think you achieved those results. Hypothesize why certain techniques or cases worked out better than the other. If you have data/plots to support your claim, even better.
- If you are using a library from somewhere else, please mention it here as well. We hope you use Piazza for this so more people benefit.
- You are encouraged to discuss and help others with anything short of giving them your code. There are many refrences online, especially with MNIST. However, you MAY NOT use code from the internet.

Suggestions

Although it hasn't been discussed yet, data pre-processing forms an integral part of performing pattern recognition. We will discuss here a few easy to perform techniques that help. Some features may not be very useful in helping determine the class, they may be uncorrelated, or linearly dependent on others, or just random noise. So, you might like to try to all possible subsets of the set of available features. Another technique, "whitening", normalizes each of the features by subtracting each feature(dimension) by the mean and divide by the standard deviation. It is useful when your features are at very difficult scales. As a caveat, these are only suggestions, you don't have to try them and they might not even help.