Exploration of the K Parameter in Hand-Written Digit Recognition by K-Nearest Neighbor Algorithm

IVAYLO PENEV, MILENA KAROVA, MARIANA TODOROVA
Department of Computer Science and Engineering and Department of Automation
Technical University of Varna
9010 Varna, Studentska Str. 1
BULGARIA
ivailo.penev@tu-varna.bg, mkarova@ieee.bg, mgtodorova@tu-varna.bg http://cs.tu-varna.bg

Abstract: - The paper presents application of the k-nearest neighbor algorithm (kNN) for recognition of hand-written digits from 0 to 9. The emphasis is on the choice of the number of the nearest neighbors (the k parameter), which has significant impact on the algorithm performance. The main steps of the algorithm are described. The function for distance calculation and the method for choosing a class of the recognized digit are explained. Experimental results are presented. According to the results recommendations for the choice of k are summarized. The aim is increasing the performance of the kNN algorithm for the hand-written digit recognition problem, regarding two criteria – percent of the correctly recognized input data and time for recognition.

Key-Words: - machine learning, nearest neighbors, kNN, hand-written digits, recognition, classification

1 Introduction

The hand-written digit recognition problem is of significant importance for practice. An example for its application are the mail services of some countries, where the packets are scheduled automatically. Furthermore the problem is a good example for image recognition and is a proper base for research of various algorithms.

The problem is a classification problem. The input data (an image of a hand-written digit) are classified to one of a set of groups (called classes), defined in advance. The classes for this problem are digits from 0 to 9. Classification problems are solved by machine learning algorithms. These algorithms are trained by proper data, for which the pairs input – output data are known. So called classifier is built, which is then tested with another test data set. Finally the trained algorithm is able to recognize with a high level of probability new data, unknown to the algorithm.

Different machine learning algorithms and methods for solving classification problems are known – for example neural networks, supported vector machines [3]. Each one has advantages and disadvantages. As a result of significant research groups of classification problems and their instances are described, for which some machine learning algorithms are recommended.

One of the most often machine learning algorithms used is the k-nearest neighbor algorithm, notated as kNN. The algorithm is especially suitable for solving the hand-written digit problem. In general the algorithm performs the following main steps:

- Calculation of distances between data sets;
- Finding the nearest neighbors on the basis of the calculated distances;
- Choosing a class for the new data set corresponding to the class of the nearest neighbors.

The main advantages of kNN in comparison to other machine learning algorithms are easy implementation, no necessity of explicit training, easy interpretation of the achieved results [4, 5]. The algorithm needs to be supplied with a data sets of known classes.

The key problem in kNN implementation is the choice of the number of the nearest neighbors (this is the k parameter). The k parameter has significant influence on the performance of the algorithm. If the k value is high, the classifier is precise and more new data sets are correctly classified, but the recognition takes long time. In case of low k value the algorithm completes fast, but reports great error of recognition.

The impact of the k parameter on the kNN performance for various problems is studied (e.g. [1, 2]). The common conclusions are, that the choice of k depends on the specific problem and its optimal value is determined experimentally.

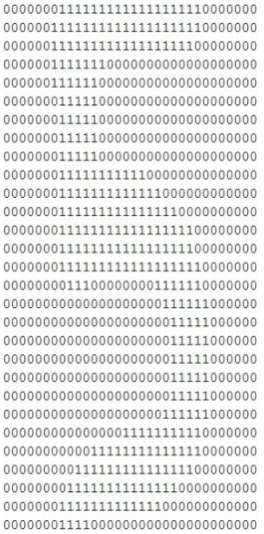
2 Algorithm kNN for recognition of hand-written digits

For solving the hand-written digit problem the kNN algorithm performs the next steps:

- Converting the image of a digit into a format, suitable for processing – representing the digit image as a set of characters;
- Calculation of distances between the sets of characters of the digit for recognition and the digits known to the algorithm in advance;
- Finding the nearest neighbors on the basis of the calculated distances;
- Choice of a class, to which the digit for recognition belongs to, according to the class of the nearest neighbors.

2.1 Converting a digit image

Typically the images of the digits are entered in some graphic format (e.g. BMP, JPG, PNG). The algorithm processes the digits as sets of characters 0 and 1. The character “1” presents availability of a pixel into the grey scale of the image, and the character “0” presents the lack of a pixel (fig. 1). All digit images, both the one for recognition and the known ones, are converted into such set. The image of each digit is presented into an array of size 32x32 (fig. 1).



```
00000001111111111111111110000000
00000011111111111111111110000000
00000011111111111111111100000000
00000011111110000000000000000000
00000011111100000000000000000000
00000001111100000000000000000000
00000001111100000000000000000000
00000001111100000000000000000000
00000001111100000000000000000000
00000001111111111100000000000000
00000001111111111111000000000000
00000001111111111111110000000000
00000001111111111111111100000000
00000001111111111111111100000000
00000001111111111111111110000000
00000000111000000001111110000000
00000000000000000000111111000000
00000000000000000000011111000000
00000000000000000000011111000000
00000000000000000000011111000000
00000000000000000000011111000000
00000000000000000000011111000000
00000000000000000000111111000000
00000000000000011111111110000000
00000000000111111111111110000000
00000000011111111111111100000000
00000000111111111111110000000000
00000001111111111111000000000000
00000001110000000000000000000000
```

Fig. 1. Set of characters after converting the digit image

Afterwards the array is transformed into a vector. The result is a set of characters for the digit, suitable for algorithmic processing:

$digit = x_1\, x_2 \ldots x_{904}$, where $x_i = \{0,1\}$.

