Machine Learning Lab3

K-NN classifier

Alice Maria Catalano 5157341

Abstract—k-Nearest neighbours classifier is a non-parametric classification method which inputs are the k closest training examples of a dataset, while the output is a class membership. The goal of this laboratory is to implement this classifier in MATLAB on the specified datasets.

INTRODUCTION

a. K-Nearest neighbour classifier

The KNN algorithm is a simple and easy to implement supervised machine learning algorithm. It's non parametric so it makes no assumption about the probability distribution building a discrimination rule directly from the data.

To implement this algorithm is needed a:

- Training set X={x_1, …, $x_{l,}$ …, x_n} and a query point $\bar{x}$
- A value k

Then given the set for training with n examples, it firstly identifies the k-nn training example of the new instance and then assigns the class label with the highest number of neighbours of the new instance.

$$\{n_1, \ldots, n_k\} = \text{top-}k\|\mathbf{x}_l - \bar{\mathbf{x}}\|$$
$$y = \text{mode}\{t_{n_1}, \ldots, t_{n_k}\}$$

b. Task description

The aim of this laboratory is to develop a k-nn classifier using the mnist data set: it contains a training set, the labels of this set, a test set and the labels of this set. Three functions are also given:

"*loadMNISTLabels.m*" to load the label,
"*loadMNISTImages.m*" to load the images,
"*loadMNIST.m*" to flexibly load the data.

The first two function are anyway useless for this implementation.

The data represent handwritten digits in 28x28 greyscale images, representing the numbers from 0 to 9, the training set contains 60000 examples while the test set contains 10000.

Loading the data, for each set (training and test) the following elements returns:

1- A matrix composed by the number of examples as rows and by 784 columns (representing the number of image's pixels 28x28);

2- A column vector composed by the number of examples: it contains the labels for 1 to 10, which represents the number of digit showed in the image;

IMPLEMENTATION

a. Building the classifier

The function *kNN.m* is used to implement the classifier. It takes as inputs:

- Training set,
- Test set and its label
- The k values that are chosen are k=[1,2,3,4,5,10,15,20,30,40,50]
  And due to the long computational time, the number of observations for both training and test set where educed, namely at 6000 and 1000.

The output is the classification and the error rate, calculated comparing the predicted label with the given ones.

The code does some checks:

- on the number of arguments received (*nargin*) equals at least the number of mandatory arguments.
- the number of columns of the test matrix should be at least equals the number of columns of the training matrix.
- Check that k>0 and k<=cardinality of the training set (number of observations, referred to as n).

After that, considering as query points each observation in the test set, the classification of the test set is done.

The Euclidean distance is computed using a MATLAB function ”*pdist2()*”, computes the distance using the metric specified by distance (D) and returns the K smallest pairwise distances to observations in X for each observation in Y in ascending order. Then, using ”*mode()”* function, the most frequent value in an array is returned (classification). Having available also the label of the test set as input of the function, the error rate may be computed.

b. Test the kNN classifier

In this task is asked to use the MNIST character recognition data and to compute the accuracy on the test set in two ways:

- On 10 tasks: each digit *vs* the remaining 9
- for several values of *k*, e.g., k=[1,2,3,4,5,10,15,20,30,40,50]

According to what it is required, the results are provided for any combination of these parameter.

RESULTS

The total results are a lot and contained in the “/results” folder. Here are analysed for explanation just some of them.

