Implementation and Error Analysis of MNIST Handwritten Dataset Classification Based on Bayesian Decision Classifier

Rui Ma¹², Long Han¹, Hujun Geng*¹
¹China Electronics Technology Corporation Network Communication Research Institute, ShiJiaZhuang, Hebei, P.R.China
²College of Electronic Science of NUDT, Changsha, Hunan, P.R.China
³College of Liberal Arts of NUDT, Changsha, Hunan, P.R.China
⁴China Electronics Technology Corporation Network Communication Research Institute, ShiJiaZhuang, Hebei, P.R.China
*email: genghujun@cetc.com.cn

Abstract. In recent years, with the continuous development of computer technology, pattern recognition technology has gradually entered people's life and learning, and people's demand for pattern recognition technology is also growing. In order to adapt to people's life and study, the application of pattern recognition theory is more and more, such as speech recognition, character recognition, face recognition and so on. The main methods of pattern recognition are statistics, clustering, neural network and artificial intelligence. Statistical method is one of the most classic methods, and Bayesian classification is widely used in statistical method because of its convenience and good classification effect.

1. Introduction
In our daily life, we are facing numbers, printed script, etc. all the time. But the recognition of numbers makes one of the tasks of our daily life. It is difficult to recognize the printed number, so it is not difficult to reverse the machine. For the recognition of handwritten numbers, because of the different habits of writing, holding pen and enlightenment, it is a difficult problem for us to distinguish the number written by others. It's even harder for machines. In our life, we can not avoid a lot of handwritten numbers identification, such as doctor's prescription list, traffic police punishment list, etc. With the development of modernization and information technology, how to realize the recognition of handwritten figures with computer or modern computing machine is an urgent problem.

In view of the fact that handwritten numerals are difficult to recognize and the recognition accuracy is low, this paper proposes a MNIST handwritten dataset classification method which uses KNN method to estimate the class probability density and designs a Bayesian classifier. Firstly, the KNN method is used to estimate the probability density, and then the dimension reduction data is used as the input of the trained Bayesian classifier to get the recognition of the input handwritten digits.

2. K-nearest neighbor algorithm

2.1. Introduction of k-nearest neighbor theory
K-nearest neighbor (k-NN), k-NN is actually a majority voting model. After inputting the instance feature vector x and the instance category y of the training data set, according to the given distance measure, k points nearest to X are found in the training set, and the neighborhood of X covering these
k points is denoted as NK (x); In NK (x), the category y of X is determined according to the decision rules as follows:
\[ y = \arg \max_{c_j} \sum_{x_{i \in N_k(x)}} I(y_i = c_j), i = 1,2,...N \] (1)

Where I is the indicator function, I is 1 when \( Y_i = C_j \), otherwise I is 0.

2.2. Experiment Introduction
This experiment first selects k = 6, 11, 16, 21 four values to test the accuracy, that is, select the k points closest to the current point to vote out the label of the current point, the output accuracy (the number of predicted correct divided by the total number) is relatively concentrated and the difference is not big, and then combined with the characteristics of the data set, select k as 17, 18, 19, 21 equivalent to test to get the approximate impact.

3. Bayesian classifier
First of all, Bayesian classification algorithm[2] is a probability Classification Method in statistics. Naive Bayesian classification principle is to use Bayesian formula to calculate the posterior probability according to the prior probability of a feature of the observation object, and then select the class with the maximum posterior probability as the class of the feature.

The prior distribution is combined with the sample information to derive the posterior distribution, that is, the conditional distribution of the observed object in the case of a given sample. In daily life, it is often used in Bayesian decision-making. First of all, let's take a look at the Bayesian formula, such as follows:
\[ P(A | B) = \frac{P(A)P(B | A)}{P(B)} \] (2)
\[ P(A | B) = \frac{P(A)}{\sum_i P(A)P(B | A_i)} \] (3)

P (A) is called prior probability, that is, before event B is tested, we judge the probability of event A.

P (A | B) is called posterior probability, that is, after event B, we re evaluate the probability of event A, P (B | A) / P (B) is called probability function, which is similar to an adjustment factor, which makes the probability of prediction closer to the real probability. So we understand conditional probability as:

Posterior probability = prior probability * adjustment factor

The size of the adjustment factor affects the possibility of event A, that is, when the probability function is more than 1, it means that the prior probability is enhanced and the probability of event a is larger; When probability function = 1, it means that event B is not helpful to judge the possibility of event A; When the probability function is less than 1, it means that the prior probability is weakened and the probability of event A is reduced. The expression of transforming Bayesian formula into classifier in this paper is as follows (4):
\[ P(\text{category} | \text{features}) = P(\text{category}) \frac{P(\text{features} | \text{category})}{P(\text{features})} \] (4)

This formula means that a thing is judged as a certain category based on whether there is a certain feature. If there is such a feature, then the probability of being judged as such becomes larger, otherwise it becomes smaller.

