LUT School of Engineering Science

BM40A0702 Pattern Recognition and Machine Learning

Lasse Lensu

PRACTICAL ASSIGNMENT

Pattern recognition system for hand-written digits

12.12.2022

Group 12

0565878 Taru Haimi: Main responsibility for the report, put final touches to the code.

0568396 Leevi Kämäräinen: Developing and optimizing the kNN algorithm.

0522866 Joona Ylijoki: Data preprocessing (removal of “outlier” strokes), report writing (literature review, part of the experiments and conclusion).

Everybody participated for deciding and improving the model.

TABLE OF CONTENTS

1. INTRODUCTION

In this assignment the task is to develop a pattern recognition system which can recognize hand-written digits and classify them correctly. The given data of hand-written digits for experimentations contains numbers between 0-9. The digits have been “written” as free-hand strokes in the air with the index finger. The motion of the finger has been captured by using a LeapMotion sensor.

The data set contains 100 samples of each digit, also altogether 1000 samples. Every sample is three-dimensional data item, each component representing location information as numeric values. The data is given both in -.csv and -.mat files.

2. LITERATURE REVIEW

The digit recognition problem has been popular for decades because of its practicality. It is possible to utilize the solutions of this problem to automatically recognize digits from papers related to banking or other hand filled forms like taxes. Another subject where this can be used is traffic images from where the number plates of cars can be recognized automatically. Through decades the solutions for the problem have been able to achieve higher and higher accuracy by improving the solutions with new algorithms, different parameters and by preprocessing the data properly. [1]

Recognizing handwritten digits comes with its problems. People have different styles of handwriting, some have bigger handwriting than others, the way the letters or in this case the digits are drawn may vary between people, also the alignment of the digits can be something that has to be considered. The way that someone writes for example, 1 or 7 can cause confusion in some other person. This is because some are used to write number 1 as a straight line, others use the tiny barb on the top. Some of the people tend to write the number 7 with the horizontal line in the middle as others leave it out.

Some of the classifiers used to recognize handwritten digits are gaussian naïve bayes, random forests, K nearest neighbors (kNN), support vector machine (SVM), neural networks (NN) and convolutional neural networks (CNN). At least the last four are able to achieve accuracy as high as 95-99% [1,2]. The results vary with the different implementations of the classifier. Solutions that utilize neural networks usually need more time to be trained to the point where they can be used to recognize the digits. The classifier chosen for the implementation was kNN because of the simplicity to implement and the fact that it was the most familiar for the group members.

3. K-NEAREST NEIGHBORS ALGORITHM

The kNN algorithm used works by average nearest distance (from now on shorten as AND) as the distance metric. The algorithm calculates the nearest distance of every point in sample to be classified to all the datapoints in a training sample (Figure 1.), and vice versa (Figure 2.). Mean value of both the nearest distances (raised to the power of two) from classified samples to training samples and from training samples to classified samples are then summed, and those values are saved for the comparison in the algorithm. This means that for every sample to be classified there will be stored N -number of distances, where N is the number of training samples.

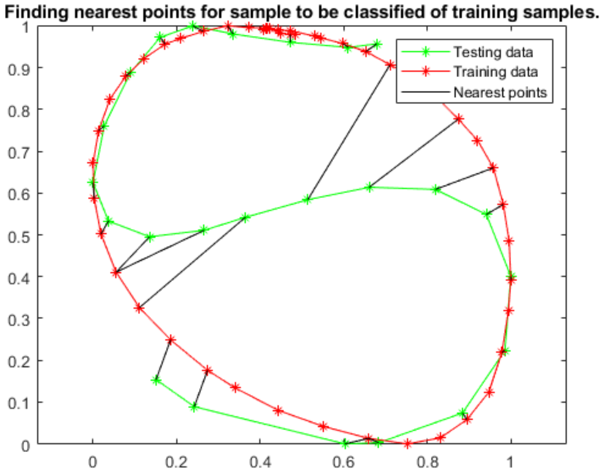


Figure 1. Example of nearest points of digit 0 (training sample) compared to digit 5 (testing sample).