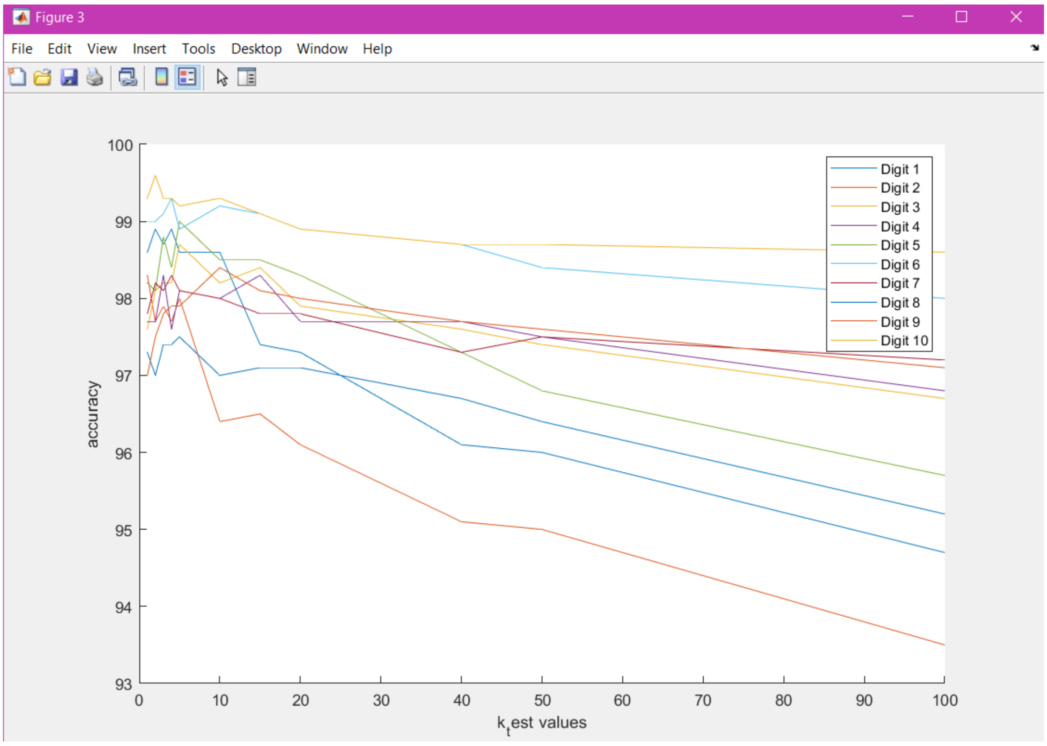

Figure 1 Accuracy for each different value of k in the digits.