Then we introduce the Naive Bayes, whose formula is shown in (5), as follows:
\[ P(Y = c_k | X = x) = \frac{P(Y = c_k)\prod_j P(X^{(j)} = x^{(j)} | Y = c_k)}{\sum_k P(Y = c_k)\prod_j P(X^{(j)} = x^{(j)} | Y = c_k)} \] (5)

\[ k = 0,1,2...K \]
In the above formula, the feature is converted to the value of Y which can be calculated. When y is a different value, the probability of which type it belongs to is the largest, and it is judged as this type. Then, in order to obtain the maximum a posteriori probability, we transform it into the value of Y which makes the maximum a posteriori probability. Then, y can be expressed as the following equation (6):

$$y = f(x) = \arg \max_{c_j} P(Y = c_j) \prod_i P\left(X^{(i)} = x^{(i)} \mid Y = c_j\right)$$

For the denominator term in formula (6), it is the result of multiplication of Y with different values, so for each k value, its denominator is the same, which can be counted as a constant. The denominator is ignored when comparing sizes. Then the comparison formula can be simplified as the following formula (7):

$$y = \arg \max_{c_k} P(Y = c_k) \prod_j P\left(X^{(j)} = x^{(j)} \mid Y = c_k\right)$$

4. MNIST handwriting dataset

The following is a description of MNIST handwritten data set. In the data set, each picture is taken as a sample. The classification results are 10 types, respectively 0,1,2,... 9. Each pixel of each picture in the data set adopts gray value. In order to facilitate the following processing, it will be transformed into a binary image, that is, “set the point not 0 to 1”. After this processing, it can be considered whether a pixel becomes a 0-1 distribution for 1 [3]. That is, the probability value of formula (8) is calculated:

$$P(\text{This sample belongs to class } j) = \prod_{k=1}^{28^28} P(\text{This sample belongs to the kth pixel of class } j)$$

Then we transform the data set classification problem into a Bayesian classification problem. Let a denote $D = \{x^{(i)}, y^{(i)}\}_{i=1}^{n}$ dataset. X (I) is a 28 dimensional vector representing the ith sample, y (I) is the labeled category, and the value range is 0...9. The formula of the sample can be written as (9): X (I) is a 28 dimensional vector representing the ith sample, y (I) is the labeled category, and the value range is 0...9.

$$p\left(y^{(i)} = j \mid x^{(i)}\right) = \prod_k P\left(y^{(i)} = j \mid x_k^{(i)}\right)$$

Then, we transform this kind of pattern recognition problem into a simple classification problem, that is, solving the target can be rewritten as the following equation (10):

$$f = \arg \max_j p\left(y^{(i)} = j \mid x^{(i)}\right)$$

This kind of problem can be transformed into comparing the size of p (y^{(i)} =0|x^{(i)}) , p (y^{(i)} =1|x^{(i)})...p (y^{(i)} =9|x^{(i)}) , Find the largest one to determine the category of the picture, that is, which number is in the picture. Next, find p (y^{(i)} =j|x^{(i)}) , The corresponding posterior probability of transforming it into corresponding pixel is p (y^{(i)} =j|x^{(i)}) . Using Bayes formula (3), the problem is transformed into the following formula (11):

$$p\left(y^{(i)} = j \mid x_k^{(i)}\right) = \frac{p\left(x_k^{(i)} \mid y^{(i)} = j\right)p\left(y^{(i)} = j\right)}{p\left(x_k^{(i)}\right)}$$

If a picture belonging to category 0 has a high probability of 1 on pixel 20, then if pixel 20 is found to be 1, then it has a high probability of belonging to category 0. It can be seen from the data set itself that the object to be identified this time is a black-and-white image, so it’s easy to think of a 0-1
distribution, where the pixel is black, it's 1, and where the pixel is white, it's 0. As a beginner, this method is similar to the color block of two-dimensional code during the learning period. In this way, the data set is magnified into color blocks one by one, which is brought into the Bayesian model to solve, and the number recognition problem is transformed into a feature recognition problem.

5. Conclusion

5.1. Implementation of classifier
By running the code, put the code and data set into a file, and then carry out the process as shown in Figure 1. Firstly, the data set is read, then the model is trained, and finally the test data is read, and the accuracy is output.

