

Application of KNN and SVM algorithms in handwritten digit recognition

Yang Liu

Abstract: Handwritten digit recognition has important practical application value in the fields of automatic reading of data, bank check recognition, and automatic identification of postal codes. In this paper, KNN and SVM are used to identify handwritten numbers on the MNIST Data Set. Try different settings for optimal performance and compare the performance of the two classes of classifiers by considering the accuracy of the number recognition and the training and test times. Experimental results show that polynomial kernel SVM with degree 2 can achieve 98.05% accuracy in shorter training time. Whether it is accuracy or recognition time, SVM is superior to KNN.

Keywords: machine learning; handwritten digit recognition; SVM; MNIST Data Set; K nearest neighbor algorithm;

1. Introduction

With the development of the economy, the financial market is also accelerating, but there are still a large number of handwritten digital reports, remittance slips, etc. that need to be recognized by the computer, if the computer can be used for handwritten digital recognition, it will save a lot of manpower, financial and material resources. Therefore, the development of handwritten digital recognition system plays an important role in production and life[1].

There are many machine learning algorithms, and while different algorithms can solve the same problem, the end result may be different. Because each algorithm has its limitations and advantages, it makes no practical sense to talk about the quality of an algorithm. KNN and SVM are currently theoretically mature algorithms, which are intuitive and simple compared with other algorithms, and have been widely used in various classification problems [2], so this paper uses these two algorithms to identify handwritten digits and compare the performance of the two on the MNIST Data Set.

This study used the MNIST Data Set. It is from the National Institute of Standards and Technology, is a large database of handwritten digits commonly used to train image processing models, which can be downloaded from the website [3] and is a very classic Data Set in the field of machine learning. It consists of a training set and a test set, which contains a total of 60,000 gray-scale images and labels, while the test set contains a total of 10,000 gray-scale images and labels [4]. Each image is a 28×28-pixel gray-scale handwritten digital picture with a standardized size number in its center, and the label of each image is a number from 0 to 9, equal to the number in its center. Since 1998, this Data Set has been widely used in the field of machine learning and deep learning to test the effects of algorithms such as linear classifiers, K-neighbor algorithms, support vector machines, neural networks, etc. This article uses 28×28=784 gray-scale pixel values as characteristics of each sample. The graph of the first 50 handwritten digit samples in the MNIST training set is shown in Figure 1.

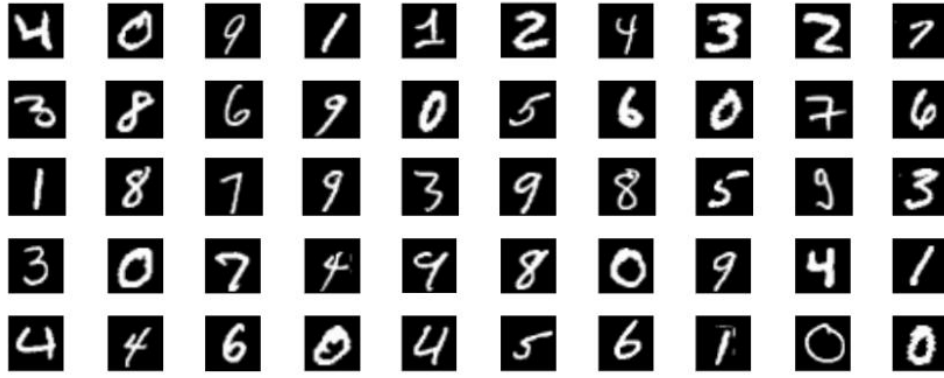


Figure 1. Sample images of the MNIST data set

2. Research status

Domestic research on handwriting numbers started relatively late and focused on the study of specific scenarios. The automation research of the Chinese Academy of Sciences uses a limited state automaton to identify handwritten digits, reaching a 95.2% accuracy rate; Tsinghua University uses a support vector machine to achieve a 92% accuracy rate on 20,000 test samples for handwritten digits in financial bills; Shanghai JiaoTong University uses an artificial neural network as a classifier to test 6,000 samples, achieving a 97.58% accuracy rate.