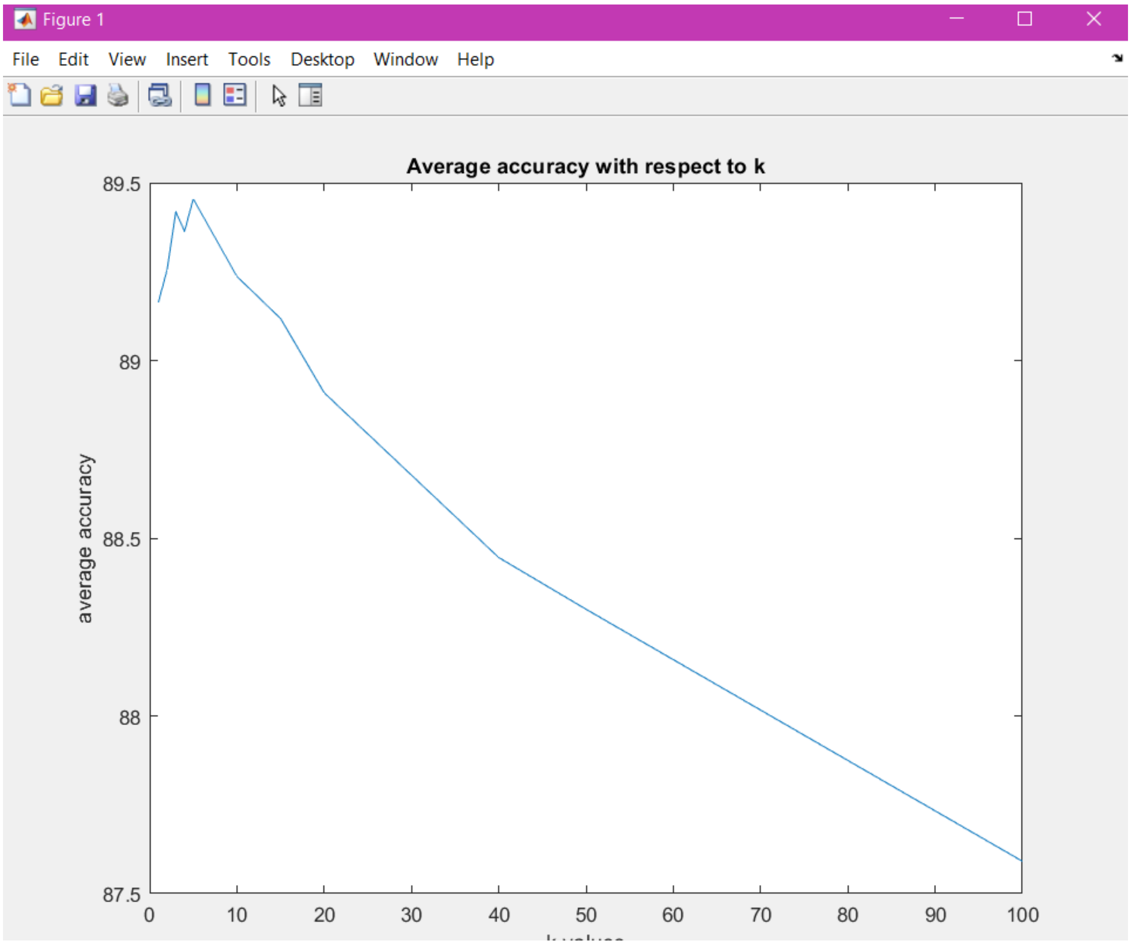

Figure 2 Accuracy average w.r.t. k.

