

Analysis of KNN algorithm

Zhibo Xu

February 2021

1 The principle of KNN algorithm

K-nearest neighbors (KNN) is a very basic machine learning method. Its basic idea is: in the training data and the condition of known labels, enter test data, the characteristics of the test data and training focused on to compare the characteristics of the corresponding, and find the most similar of training focus and former K data, then the test data, the corresponding category is K a classification of the data in the most times.

Because the KNN method mainly depends on the surrounding limited adjacent samples, rather than the method of discriminating the class domain to determine the category, therefore, for the sample set to be divided which has a lot of crossover or overlap of the class domain, the KNN method is more suitable than other methods. The KNN algorithm can be used not only for classification, but also for regression. The attribute of a sample can be obtained by finding the k nearest neighbors of the sample and assigning the average value of the attribute of these neighbors to the sample. A more useful method is to assign different weights to the influence of neighbors at different distances on the sample, such as the weight is inversely proportional to the distance.

2 Advantages of the algorithm

- (1) Simple, easy to understand, easy to implement, no need to estimate parameters.
- (2) Training time is zero. It does not show training, unlike other supervised algorithms that use the training set train as a model (that is, fit into a function), and then the validation set or test set is classified by that model. KNN just saves the samples and processes them after receiving the test data, so the training time of KNN is zero.
- (3) KNN can deal with classification problems, and nature can deal with multiple classification problems, so it is suitable for the classification of rare events.
- (4) It is particularly suitable for multi-modal problems (multi-modal objects have multiple category labels), and KNN performs better than SVM.
- (5) KNN can also deal with regression, that is, prediction.
- (6) Compared with algorithms such as Naive Bayes, it has no assumptions on data, high accuracy and insensitivity to outliers.

3 Disadvantages of the algorithm

- (1) Too much computation, especially when the number of features is very large. Each text to be classified needs to calculate its distance to all known samples in order to get its KTH nearest neighbor.
- (2) Poor intelligibility and inability to give rules like decision trees.
- (3) It is a lazy learning method, which basically does not learn, resulting in a slower prediction speed

than algorithms such as logistic regression.

(4) When the samples are unbalanced, the prediction accuracy of rare categories is low. When the sample size is unbalanced, for example, the sample size of one class is large while the sample size of other classes is small, it may lead to that when a new sample is input, the samples of the large-size class in the K neighbors of the sample are in the majority.

(5) Extremely high dependence on training data and poor fault tolerance of training data. If there are one or two errors in the training data set, which are just right next to the values to be classified, it will directly lead to the inaccuracy of the predicted data.

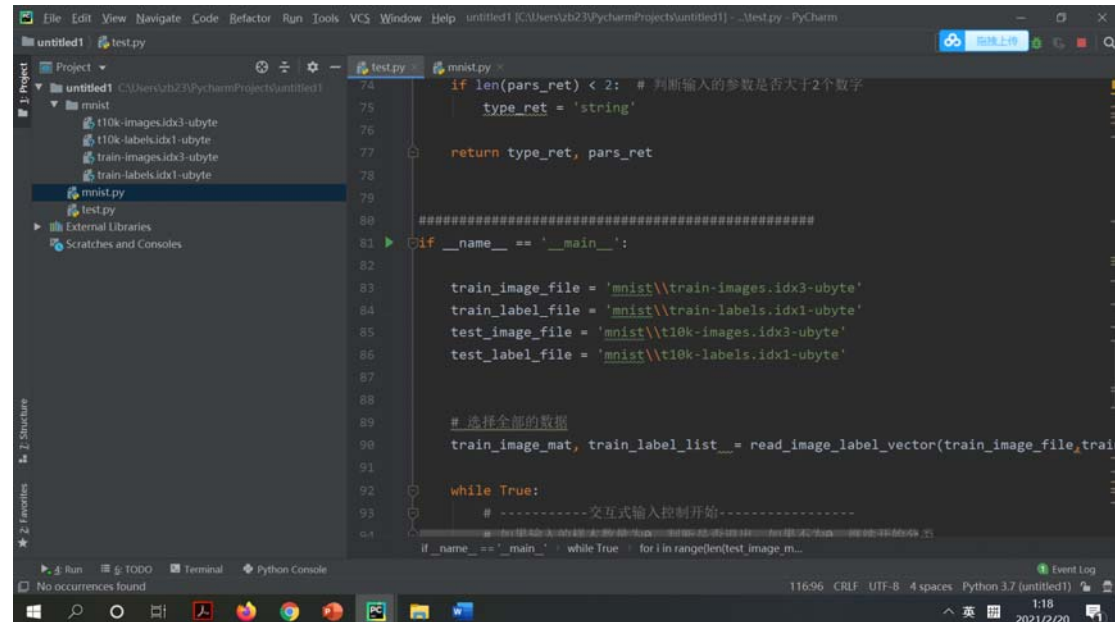
4 Distance metrics

In this experiment, KNN algorithm is applied to MNIST data set and CIFAR10 data set, and different distance measurements and different K values are adopted. Distance measurement is mainly used to measure the similarity between test set and training set. The distance measurement methods selected in this experiment are as follows:

1. Euclidean distance
2. Manhattan distance
3. Chebyshev distance
4. Cosine distance

And a common way to measure the difference between two probability distributions is relative entropy.

