Università degli Studi di Genova

Robotics Engineering

Saivinay Manda

Report of Machine Learning
Assignment.3

kNN Classifier

Contents

Chapter 1

Introduction

The goal of this assignment is to implement the kNN Classifier on MATLAB environment. This algorithm permits to classify big datasets of an object based on the k nearest objects with a particular accuracy.

1.1 k-Nearest-Neighbours Classifier

The kNN Classifier is a non-parametric method used for classification or regression problems. Differently from other type of supervised algorithms, there is not an implementation of a model with parameters, but directly a classification rule is defined from data

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. The general idea is given a training set, a query point and the values of k, we select the k training points which have a shortest distance to the query point. The distance computed is the Euclidean Distance. Then, given the test set, the algorithm predicts the class labels of them.

Summing up, the algorithm requires the following input and output:

Input:

1) A training set $X = \{x_1, x_2, \ldots, x_n\}$;

2) A query point x_0 ;

3) Value of k, which is the number of neighbors;

Output: Classification y for the query point

1) $\{n_1, ...n_k = top - k||x_l - x_0||\}$

2) $y = mode\{t_{n_1}, ..., t_{n_k}\}$

Where mode = most freqeunt item in a sample.

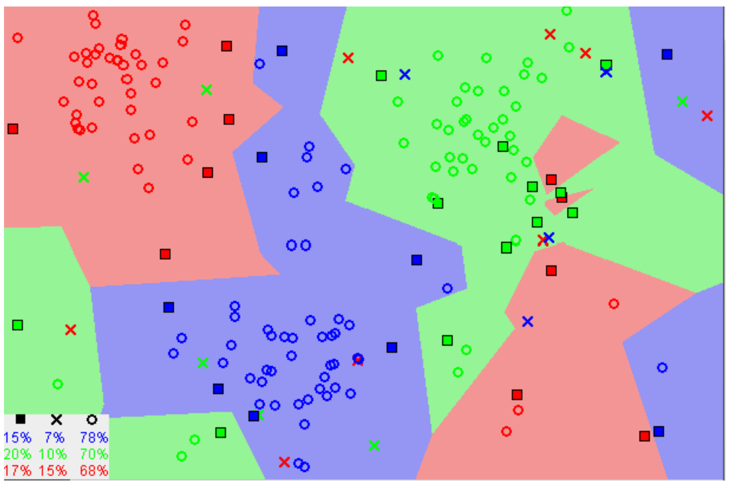

Figure 1.1: Image showing how similar data points typically exist close to each other.

Chapter 2

Description

On MATLAB, the main code is kNNClassifier.m", which uses some functions to compute our task.

2.1 Task 1: Obtain a data set

The given dataset is the MNIST database. The file contains a training set, corresponding labels, a test set, corresponding labels (for comparison) and two MATLAB functions to load data and labels, and one flexible function to load the data named loadMNIST.m". The examples are handwritten digits in 28x28 greyscale images 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 (60000 for the training and 10000 for the test), and by 784 columns (28x28 number of pixels of the images);

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.

2.2 Task 2: Build a kNN classifier

A function kNN.m" is created in order to implement kNN algorithm.

The inputs are:

1) Training set;

2) Test set;

3) k values (I used k = [1,2,3,4,5,10,15,20,30,40, 50]); 4) the label of the test set;

The outputs are:

1) the classification;

2) the error rate (doing the comparison with the predicted label and the given label).

First of all, the function checks some assumptions as well as:

1) the number of arguments received (margin) is equals at least the number of mandatory arguments;

2) the number of columns of the training set should be at least equals to the number of columns + 1;

3) the value of k must be grater than 0 and lower than the number of columns of the training set (num of observations);

After that, the classification of test set is done, considering as ”query” points each observation in the test set. The euclidean distance is computed using a MATLAB function ”pdist2()”, which returns the k smallest pairwise distances in D (distance between the observations in X and each observation on Y). Then, using ”mode()” functions, the most frequent value returns (classification).

Having available also the label of the test set as input of the function , the error rate may be computed.

2.3 Task 3: Test the kNN Classifier

The task 3 required to compute the accuracy on 10 tasks (each digit vs the remaining 9) of the test set, for several values of k. (In this case k=[1,2,3,4,5,10,15,20,40,50,100]). According to what it is required, the results are provided for any combination of these parameters (recognize 1 with k = 1,2,3,4....; recognize 2 with k = 1,2,3,4,...; and so on).