The modern programming frameworks support effective tools to perform the converting operations.

2.2 Calculation of distances

On the next step the algorithm calculates distance between the set of a digit for recognition and the sets of other digits, known to the algorithm in advance.

If $digit_j$ and $digit_k$ are the sets of two digits, i.e.:

$digit_j = x_1\, x_2 \ldots x_{904}$
$digit_k = y_1\, y_2 \ldots y_{904}$,

the function D for calculation the distance between $digit_j$ and $digit_k$ has to satisfy the following conditions:

- $D(digit_j,digit_k) \geq 0$, $D(digit_j,digit_k) = 0$ only if $digit_j = digit_k$, i.e. the two sets are equal;
- $D(digit_j,digit_k) = D(digit_k,digit_j)$.

For calculation of the D function the Euclidian distance D_E is used:

$$D_E(digit_j, digit_k) = \sqrt{\sum_{i=1}^{904} \left(x_i - y_i\right)^2} \tag{1}$$

This function is calculated for the sets of the digit for recognition and all the known digits.

2.3 Finding the nearest neighbors and choosing a class

On fig. 2 example Euclidian distances between a digit for recognition and other known digits are shown.

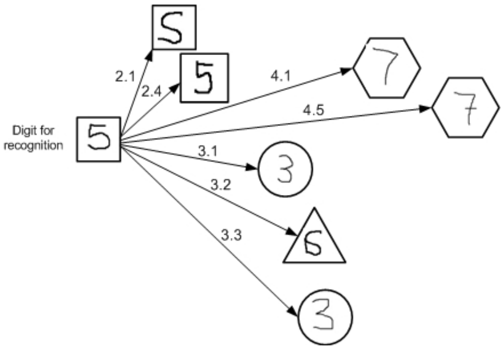


Fig. 2. Example distances

The algorithm finds the nearest neighbors according to the calculated distances depending on the value of k (number of nearest neighbors). Rating of the nearest neighbors regarding the distances to the data set for recognition is built. Fig. 3 shows the nearest neighbors for k=4.

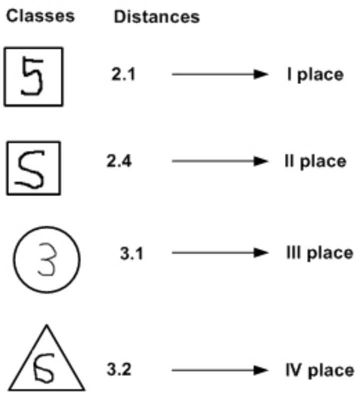


Fig. 3. Nearest neighbors in the case of k=4

After the nearest neighbors are found a class for the digit is determined. The class “winner” is the one with most participants in the list of the nearest neighbors. This is the class to which the digit for recognition belongs to. For the rating from fig. 3 the final choice of a class is presented on fig. 4.

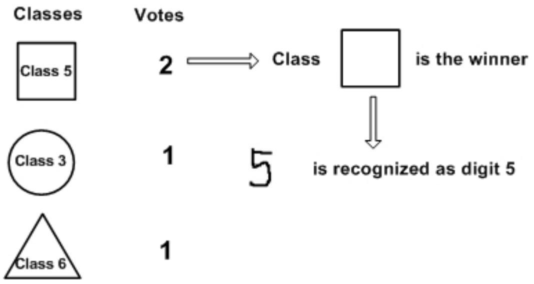


Fig. 4. Choosing a class to which the digit for recognition belongs to

A special case with two or more classes with equal number of participants in the list of nearest neighbors is possible. In this case additional weighted vote is performed. A coefficient is calculated:

$$\frac{1}{D^2(digit\_new, digit\_known)} \qquad (2)$$

, where *digit_new* – digit for recognition, *digit_known* – digit from the candidate class, for which the vote is calculated and *D* – the calculated distance between *digit_new* and *digit_known*.

The vote for each candidate class is calculated as follows:

$$vote = \sum_{i=1}^{n} \frac{1}{D^2(digit\_new, digit\_known)} \qquad (3)$$

, where *n* – number of known digits from the candidate class, for which the vote is calculated.

Equation 3 shows, that less distance between the digit for recognition and the known digits has greater impact on the calculated vote. The winner is the class with most votes.

3 Experimental Results

3.1 Experimental environment

The algorithm is implemented as an application in C#. The machine learning framework Accord.NET is used. The framework provides tools for audio, image processing, statistics as well as other features, implemented in C# [6].