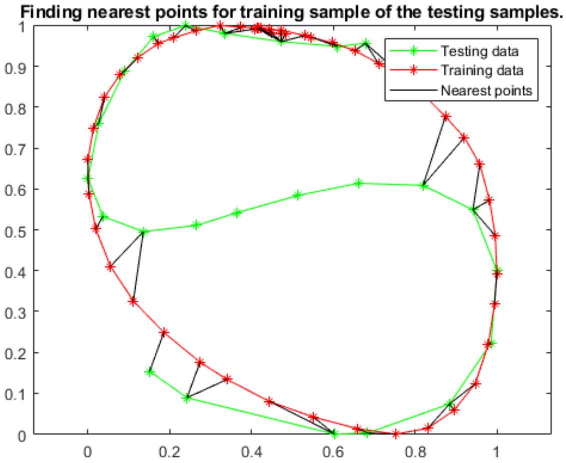


Figure 2. Example of nearest points of digit 5 (testing sample) compared to digit 0 (training sample).

When each of the samples to be classified has the ANDs calculated for the training samples, the algorithm follows the more standard version of kNN. All the distances will be sorted in ascending order and the classes will be sorted to correspond to the distances. This means that from the class matrix the first k-values will be taken into comparison, and the most common class found in there will be the one that the sample is classified into.

4. EXPERIMENTS

The hand-written digit data set is first preprocessed. Then the data is divided into training and validation sets. With training data, the suitable kNN-model is trained, and with validation data the best k-value is estimated.

4.1 Preprocessing data

To make data analysis and pattern recognition system more accurate, data preprocessing is needed. Preprocessing methods used in this practical assignment are normalization, dimensionality reduction, and outliers' removal. These methods were chosen based on the visual appearances of the strokes. Missing values were not found.

First the data is normalized with min-max -method so that every value of each stroke is scaled to between 0 and 1. Each stroke is scaled independently from each other so that when plotting, the digits appear approximately in same sizes. In addition, each strokes normalization will be done independently for the x- and y- values. Secondly, every dimension of the stroke is examined, and it has been concluded that it's meaningful to use only x- and y-dimensions in algorithm and leave out the z-dimension. This is visualized in the figure 3. Then the outliers were removed. Removals were chosen by visually comparing if some of samples look too unrecognizable, like in the figure 4 representing two quite bad samples of digit four, or if the sample looks more like some other digit than the predetermined digit. Altogether 16 samples were removed.

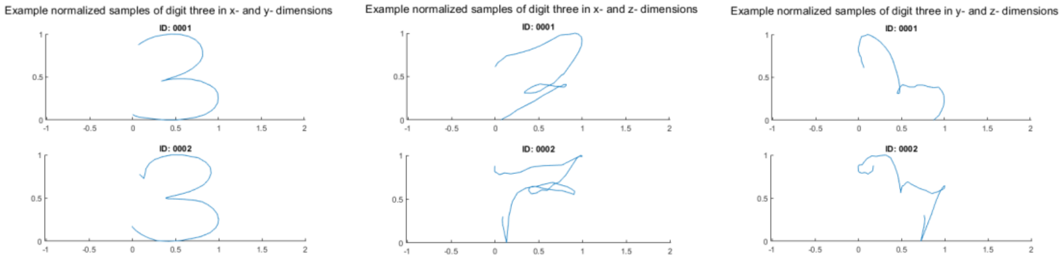


Figure 31: Examples of how normalized hand-written digit 3 looks like in different dimensions.

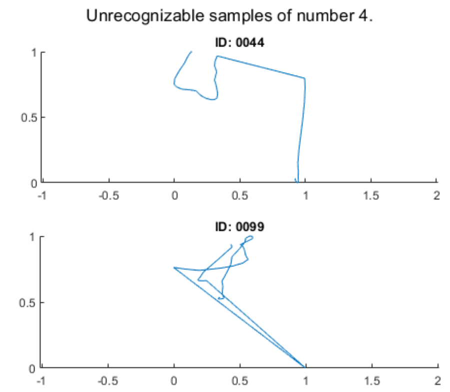


Figure 4: Unrecognizable examples of digit 4 samples.

