SSI Pattern Recognition: Homework 2
MSCV, second semester 2016
Sepideh Hadadi

I. Introduction

In this homework, we will implement logistic regression and linear discriminant analysis.
You shall submit a clearly written and commented report as well as your own code.

II. F*!& Spams !

We want to build a classifier to filter spam emails. We will use a dataset which contains a training set and a test set of, respectively, 3065 and 1536 emails. The data is given is the file `spamData.zip`. Ecah email has been processed and a set of 57 features were extrated as follows:

- 48 features, in $[0, 100]$, giving the percentatge of words in a given message which match a given word on a predifined list (called vocubulary). The list contains words such as "business", "free", "george", etc.
- 6 features, in $[0, 100]$, giving the percentatge of characters in the email that match a given character on the list. The characters are ; ([! $ #.
- Feature 55: the average length of an uninterrupted sequence of capital letters.
- Feature 56: the length of the longest uninterrupted sequence of capital letters.
- Feature 57: the sum of the lengths of uninterrupted sequence of capital letters.

The data format and the context in which is data is collected and processed is well understood and to that base following questions were coded and answered

1. What are the *max* and *mean* of the average length of uninterrupted sequences of capital letters in the training set?

Max average length = 1102.5
Mean average length =4.9

2. What are the *max* and *mean* of the lengths of the longest uninterrupted sequences of capital letters in the training set?

Max length =9989
Mean length =52.67

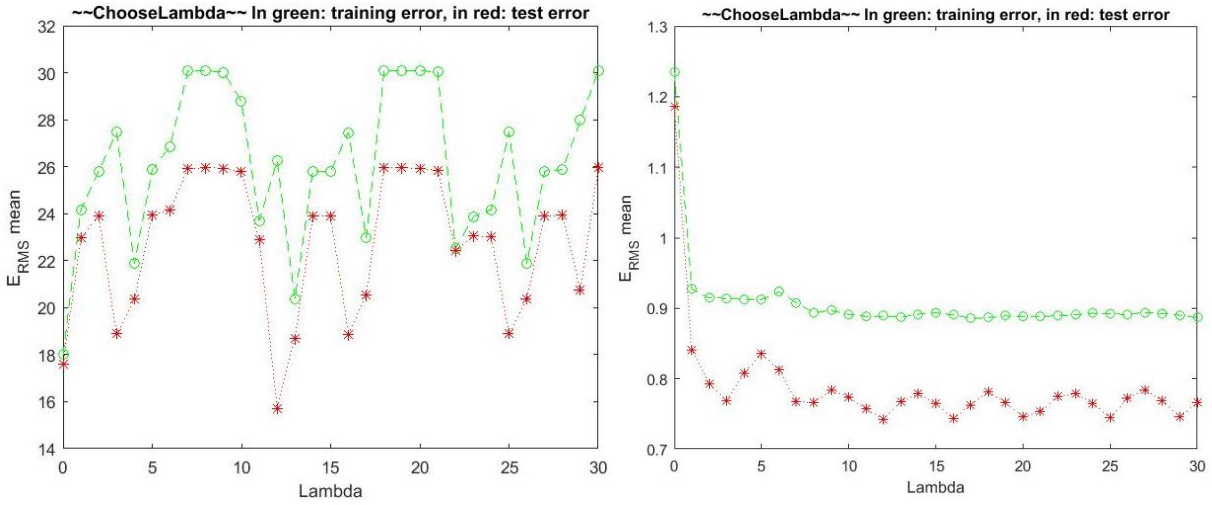
The data processing is done in the code and commented.

For each version of the data, i.e. using a different preprocessing, fit a logistic regression model.

- use cross validation to choose the regularization parameter.
- report the mean error rate on the training ad test sets.
- what is the best preprocessing strategy? why?

4. Compare with the results of a Naive Bayes classifier.

The result of the training is shown in fig.1.