The application is installed and tested in the following computer configuration:

- Processor Intel Xeon E5450 3.0 GHz;
- 4 GB RAM;
- Windows 7 Ultimate Service Pack 1 64-bit (x64);
- .NET Framework 4.5.

3.2 Experiments

Before the tests the algorithm is provided by 1934 images with known classes. The tests are carried out with 946 examples of images of digits from 0 to 9. The algorithm is run with a set of 946 digits for recognition. The tests are performed with different values of the k parameter. Table 1 presents the results from the tests.

Table 1. Results from the tests

k	**Number of correct recognized digits**	**Number of incorrect recognized digits**	**% of the correct recognized digits**	**Time for kNN execution (ms)**
1	935	12	98,73	7,591
2	933	13	98,63	7,532
3	934	12	98,73	7,582
4	936	10	98,94	7,483
5	929	17	98,2	7,445
6	929	17	98,2	7,47
7	926	20	97,89	7,571
8	927	19	97,99	7,601
9	926	20	97,89	7,574
10	927	19	97,99	7,504
15	922	24	97,46	7,51
20	921	25	97,36	7,524
25	916	30	96,83	7,544
30	912	34	96,41	7,577
35	908	38	95,98	7,504
40	903	43	95,45	7,535
45	901	45	95,24	7,504
Average time for kNN execution				7.532

Fig. 5 presents the part of the correct recognized digits from all the tests for various values of k.

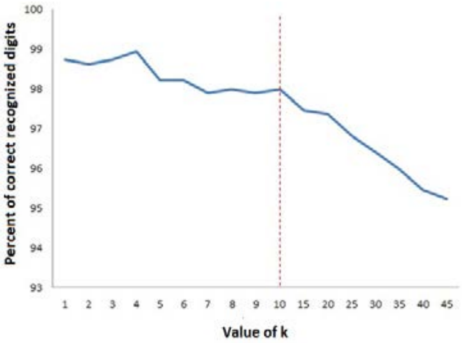


Fig. 5. Part of the correct recognized digits

The impact of the k value on the time for algorithm completion is also important (fig. 6).

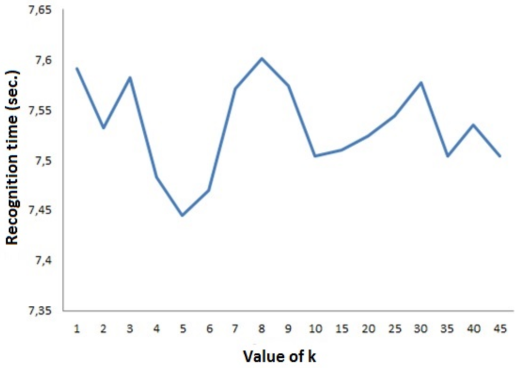


Fig. 6. Recognition time for different k

The experimental results show, that the part of the correct recognized digits is best for k = 4 (98.94%). The recognition time is least for k = 5 (7.445 sec.).

This result is explained with the specifics of the kNN algorithm. In the case of k=4 (four nearest neighbors) there is high probability for additional vote between two classes with equal number of nearest neighbors, which increases the recognition time. In the case of k=5 (five nearest neighbors) the probability for additional vote is less. Consequently the recommended value of k in the hand-written digit recognition by kNN is $k \approx \sqrt{N}$, where N is the number of classes. To achieve better recognition time the k value should be odd (table 2).

Table 2. Recommendations for the k value in hand-written digit recognition by kNN algorithm

Target criteria	**Value of k**	**Recommended value of k**
High part of the correct recognized digits	$k \approx \sqrt{N}$, where N – number of classes (in the given problem N = 10)	k =4
Less recognition time		k = 5 – odd value

4 Conclusion

The recommendations for the choice of k, summarized in the previous part, combined with the positive features of the kNN algorithm, could make its implementation for the hand-written digit recognition more effective. The good recognition time makes the algorithm proper to work in real-time systems, where large data sets should be processed quickly (for example in robots and controllers).

The future work will be concentrated on the following directions:

- Comparison of kNN with other algorithms for hand-written digit recognition;
- Experiments with the kNN in other machine learning problems to confirm or reject the recommendation for k, given in this paper;
- Implementation of kNN for hand-written digit recognition in robots.

References:

[1] A. Ghosh, On optimum choice of k in nearest neighbor classification, *Computational Statistics & Data Analysis*, 2006.

[2] K. Weinberger, J. Blitzer, L. Saul, Distance metric learning for large margin NN classification, *Advances in neural information processing systems*, 2005, pp.1473-1480.

[3] P. Harrington, *Machine learning in action*, Manning Publications, ISBN: 9781617290183, 2012.

[4] Y. Lecun, Comparison of learning algorithms for handwritten digit recognition, *International conference on artificial neural networks*, 1995, pp. 53-60.

[5] Y. Song, IKNN: Informative k-nearest neighbor pattern classification, *European Conference on Principles of Data Mining and Knowledge Discovery*, Springer Berlin Heidelberg, 2007, pp. 248-264.

[6] http://accord-framework.net