4.2 Training the digit recognition model

Before the model training and sample testing is begun, the data set is divided into the training and validation sets. The division is performed randomly, but so that both sets have relatively as many samples of each digit. The used division threshold is 80-20.

When developing the model, at first the standard kNN-algorithm was developed with only calculating the distances between training and testing samples (not vice versa) and that worked quite okay with accuracy being approximately 60%. Then it was realized that some digits resembling each other are easily misclassified among themselves. For example, digit 3 is close to digit 8, so if the distances from points of digit 3 to digit 8 are compared, the mean distances are smaller than otherwise. That led us to consider the improvement that both distances are calculated and taken into account when trying to find the correct class.

The classification was repeated with multiple k values in order to determine the best k-value. Resulting accuracies for k-values from 1 to 10 can be seen in Table 1. Value 7 was chosen to be the best k-value because it is the highest k-value with the best accuracy achieved.

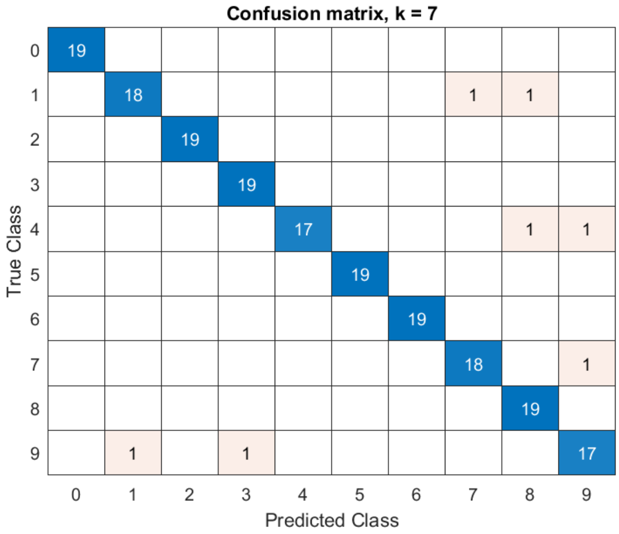
In addition, there is no right way of choosing the optimal value for k but choosing a small value of k can lead to unstable decision boundaries. Therefore, small k-values are not suitable for classification. [3]

Table 1: Estimates for model accuracy with different values of k.

K-value	1	2	3	4	5	6	7	8	9	10
Accuracy	0.9634	0.9634	0.9581	0.9581	0.9634	0.9634	0.9634	0.9581	0.9529	0.9529

4.3 Results

For the k-value 7 the digit recognition model achieves accuracy of 96.34%. The results of classifications are visualized in the figure 5. Only very few samples are misclassified.



	0	1	2	3	4	5	6	7	8	9
0	19									
1		18						1	1	
2			19							
3				19						
4					17				1	1
5						19				
6							19			
7								18		1
8									19	
9		1		1						17

Figure 5: The confusion matrix of model's accuracy with k-value = 7.

5. CONCLUSION

The results of getting the model accuracy of 96% correspond to the ones found in literature of the subject. The reached accuracy is very good when taking into account that in practice it’s not realistic to create model with 100% accuracy.

The pattern recognition model created model can probably still be improved. 1000 samples itself is not that small amount but that divided for the 10 digits, it is only about 100 samples for each digit. By increasing the size of the data set the results could be improved. What comes to the classifier, as the data set’s size increases, the calculations are heavier and thus more time consuming. This is something to consider if the classifier is used to classify larger amounts of data. Even with this mid-size data the calculations are taking some time (however still performed in a reasonable time), so the algorithm for distance calculations could be optimized in some manner. In addition, the best option for k-value could be investigated more by examining even larger k-values than just 1-10, but as mentioned earlier, there is no clear way to choose the best k.

REFERENCES

[1] Himanshu Beniwal. Handwritten Digit Recognition using Machine Learning. https://medium.com/@himanshubeniwal/handwritten-digit-recognition-using-machine-learning-ad30562a9b64

[2] Mahnoor Javed. The Best Machine Learning Algorithm for Handwritten Digits Recognition. https://towardsdatascience.com/the-best-machine-learning-algorithm-for-handwritten-digits-recognition-2c6089ad8f09

[3] Amey Band. How to find the optimal value of K in KNN. https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb