CSci 5521 Homework 2
Due: 23 October 2018 at 11:59 PM CDT

- You are encouraged and expected to do this assignment in groups of two. You must use MATLAB for this assignment. Only **one** group member should upload solutions to Canvas. However, **both** group member names, student IDs, and emails must be on the first page of the solution. When submitting, please combine all the matlab files into one `.zip` file.
- You must implement the solution yourself for computing the projections and discriminate functions.

The `Digits089.csv` dataset consists of 3000 data points, each representing an image of the three digits 0, 8, 9. Each row in the file is one data sample consisting of 786 numbers in "CSV" format:

flag	label	. . . 784 pixel values . . .
(1,...,5)	(0,8,9)	(values 0,...,255)

Separate the data into three parallel arrays one with the flag values, one with the labels, and one with the pixel values representing 28×28 images of the digits `0, 8, 9`. Use the entries with flags `1, 2, 3, 4` as training data and samples with flag value `5` as a test set.

1. (*10 pts*) Consider the training set extracted from this dataset for unsupervised dimensionality reduction using PCA: Implement PCA and plot all data points using only the first 2 principal components. Display the plot of the points projected onto the first 2 principal components. Distinguish between the three classes (`0, 8, 9`) with a different plot symbol and color.

2. (*30 pts*) Consider the same dataset (training data only) for supervised dimensionality reduction using LDA: In the steps below, consider only those points corresponding to `8` and `9`.

   Apply PCA to the training samples and project onto the first 2 principal components. Then, apply LDA to project onto 1 dimension.

   Use the LDA projection to give classifier minimizing the error-rate on the 1D projection from LDA. That is, determine the separator using only the training set and then report the confusion matrix for both the training and test sets separately. The 2×2 confusion matrix contains (number of `8`'s classed as `8`, number of `8`'s classed as `9`, number of `9`'s classed as `8`, number of `9`'s classed as `9`).

   Draw the plot of the first two principal components for just `8` and `9` and include the line separating the two classes computed from the LDA projection.

   Also show the images for the "most incorrectly" classified examples. That is, show the image for the `8` test sample whose LDA projection lies the most in the `9` direction and the image for the `9` test sample whose LDA projection lies the most in the `8` direction.

3. (*30 pts* Repeat the previous LDA classification, but this time use enough principal components to capture 90% of the variance in the training data. You can compute the needed number of components separately and just report the number.

4. (*30 pts* Implement the K-nearest-neighbor algorithm and apply it to the same dataset as in the previous problem, using the PCA needed to capture 90% of the variance. Try number of neighbors $k = 1, 3, 5, 7, 9$ and choose the k yielding the lowest error rate on the training set. Then report the resulting confusion matrices for both the training and test set using that best k. Also show the images for your choice of two misclassified test samples, one for a misclassified `8` and one for a misclassified `9`.

5. (Extra Items – to explore further analysis)

   - Use training examples from all three digits to learn a classifier and show results on the test set.
   - Adjust the parameters (number of dimensions in PCA, number of neighbors in KNN) using 4-way cross-validation on the training set.
   - Try PCA, LDA and/or KNN on the entire data set, available at "`https://pjreddie.com/projects/mnist-in-csv/`". Note: the data at this web site stores the one `label` and the $784 = 28^2$ `pixel values` per row; there is no `flag` entry.

Instructions

Follow the rules strictly. All code must be written in MATLAB. If we cannot run your code, you get 0 points.

- **Things to submit**

  1. `hw2_sol.pdf`: A document which contains the solution to Problems 1, 2, and 3 including the summary of methods and results and the PCAplots from problem 1 and the LDA/PCA plot from problem 2. The front page of the PDF file should have names and UMN email addresses of the student(s) submitting the document. Also include any experiments and results you carry out in problem 4.

  The following is to be zipped into a single ZIP file:

  2. `myLDA.m` starting with `function [Projection, classification]=myLDA('`*filename*`',l);` where *filename* is the name of the file containing the data, `Projection` is a 2000×2 matrix of projections of all the `8`'s and `9`'s onto the first two principal components (exactly what is plotted in the 2D plot), `classification` is the predicted label `8` or `9` from the LDA classifier, and `l` is the number of principal components to keep for the LDA step. You can figure the number of principal components needed to capture 90% of the variance separately. The PCA can be computed within the myLDA function or in a separate function, as you prefer, but be sure to identify the code computing the PCA with some comments.
  3. `myKNN.m` starting with `[classification]=myKNN('`*filename*`',l,k);`
     where `classification` is the predicted label `8` or `9` from the KNN classifier, `l` is the number of principal components, and `k` is the number of neighbors to use in the KNN classifier.
  4. Any other files, except the data, which are necessary for your code.