Mnist



```
File Edit View Navigate Code Refactor Run Tools VCS Window Help untitled1 [C:\Users\zb23\PycharmProjects\untitled1] - test.py - PyCharm
Project
  untitled1
    mnist
      110k-images.idx3-ubyte
      110k-labels.idx1-ubyte
      train-images.idx3-ubyte
      train-labels.idx1-ubyte
      mnist.py
      test.py
    External Libraries
    Scratches and Consoles
  Structure
  Favorites
  Run
  TODO
  Terminal
  Python Console
  No occurrences found
  116/96 CRLF UTF-8 4 spaces Python 3.7 (untitled1)
  1:18
  2021/2/20

test.py
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94

mnist.py
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94

if len(pars_ret) < 2: # 判断输入的参数是否大于2个数字
    type_ret = 'string'

return type_ret, pars_ret

#####

if __name__ == '__main__':

    train_image_file = 'mnist\\train-images.idx3-ubyte'
    train_label_file = 'mnist\\train-labels.idx1-ubyte'
    test_image_file = 'mnist\\t10k-images.idx3-ubyte'
    test_label_file = 'mnist\\t10k-labels.idx1-ubyte'

    # 选择全部的数据
    train_image_mat, train_label_list = read_image_label_vector(train_image_file, train_label_file)

    while True:
        # ----- 交互式输入控制开始 -----
        if __name__ == '__main__': while True: for i in range(len(test_image_m...)
```

Comparison of different distance metrics (k=5)

Euclidean distance

Manhattan distance

K=3

Euclidean distance

```
输入验证集数据个数50
准确率 0.240000
```

Manhattan distance

```
输入验证集数据个数50
准确率 0.160000
```

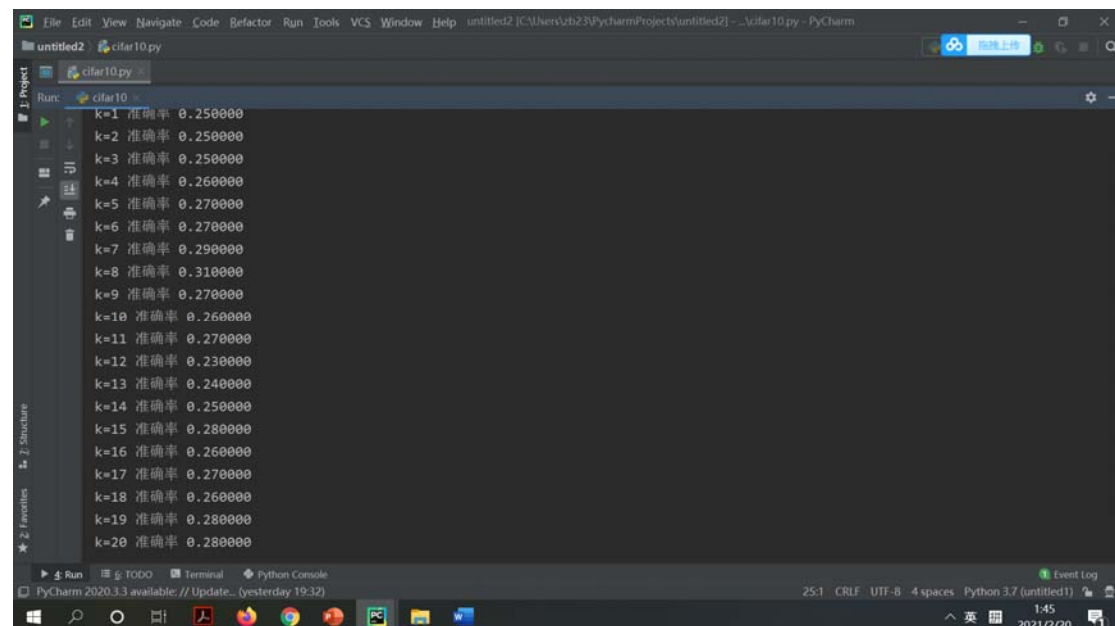
Chebyshev distance

```
输入验证集数据个数50
准确率 0.040000
```

Cosine distance

```
输入验证集数据个数50
准确率 0.140000
```

By comprehensive comparison, when KNN algorithm is used to identify MNIST data set and CIFAR10 data set, among the above four distance measures, Euclidean distance is the best.



For the CIFAR10 data set, DATA_BATCH_5 was selected as the training set, and 100 data of the verification set were selected. When the highest accuracy was obtained, the corresponding K value was 8.

5 The choice of K values

If you choose the smaller values of K, equivalent to a smaller training instances in the field of forecast, the approximation error will decrease, only with the input instance is close or similar training instances will only work on forecast results, at the same time the problem is to "learn" the estimation error will increase, in other words, the decrease of the K value means the whole model is complicated, prone to fitting;

If a larger value of K is selected, it is equivalent to using training examples in a larger field to make predictions, which has the advantage of reducing the estimation error of learning, but the disadvantage is that the approximate error of learning will increase. At this time, the training

instance which is far (not similar) to the input instance will also act on the predictor, making the prediction wrong, and the increase of K value means that the overall model becomes simple.

$K=N$, then it is completely inadequate, because at this time, no matter what the input instance is, it is simply predicted to be the most tiring in the training instance, and the model is too simple, ignoring a lot of useful information in the training instance.

In practical applications, K value is generally taken as a relatively small value. For example, cross-validation method is adopted (in short, part of the sample is used as the training set and part of the test set) to select the optimal K value.

References

[1] <https://blog.csdn.net/ymilton/article/details/89341652>

[2] <https://blog.csdn.net/eleclike/article/details/79994846>