Handwritten Digits Recognition Using KNN

Yang, J.

Abstract—These instructions give you guidelines for preparing papers for IEEE Computer Society Transactions. Use this document as a template if you are using Microsoft Word 6.0 or later. Otherwise, use this document as an instruction set. Please note that use of IEEE Computer Society templates is meant to assist authors in correctly formatting manuscripts for final submission and does not guarantee how the final paper will be formatted by IEEE Computer Society staff. This template may be used for initial submissions; however, please consult the author submission guidelines for formatting instructions as most journals prefer single column format for peer review. An abstract should be 100 to 200 words for regular papers, no more than 50 words for short papers and comments, and should clearly state the nature and significance of the paper. Abstracts *must not* include mathematical expressions or bibliographic references. Please note that abstracts are formatted as left justified in our editing template (as shown here).

1 INTRODUCTION

THE handwritten digits recognition is the ability which allows computer or some other devices to recognize digits. To recognize digits, computer need to read input from scanned documents, then use classification algorithm to classify the digits shown on these documents. In this report, I will use KNN to do classification job.

2 PROBLEM DECOMPOSITION

2.1 Dataset

In this classification job, I will use MNIST handwritten digit dataset, which is commonly used for traning handwritten digits recongnition, and the images form this dataset were normalized to the size of 28*28 pixel.

2.2 Construct training, testing and validation split

75% of the data are used for training, and the rest of 25% data is used for testing, furthermore, take 10% of the training data as validation set to find the value of k which can lead to the largest accuracy.

2.3 KNN

K-nearest neighbors(KNN) is a surpervised algorithm which is used for solving classification tasks. Supervised learning means given a bunch of labels, for each incoming unlabeled data, they need to output their corresponding label based on the learning process from the labeled data. Hence, for every data point from test data, it will look for k nearest points from other points(training data), then assign the class in majority. Furthermore, if choose k = 1, that means the test data point is belonging to the class which its nearest point has, and conversely if choose k = n, that means the test data point is belonging to the class which has the highest occuring probability within the whole dataset.

2.4 Train the classifier

Loop over various values of k(here is 30) for the sklearn KneighbourClassifier, and record accuracies in a list.

2.5 Evaluating on test data

Using classification_report and confusion matrix to demonstrate the accuracy of the classifier.

3 Algorithm implemented

3.1 Libraries

1. Sklearn
2. numpy

3.2 Load dataset

```
mnist = datasets.load_digits()
```

3.3 Split dataset

```
(trainData, testData, train-
Labels, testLabels) = train_test_split(np.array(mnist.data),
mnist.target, test_size=0.25, random_state=42)
(trainData, valData, trainLabels, valLa-
bels) = train_test_split(trainData, trainLabels,
test_size=0.1, random_state=84)
```

3.4 Train Classifier

```
for k in range(1, 30, 2):
        model = KNeighborsClassifier(n_neighbors=k)
        model.fit(trainData, trainLabels)
        score = model.score(valData, valLabels)
        print("k=%d, accuracy=%.2f%%" % (k, score * 100))
        accuracies.append(score)
```

3.5 Evaluating on test data

```
model = KNeighborsClassifier(n_neighbors=kVals[i])
model.fit(trainData, trainLabels)
predictions = model.predict(testData)

print("EVALUATION ON TESTING DATA")
print(classification_report(testLabels, predictions))
```

```
print ("Confusion matrix")
print(confusion_matrix(testLabels,predictions))
```

4 Discuss

Advantages:
1. KNN is easy to implement
2. Training is fast
3. Don’t lose Information

Disadvantages:
1. Slow at query time: KNN needs to scan entire training data to derive a prediction.
2. It may be fooled by noisy data because It will go through all data points.

5 References

[1] https://www.kaggle.com/marwaf/handwritten-digits-classification-using-knn