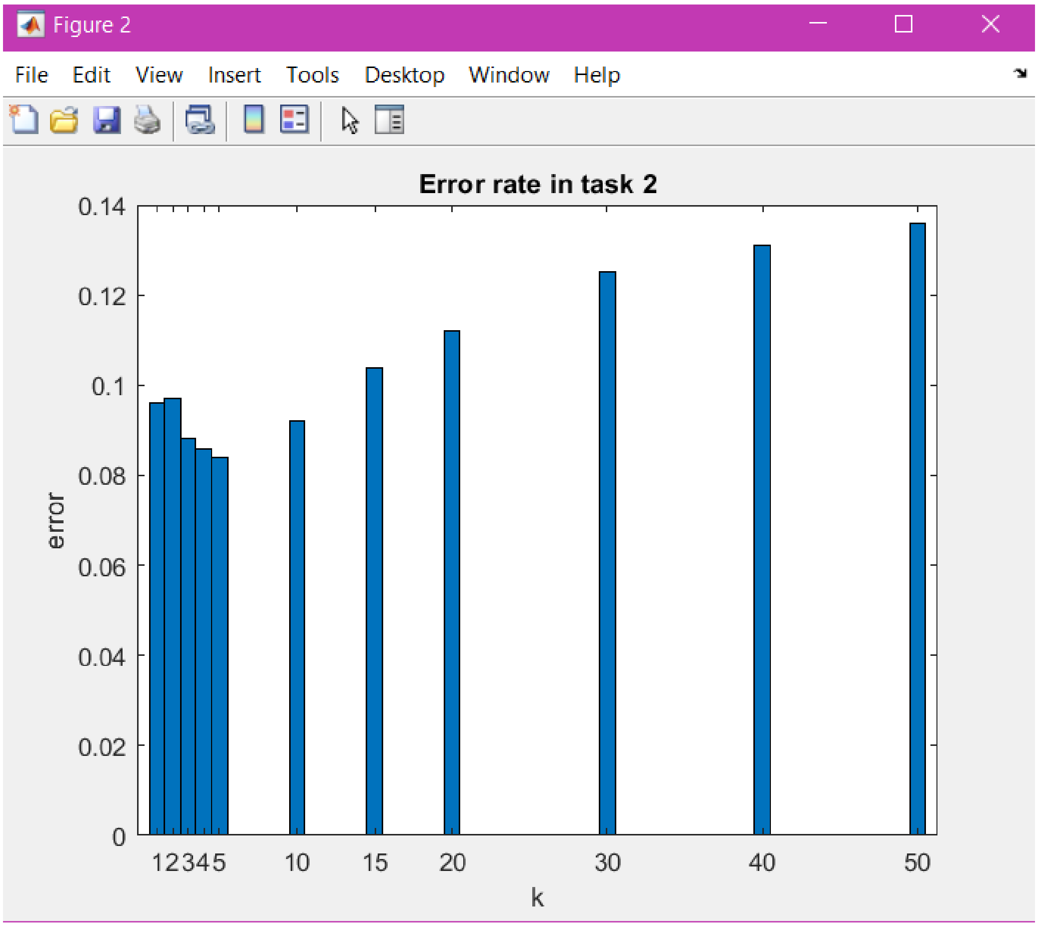

Figure 3 Error rate for each k

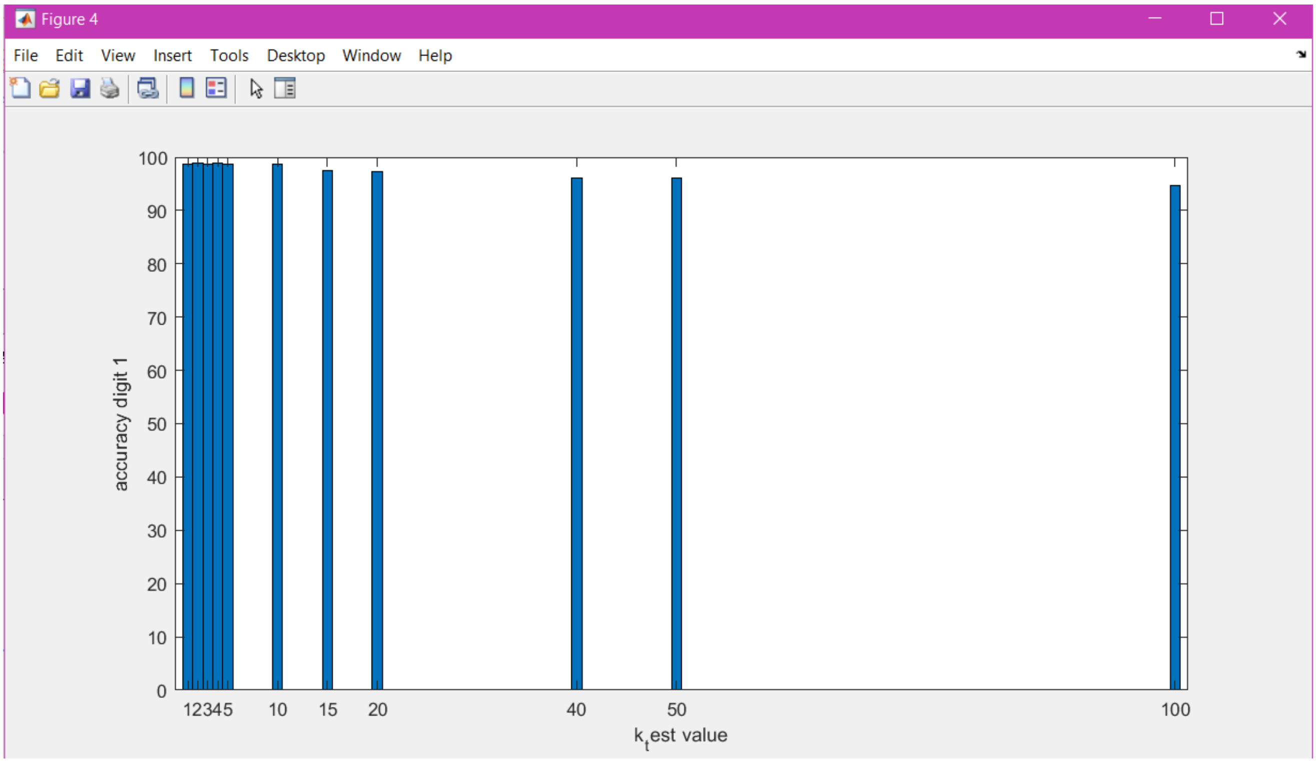

Figure 4 Accuracy for digit 1 using different values of k

The accuracy is computed for each digit, but only the bar graph of the digit 1 is showed. According to the bar graph, the accuracy of each digit with respect all value of k is very high (usually between 97/100 per cent)