Foreign research started relatively early, mainly focusing on theoretical research. In 1962, Hubel and Wiesel studied cat visual structure, proposing a model of visual structure and the concept of a sensory field. In 1969, Marvin Minsky and Seymour Papert studied that neural networks with only one implicit layer have arbitrary expressive capabilities, that is, they are able to separate arbitrary data. In 1974, Paul Werbos proposed a backpropagation algorithm that made the training of neural networks possible, and neural network research entered a boom. In the 1990s, due to the very large computational cost of neural networks, the research of artificial neural networks fell into a trough, with the continuous development of computer computing power and the accumulation of data volume, the most prominent work in image classification in recent years was done using artificial neural networks.

In this paper, the gray-scale pixel value of the handwritten digital image is featured, and the KNN and SVM methods are used on the MINST Data Set to try different settings by considering the accuracy of the number recognition and the training and testing time to obtain the best performance and compare the performance of the two types of classifiers.

3. Classifier settings and Experimental Results

3.1 KNN

K-neighbor algorithm belongs to the classification and logistic regression algorithm, which is a simple and effective algorithm in machine learning. K-neighbor learning algorithm belongs to the classification algorithm in machine learning[5], and the working mechanism is relatively simple: given a sample set, also known as a sample training set, and know the correspondence between the data in the sample Data set and the classification to which it belongs. After entering a new data without classification, the computer extracts the characteristics of the data and compares them with the features in the Data set to find K data that are similar to its characteristics, and the classification with the highest classification frequency in the K data is the classification of the new

data. The choice of distance measurement method and K value is critical to performance. In this experiment, we first calculate its K nearest neighbors by Euclidean distance, and then assign this new sample to the most common class of K nearest neighbors. In determining the new sample class, we can also weight the neighborhood.

3.1.1 The choice of K-values

If you choose a smaller K value, the training instance set will become smaller, the approximate error of the training will become smaller, and the training data will be similar to the new data entered to have an effect. When the K value becomes smaller, it is susceptible to the influence of outliers and is more likely to have over-fitting problems. Choose a larger K value, the training instance will become larger, the training approximation error will also become very large, if the training data is not related to the new input of the new data will also produce different effects, easy to appear sample balance problem, so that the model becomes simple, there is a problem of underfitting. Therefore, in order to better evaluate the performance, set the k value to K (K=1,3,5,7,9) for experimentation.

3.1.2 Evaluation of KNN

When different K values are set, the experimental results are shown in the figure, and the accuracy rate reaches a maximum of 97.07% when K=3. KNN has no training time, but the testing process is very time-consuming, with 10,000 test samples taking about 85 minutes per k.