5.2. Discussion on the influence of K value on the classification ability of classifier
In the first experiment, K values of 6, 11, 16 and 21 are selected for the experiment, and the results are shown in Table 1 below. The coordinate diagram is drawn with K value as abscissa and accuracy rate as ordinate, as shown in Figure 2 below:

| Tab. 1 The influence of K value on accuracy in the first experiment |
|-----------------------------|----------------|----------------|----------------|----------------|
| K | 6 | 11 | 16 | 21 |
|---|---|---|---|---|
| Accuracy | 0.8243 | 0.8392 | 0.8401 | 0.84 |

Fig. 1 Operation flow chart
Fig. 2 Data of the first experiment

In the process of analyzing the influence of data, there are too few redundant data, and the accuracy rate in the image has an obvious increasing trend with the increase of K value, but there is no obvious downward trend with the increase of K value; Then we can not determine the best value of K, and we can not analyze the subsequent impact. So I carried out the second experiment. First, I took K as 18, 26 and 36. When \( k = 18 \), the correct rate is the highest at present. We can see that the best K value is around 18. So I further took 17, 19, 20 and 23. The two experiments obtained the data shown in Table 2.

Tab. 2 Final experimental data

| K  | 6   | 11  | 16  | 17  | 18  |
|----|-----|-----|-----|-----|-----|
| Accuracy | 0.8243 | 0.8392 | 0.8401 | 0.8392 | 0.8401 |
| 19  | 20  | 21  | 23  | 26  | 36  |
| 0.8413 | 0.8393 | 0.84 | 0.8387 | 0.8395 | 0.8356 |

Fig. 3 K-accuracy chart

As can be seen from the above figure, the values of 0.8392 and 0.8413 are obtained by taking points 17 and 19 near 18, and 0.8413 > 0.8401 indicates that the classifier plays a better role at \( k = 19 \),
and the classification accuracy has been improved. In addition, when \( k = 20 \), the accuracy rate drops to 0.8393, but when \( k = 21 \), the accuracy rate rises, but it is stable at about 0.84 at 20-26. It can be seen that the classifier can play a good role in recognition when \( k \) is 17-26.

In addition, I think that the influence of \( K \) value on the final accuracy rate should go back to the meaning of \( K \) itself. That is to say, the meaning of \( K \) value mentioned above is to select the \( k \) points closest to the current point to vote and select the label of the current point. In theory, the more nearby points are selected, the more sufficient the estimation of the current point will be.

If we take too many points, for example, if we exaggerate to get all the points, then the estimation of all the points is the same, which greatly reduces the accuracy of the classifier; If you take too few points, such as zero points, then the probability of all points will not be different, so the \( K \) value is too large or too small will affect the accuracy. Considering the size of the data itself, selecting the appropriate \( K \) value is also a factor for the performance of the classifier.

Finally, in the problem environment of this paper, \( k = 19 \) is selected, and the trained classifier can achieve the accuracy of 0.8413 on MNIST handwriting data set.

**References**

[1] Qiao J F, Wang G M, Li W J and Chen M 2018 An adaptive deep Q-learning strategy for handwritten digit recognition J. Neural networks 12 pp 107-109

[2] Arnaldo P C 2018 Handwritten digit recognition J. Practical artificial intelligence 2 pp 461-478

[3] Jiang L, Zhang L, Li C, et al. 2019 A correlation - based feature weighting filter for naive Bayes J. IEEE transactions on knowledge and data engineering 31(2) pp 201-213

[4] Liu P, Zhao H H, Teng J Y, et al. 2019 Parallel naive Bayes algorithm for large - scale Chinese text classification based on spark J. Journal of Central South University 26(1) pp 1-12

[5] Mori M, Uchida S and Sakano H 2014 Global feature for online character recognition J. Pattern Recognition Letters 35

[6] Burges C J C 1998 A Tutorial on Support Vector Machines for Pattern Recognition J. Data Mining and Knowledge Discovery 2(2) pp 121-167

[7] Huang B, Zhang H, Jiang L G, et al. 2008 Using Instance Cloning To Improve Naive Bayes For Ranking J. International journal of pattern recognition & artificial intelligence 22(6)

[8] Kuen J, Liu T, Gu J X, et al. 2018 Recent advances in convolutional neural networks J.Pattern Recognition: The Journal of the Pattern Recognition Society pp 77354-377

[9] Juergen S 2015 Deep learning in neural networks: An overview J.Neural Networks: The Official Journal of the International Neural Network Society pp 6185-117

[10] Suen C Y and Niu X X 2012 A novel hybrid CNN-SVM classifier for recognizing handwritten digits J. Pattern Recognition: The Journal of the Pattern Recognition Society 45(4) pp 1318-1325