Project Report

CS229 Wei Wang #861194381

The classification of handwritten letters is a multi-class task. There are a few methods to achieve multi-class classification, including KNN, Neural Network, multi-class SVM, convolutional Neural Network.

(1) First Approach : KNN is the simplest the method to be implemented. Thus, it is chosen to be the first approach. The data contains 1 column for the end of word and 128 pixel points. Intuitively, without considering anything about the location of the letter, this feature is dropped directly. Every column of pixels is directly taken as one feature. The 10-fold cross-validation error with respect to the number of neighbors is plot as following:



The K=1 or 3 has the smallest error rate. There are 129 features used in the classification. However, some features may variate in a similar behavior, and some features don’t provide any useful information. For example, in the graph of each letter, there may be some common white space, which will not be useful to distinguish different letters.



It may be a good idea to reduce the dimension using PCA. From the reconstructed plot with principal component, we can see that the first 80 components can almost approximate original feature. And we can see that the white space on the two sides are not affected. Therefore, 1NN with different number of principal components are investigated with 10 fold cross-validation.



As we can see from the plot, the 1NN achieves the best error rate with the first 40 principal components. For KNN, the non-significant features add more noise to the distance difference, as the Euclidian distance are added with the same weight.



From the plot of wrong label for letter ‘a’, we can see the misclassification to letter ‘d’ is decreased. As we can think, the direct pixel compared will make some biased letter ‘a’ has a closer distance to letter ‘d’ as the pixel difference all have the same significance. For example, letter ‘a’ with a larger height would make it more ‘like’ letter ‘d’ in pixel comparison. However, with PCA, some more significant feature in a transformed space are abstracted, which makes some ‘biased’ a further from letter ‘d’.

(2) Second Approach: KNN+HMM. For the first approach, the first feature of letter location is dropped directly. However, in handwriting, there are many ambiguous letters, for example, letter ‘l’ and ’i’, and letter ‘h’ and ‘b’ share a lot of characteristics. Therefore, it would be very likely to have misclassification with these letters. However, the words in real world are always have limited possible combining sequences. If the words are model as Markov chains of characters, then we can use the hidden Markov model method to find the most possible sequence to correct the ambiguous letter in a word.

The KNN plus HMM correction method is as follows:

1. Use KNN as a base classifier.
2. The transition probabilities of HMM is modeled on the training dataset. The emission probabilities is model on the base classifier.
3. The base classifier makes prediction on the test set first.
4. The Viterbi algorithm is adopted to correct the predictions with HMM built in previous steps.

For estimating the transition matrix and emission matrix the following equation are used:

Define the number of transition from letter i to j, Thus transition probability matrix can be obtained with . Define the the number of starting with letter I, the starting probability is . Define the times of letter i is classified as j, Thus transition probability matrix can be obtained with .

|  |  |
| --- | --- |
| Classifier | 10-fold cross-validation error |
| 1NN | 18.11% |
| 1NN+ PCA | 15.09% |
| 1NN +PCA+ HMM | 10.54% |

From the table above we can see that, the 1NN+PCA+HMM improve the performance of 1NN+PCA by 5%. From the test result, we can see that ‘r’ is easily misclassified as ‘v’. But with the constraint of a word sequences, the misclassified ‘r’ are corrected by HMM very well.



(3) Conclusion: The PCA helps abstract the most significant features and increase the difference of letters in a transformed space. The HMM correction add the word sequence constraint to the classifier and improve the performance.

References: [1] Discriminativ Training for HMM-Based Offline Handwritten Character Recognition, Roongroj Nopsuwanchai et. al., Computer Laboratory, University of Cambridge, {CB3 0FD, CB2 1PZ}, UK

[2] Using HMMs to boost accuracy in optical character recognition, Prasanna Velagapudi