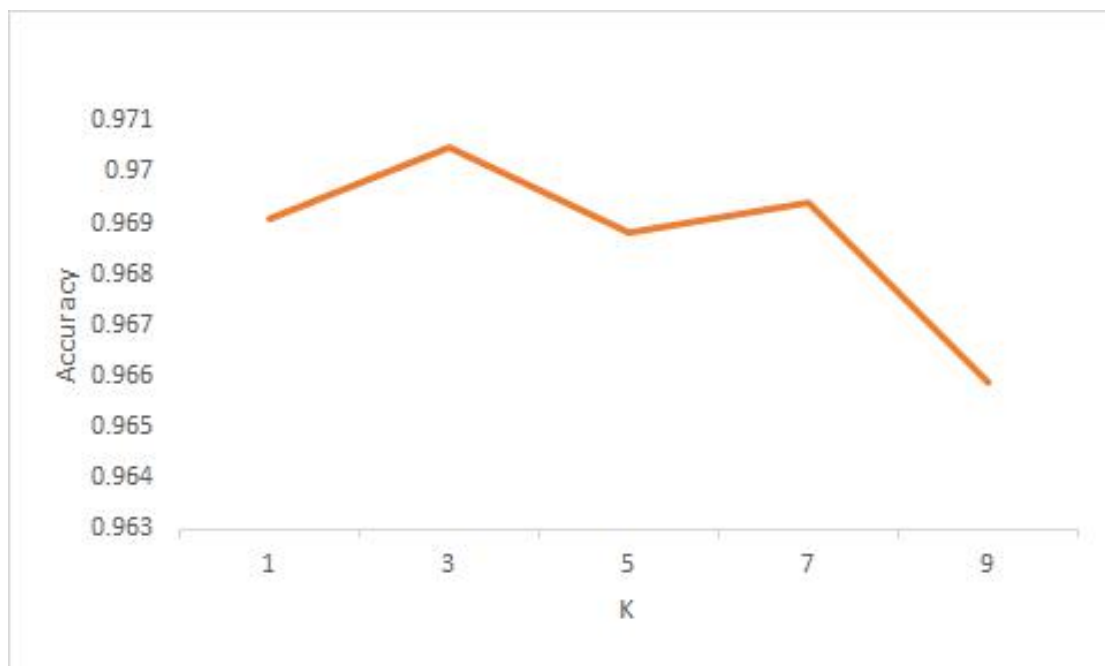


Figure 2. Accuracy with different k

3.2 SVM

3.2.1 Introduction to SVM

The support vector machine (SVM) is a linear classifier defined in the feature space so that the class break is the largest [6], the goal of which is to find the optimal classifier that can achieve the above requirements. In the case of linear separability, if a given set of samples has m samples,

i.e.: $T = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$, $y_i \in \{-1, +1\}$ where x_i represents the i sample in the feature space and y_i represents its classification result. The core of the support vector machine is to find an optimal classification surface $w^*x + b = 0$ in the feature space of the sample set T . The class polygon should be such that the class break is the largest, that is, the sample closest to the plane (i.e., the support vector) is the largest distance to the plane. The process of solving the optimal classification surface is essentially to solve a quadratic convex programming problem, that is, to find w and b that meet the conditions of $y_i(w^*x + b) \geq 1$, and also to maximize the class breaks. In the case of linear indivisibility, a relaxation term ξ is introduced, that is, the problem that satisfies the condition of $y_i(w^*x + b) + \xi \geq 1$ can continue to use the optimal classification plane for approximate linear division, so that the generalized optimal classification surface can be obtained; in the case of non-linearity, it can be converted to a linear problem in a high-dimensional space, and then the optimal classification surface can be obtained. However, this kind of transformation and solution is usually more complex and not easy to implement, so a very important concept has been introduced into the SVM - the kernel function. Kernel functions map sample spaces to high-dimensional spaces without increasing the complexity of the calculation. With the addition of kernel functions, the support vector machine eventually tends to be perfected.

In this experiment, assuming that nonlinear decision boundaries and misclassified samples are allowed, the SVM can be optimized as follows:

$$\max \quad \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j [\phi(x_i) \cdot \phi(x_j)] \quad (1)$$

$$\text{s.t.} \quad 0 \leq \alpha_i \leq C, i = 1, \dots, n \quad (2)$$

$$\sum_{i=1}^n \alpha_i y_i = 0 \quad (3)$$

where $\phi(x)$ is the eigenvector of input x and C is the relaxation penalty. By solving the optimization problem described above, we can get the optimal hyper-plane and use it to classify new samples.