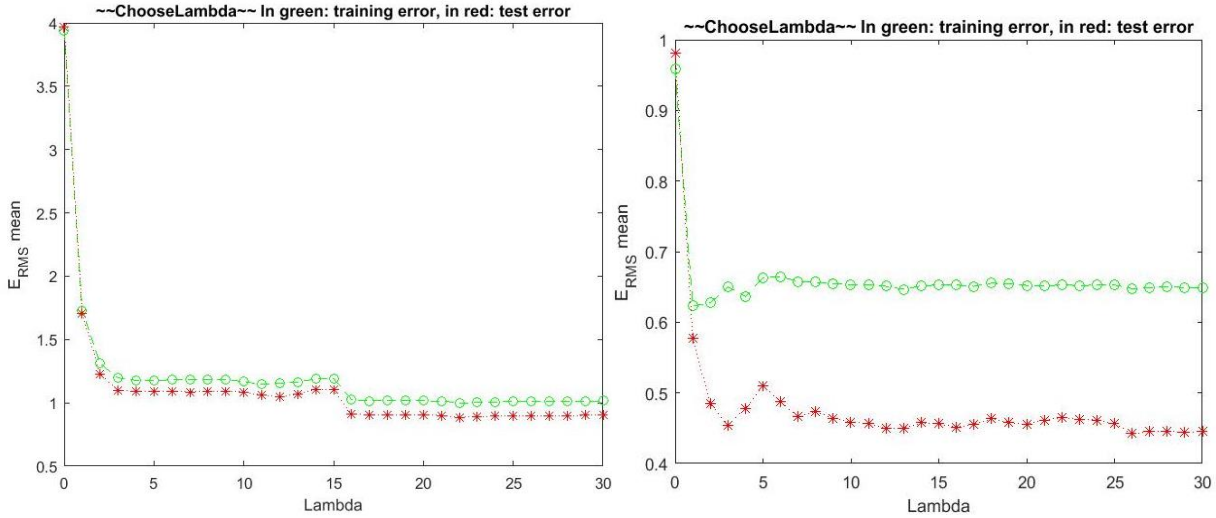


3. Before training a classifier, we can apply several preprocessing methods to this data. We will try the following ones:

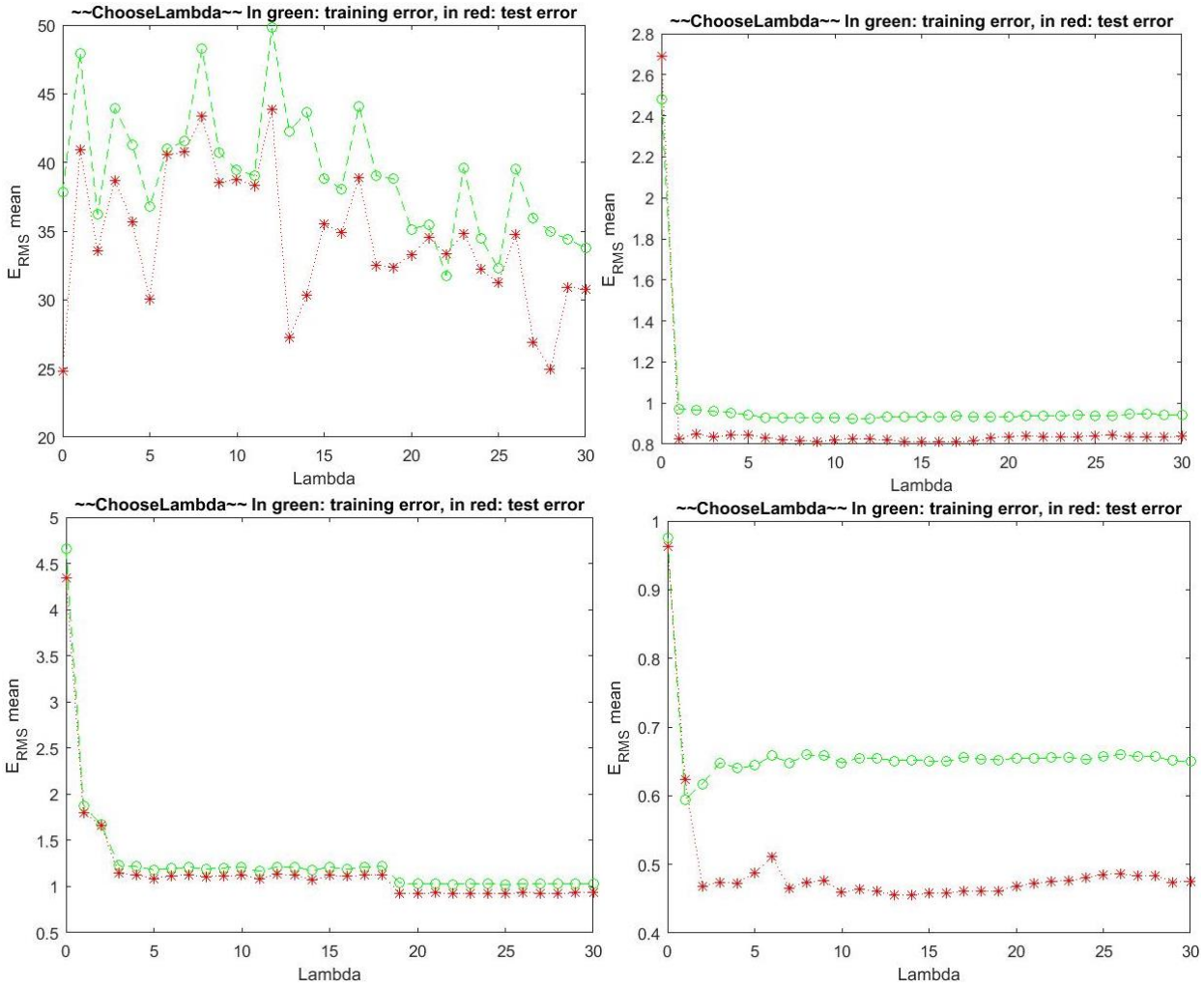
   (a) Standardize the columns so they all have mean 0 and unit variance