2.4 Results

Due to the long time for computation, I have reduced the number of observations of both training and test set, considering 6000 for the training and 1000 for the test. In this section we show only a part of the graphical obtained results.

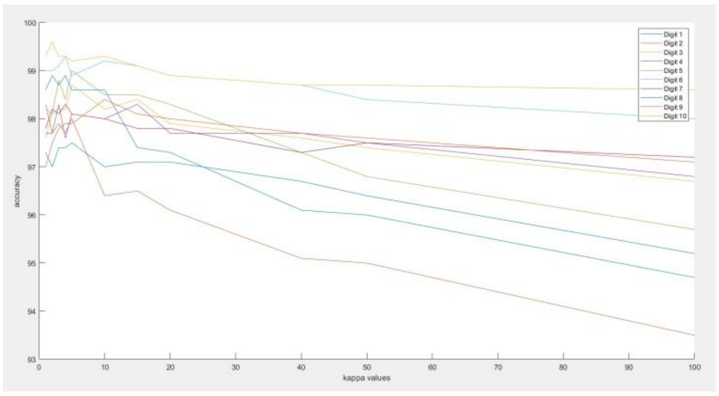

Figure 2.1: Accuracy for each digit using different value of k

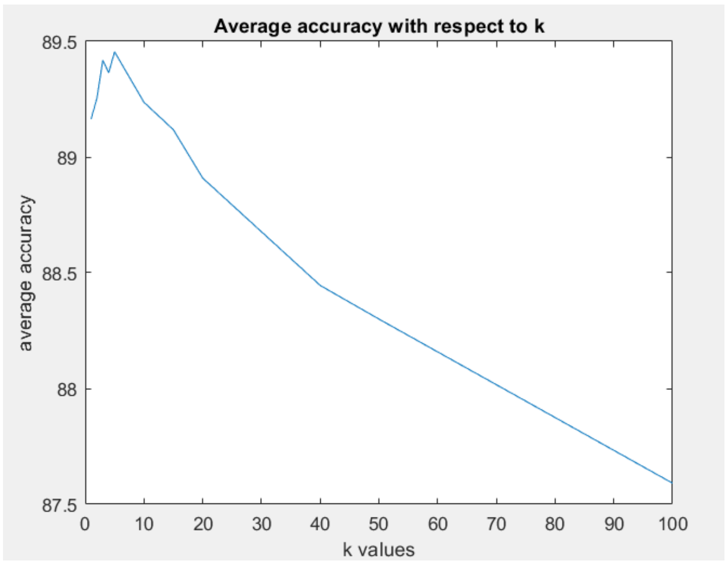

Figure 2.2: Accuracy average using different values of k

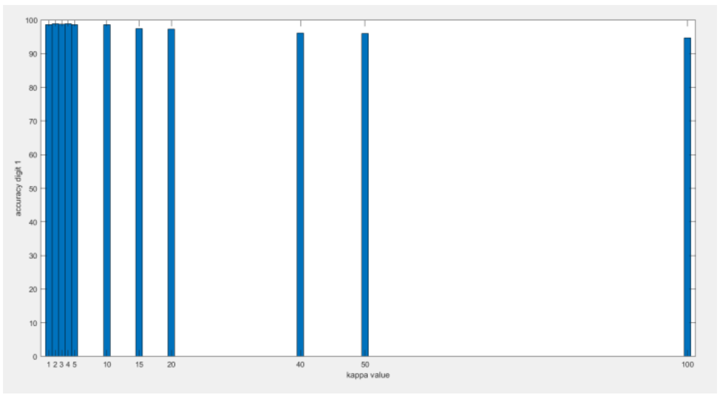

Figure 2.3: Accuracy for digit 1 using different values of k

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).

References

k-nearest neighbors algorithm (2020). Wikipedia. URL: https://en.wikipedia.org/wiki/K–nearest_neighbors_algorithm (visited on 10/20/2020).

Machine Learning Basics with the K-Nearest Neighbors Algorithm (2018). towards data science. URL: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761.

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

https://2020.aulaweb.unige.it/ pluginfile.php/276344/mod_folder/content/0/ml–2020–21––05.pdf?forcedownload= 1.