3.2.2 Kernel functions

The kernel function of the SVM has an important impact on the accuracy of the identification. The more common kernel functions are linear kernel functions, Sigmoid kernel functions, Gaussian kernel functions, and polynomial kernel functions[7]. Different cores may have different performance. $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ is a kernel function. In the experiment, the following three kernel functions were used to compare their performance on handwritten number recognition problems, mainly linear kernel functions, polynomial kernel functions, and radial base kernel functions. The first one is the Linear Kernel, second one is the Polynomial Kernel, third one is the Radial Basis Kernel.

$$\begin{aligned} K(x, x') &= x \cdot x' \\ K(x, x') &= (1 + x \cdot x')^p \\ K(x, x') &= \exp(-\gamma \|x - x'\|^2) \end{aligned}$$

3.2.3 Evaluation of SVM

Using Libsvm to train and test the SVM classifier. Table 1 shows experimental data for three kernel functions. For radial basis functions, the target value of the SVM is always oscillating when the $\gamma=1$ is set. Therefore, when changed to $\gamma=1/d$ (d is the number of features, set to $d=784$), it takes less training time and obtains good accuracy compared to $\gamma=1$.

Table1. Summary for the experimental results of SVM.

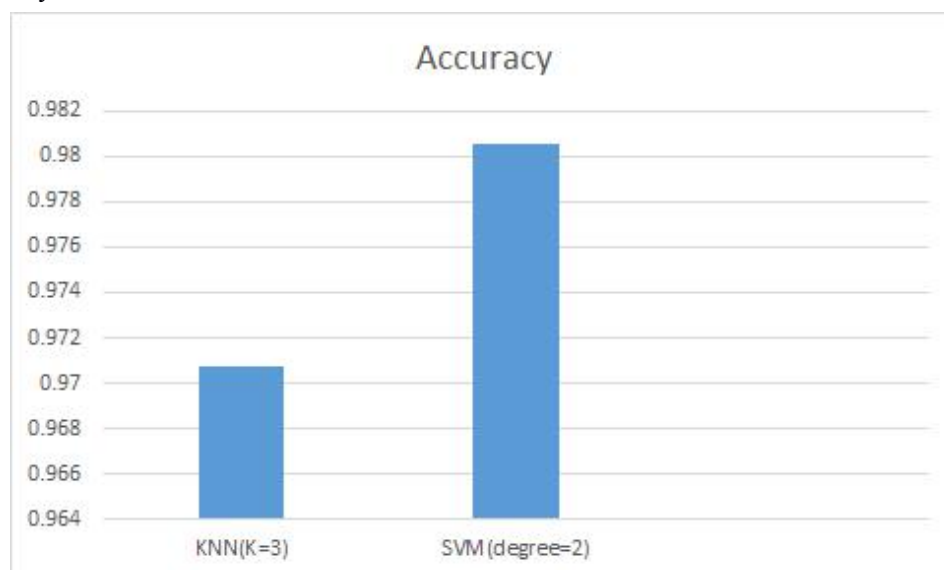
Type	Accuracy	Training Time(min)	Testing time(min)	Total time(min)
Polynomial Kernel(degree=2)	98.05%	3.9021	2.4035	6.3056
Linear Kernel	93.95%	6.6421	2.6315	9.2736
Radial Basis Kernel($\gamma=1/784$)	94.41%	11.0024	3.8893	14.8917

By analyzing the data in the table, it can be obtained that the training and testing time of the polynomial kernel with degree 2 is the shortest and the accuracy is the highest, reaching 98.05%. Linear kernel function training time and test time are reasonable, and the performance is good. The radial base kernel function is slightly more accurate than the linear kernel function, but takes the longest to train and test.

As can be seen from the data in the table, the polynomial kernel function has the shortest training and testing time among these three kernel functions, while the radial base kernel function has the longest time. In this experiment, the total number of support vectors for the linear kernel function and the radial base kernel function is 10253 and 18921, respectively, while the polynomial kernel with degree 2 is the smallest total support vector of the three kernel functions with 9122.

4. Comparison of different classifiers

As shown in the figure, in order to compare the performance of KNN and SVM, the optimal accuracy and total run time (including training time and test time) of the two classifiers are plotted, respectively.