   (b) Transform the features using $\log(x_{ij} + 0.1)$, i.e. add 0.1 to each feature for every example and take the $\log$. We add a small number to avoid taking $\log$ of zero !

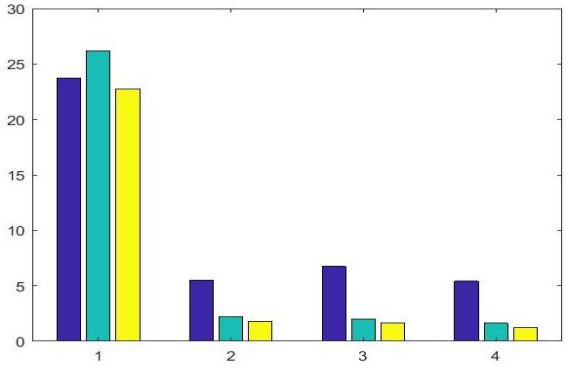
   (c) Binarize the features using $\mathbb{I}(x_{ij} > 0)$, i.e. make every feature vector a binary vector.



(a) The regression regularization using different λ. At the top from left to right there is original data and normalized data. At the bottom from left to right, the result of analysis for log and binary feature is shown. This analysis is for training data set and using cross-validation.



(b) The regression regularization using different λ. At the top from left to right, analysis of the error for original and normalized features. At the bottom from left to right, the result of analysis for log and binary feature is shown. This analysis is for test data set and using cross-validation.



(c) calculation time and number of error in train and test data set the detail is shown in the table.

	Number of error	time	Error training	Error test
original	708	23.7433522000000	26.1721027543100	22.7730405335899
normalized	167	5.52243539999995	2.18509235049899	1.79877643621093
log	626	6.73924320000003	2.01047224104412	1.62317059712819
Binary	0	5.41323469999998	1.61545264617129	1.20772325074118

Comparing the three features shows that the error in test in binary feature was the minimum as a result we can conclude that the binary feature is better feature for classification of the given date.

(d) Comparison between naïve Bayesian results and regression shows that the error rate of the naïve Bayesian is higher than normalized and binary feature using regression but is lower than original data and log feature. So as a final conclusion we can say that for the data set regression provides better result comparing to naïve Bayesian classifier.

III. Cats & Dogs !

In this part, we want to recognize dogs and cats. The files `catData.mat` and `dogData` contains respectively 80 examples of cat and dog images. Figure 1 show two images of a cat and a dog.

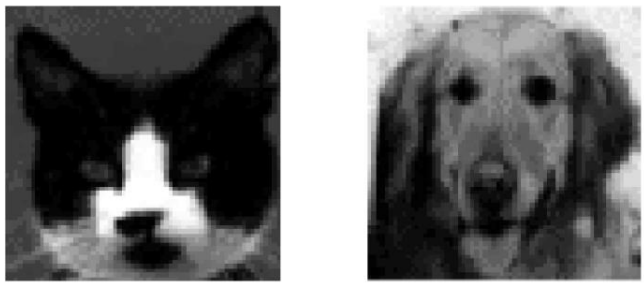


Figure 1: Example of a cat and a dog image.

In the previous task, digits recognition, we only had binary images. Now, we have more content in the images and we shal use this in building a classifier.
Your task is to

1. Represent each training example using a 'useful' feature descriptor. You should explain your choice for a descriptor.

The data is shown in 2 different formtes: image format by reshaping original data and putting them into 64x64 images. Then using SVM the eigenfaces were calculated and presented in the formate of image. These to data is shwin in the following figures.

