Final Project

CS 677

Huijun Hong

**Introduction**

Digital recognition is the computer's ability to receive and understand and recognize readable numbers from paper documents, photographs, or other sources. According to the different ways of digital source, the current digital recognition problem can be distinguished into handwriting digital recognition, print digital recognition, optical digital recognition, digital recognition in natural scenes. It has great practical application value. My project’s goal is using KNN and SVM to realize hand-written numeral recognition in python, then compare the accuracy of them. I choose one of the most common-used sources of data for handwritten numbers: MNIST [1].

Since MNIST include 60000 training samples and 10000 test samples. If we train directly on the original data set, then there will be too much data and cost much time for training , so I scale down to 1/10 to MNIST: 6000 training samples and 1000 test samples. Each image size is 28 x 28 pixels, so the feature dimension will be 784.

**Design and Process**

In KNN, the training is divided into three stages:

1, load the data set: we use our own data set.

2, visual data description: sample size, image length, input and output characteristics.

3, classification stage: through training can get the classifier, and then use the test set for accuracy calculation.

We construct a KNN classifier knn\_classifier, pass the data from the training set into the constructed knn\_classifier, and predict the results from the test set, compare with the results of the test set, and get the KNN classifier accuracy. Then we get final KNN training and test accuracy were 95.32% and 94.30%.

In SVM, we use train\_test\_split functions to divide training and test sets randomly. After that, we choose SVC for our classifier. Then, we have three steps:

1. loading set:

2. training set: build models with training set training.

3. test set: verify model accuracy with test sets.

We use classifier.fit() in training data and then save data for calculating. Use score to evaluate the training and test accuracy after learning. Then we get final SVM training and test accuracy were 98.43% and 95.40%.

**Conclusion**

We can find SVM training and test accuracy are both better than KNN. KNN's principle is to find the k values in the training dataset that are closest to the sample points that need to be predicted, that makes we can't have a statistical estimate of the predicted error and the results can be very volatile. That is why SVM has better training and test accuracy.

**Link:**

[1] http://yann.lecun.com/exdb/mnist/