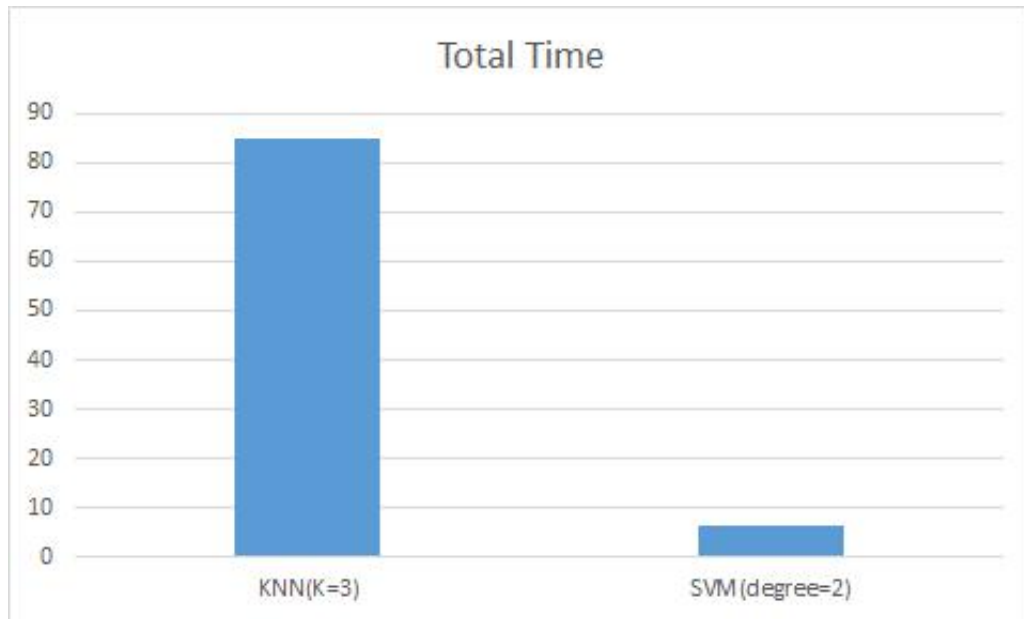


Figure 3. The performance of KNN and SVM

In both classifiers, the accuracy of polynomial kernel SVM with degree 2 is higher than that of KNN (K=3), which is 98.05%. The total run time of the SVM is also less than KNN (K=3), and it takes only 6.3056 minutes to complete training and testing.

5. Conclusions

This article briefly introduces the basic principles of KNN and SVM, and implements the recognition of handwritten digital images. There are several interesting directions to explore in the future. First of all, we can try multiple classification methods. Get better performance by combining the power of multiple learning algorithms. Secondly, we can use feature extraction methods to improve recognition accuracy while reducing calculation time.

This is the first time to learn programming languages and artificial intelligence, based on this experiment, more is the understanding of the course content and the learning of existing solutions, after which more exercises are needed, improve hands-on ability and innovation ability, and propose more efficient methods when solving problems.

Reference

- [1] Chi Dian Committee. A Handwritten Number Recognition Implementation Based on A Algorithm[J]. Information and Computers (Theoretical Edition), 2019, 31(17): 20 - 22.
- [2]GUO Mingzhao. Application of Decision Tree Algorithm and KNN in Handwritten Number Recognition[J]. China Strategic Emerging Industries, 2017(40): 130 - 131.
- [3]LeCunY,CortesC , BurgesCJ.MINST hand written digit database[DB/OL].AT&T Labs Available,2010[2021.8.12]:[http://yann.lecun.com/exdb.minst,2](http://yann.lecun.com/exdb/mnist,2).
- [4]Chen M.Convolutional Neural Network with Pruning Method for Handwritten Digit Recognition[C/OL].CornelUnivercity,2021[20210312]:<https://arxiv.org/abs/2101.05996v1>
- [5]Pi yachen. Application of K-Neighbor Classification Algorithm[J]. Communication

World,2019,26(1):286-287.

[6]LI Hang. Statistical Learning Methods (2nd Edition) [M]. Beijing Li:Tsinghua University Press,2019:111-153.

[7]Zhou Zhihua. Machine Learning[M].Beijing:Tsinghua University Press,2016:121-140.