Fig.2 shows the cat and dog images after reshaping

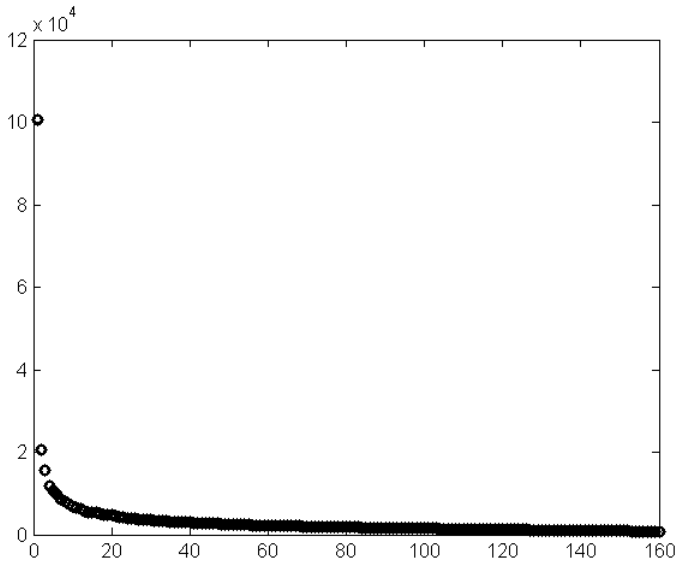


Fig.3 eigenvalues for analysis and feature selection

Eigenvalues are plotted and as seen the first feature is totally independent between dogs and cats. Therefore, any feature after this feature can be selected to use in the classifier because it has data from cats and dogs which help better classification. The for first feature is shown in Fig.4

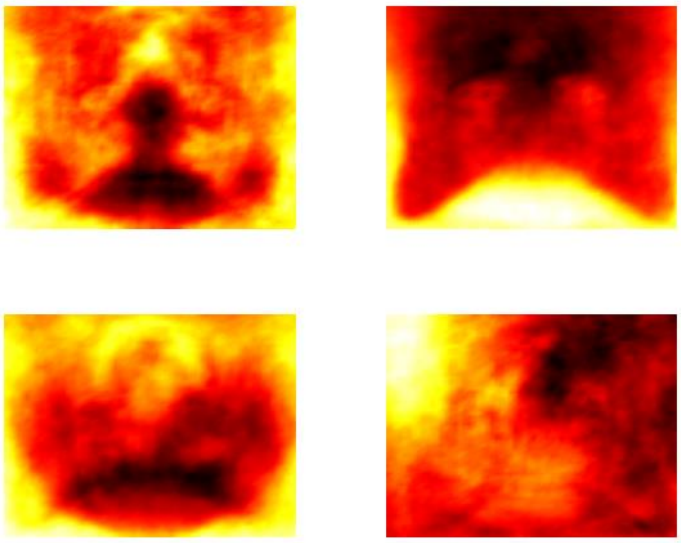


Fig.4 4 first Eigenfaces taken from dog and cat data after applying SVD

2. Train a SVM to distinguish cats from dogs using your selected features. You should apply cross-validation.

The main objective is to classify following sets into two separate classes using SVM classifier.

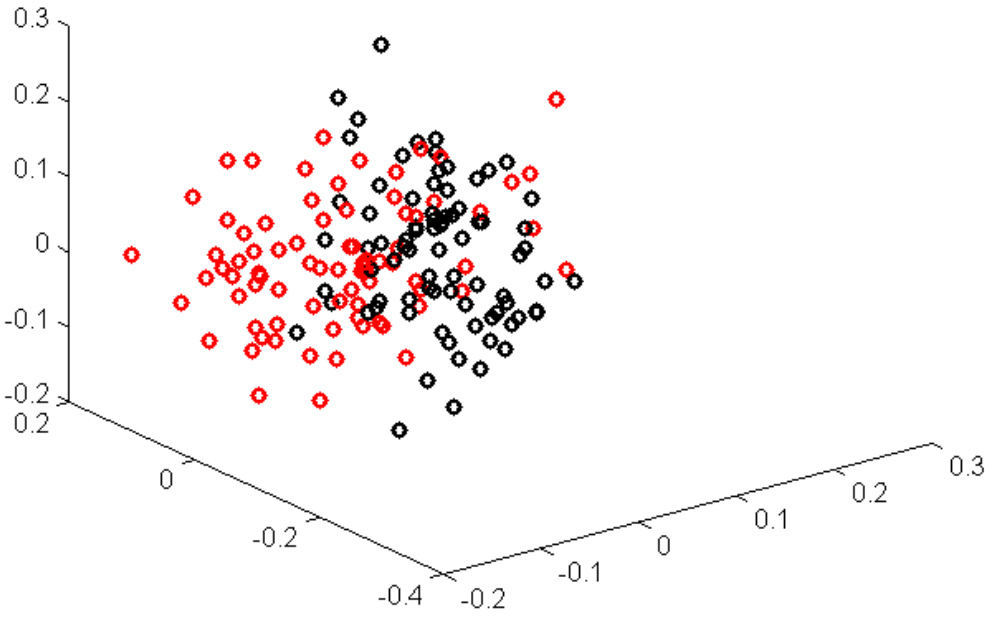


Fig.5 Feature repetition in 3D

3. Submit a report describing each step and your code.

The classifier code was written and commented which will be attached to the report.