

# Computer Science in Practice

## Exercise No. 3

Due: Tuesday, June 5, Ross closing time

**This exercise should be submitted in pairs. Submit your code electronically in a file called `fisher.m` (other formats will not be checked) containing only your code for question 1 through the web site of the course. Submit the results of your experiments to the course box at Ross -2. Write clearly your name, login name and id on your submission. Stick to the prototype defined in the exercise (otherwise your exercise will not be checked).**

1. Implement the Fisher Linear Discriminant Analysis in Octave: Write a function that will receive a set of training examples (feature vectors and their labels) and return the vector  $w$  which is the linear classifier.

You need to implement 3 different methods for finding  $w$  (the desired method of calculation will be selected by a parameter of the function): (1) Calculate the generalized eigenvalue of  $A$ ,  $B$  ( $A$  and  $B$  are the symmetric positive definite matrices that were defined in class) (you can use the function `geig` that we supply). (2) Solve the equation  $Bw = (\mu_1 - \mu_2)$ . (3) Perform linear regression on the labels  $y$  ( $w^* = \arg \min_w \|Xw - y\|_2^2$ ).

The prototype of the function should be as follows:

```
[w] = fisher(X,y,alg);

% [w] = fisher(X,y,alg);
% Find the Fisher linear separator w (a column vector).
% X is the training set (X is a matrix. Each row of X
% is a vector containing the features of a single sample). y is
% a column vector with the labels of the training set (1 and -1).
% alg is a number between 1 and 3 that specifies how to find w:
% alg = 1: w is the generalized eigenvalue of A,B.
```

```
% alg = 2: w is the solution to  $Bw = (\mu_1 - \mu_2)$ 
% alg = 3: solve the corresponding regression problem
```

The function should be in a file called `fisher.m`

A simple way to test your implementation is to separate two 2D normal distributions: draw 1000 samples from a normal distribution with mean  $(-1,-1)$  and covariance matrix  $I$  and draw another 1000 samples from a normal distribution with mean  $(1,1)$  and covariance matrix  $I$ . In this case,  $w$  should be close to  $(1,1)$  and the separation will be good but probably not perfect. (This is only for you to test your code, do not submit this result).

2. Test the performance of the Fisher linear separator using the datasets in the files `ex3_data_1`, `ex3_data_2` and `ex3_data_3` available on the course web site. The files contain pictures (or subsets of the pictures) of two handwritten digits (2 and 9). We will use the raw pixels as our features for classification. We will test the influence of the number of features on the classification by taking different subsets of the pixels and comparing the train and test classification errors.

The files contain 3 datasets, each of which is divided into train and test sets (load the first file, for example, with `'load -binary ex3_data_1'`).

Each dataset consists of a training set  $X_{\text{train}}$ ,  $y_{\text{train}}$  and a test set  $X_{\text{test}}$ ,  $y_{\text{test}}$ . Each row of the matrices  $X_{\text{train}}$  and  $X_{\text{test}}$  is an image (you can view the image at row  $i$  of  $X_{\text{train}}$  (in the second dataset, for example) with `'imshow(reshape(X_train(i,:),20,14));'`). The first 5000 rows of  $X_{\text{train}}$  and the first 1032 of  $X_{\text{test}}$  contain the images of the digit 2, and the rest contain the images of the digit 9. The vectors  $y_{\text{train}}$  and  $y_{\text{test}}$  contain the corresponding labels.

For each dataset, find its classifier  $w$  and choose the threshold  $b$  that minimizes the number of errors on the training set (explain briefly how did you find it). Then count the number of classification errors for the test set and for the train set. Write the results for each of the datasets (the number of classification errors for both the train set and the test set) and explain briefly their meaning (why is the number of classification errors higher or lower in the test and train sets between the different datasets).

Compare the performance of the 3 algorithms. Refer to the following points: (1) Do you get the same  $w$ ? (2) Compare the run time of the 3 algorithms.

3. A list of things to submit:

- (a) The file `fisher.m` with your implementation of question 1. Your implementation should conform with the prototype defined in question 1 and should work with octave on the cs machines. Submit the file electronically through the web site of the course.
- (b) A paper (printed or handwritten) with the results of your experiments in question 2 and a brief explanation of these results and how you found  $b$ . Make sure that you write the name, id and login name of both persons submitting it. Submit this paper to the course box at -2.