CSE 514 Programming Assignment 2 Report

Xiaoyan Zheng

November 29, 2022

Contents

1 Introduction

1.1 Description of the problem and the practical impacts

The problem we have is letter recognition. We are given 16 features of a image of a handwritten capital letter, and we would like to classify this object as one of the letters in the English alphabet.

A possible practical application of this problem is to transcribe handwritten text to plain print text for privacy protection.

1.2 Motivation for multiple classifiers

There are many possible classifiers that does the job of classifying the letters, and we would like to pick the best one out of them. When evaluating a model, several factors should be considered. Accuracy of classification is definitely one of the most important one, as we would like the classifier to recognize the letters correctly. Accuracy of the model is measured on its accuracy on the testing dataset. Efficiency (time and space required to train the model and to classify new instances) is also an important consideration. A model that takes forever to be trained is considered less useful than a model with slightly less accuracy but runs much faster.

1.3 Motivation for dimension reduction

Dimension reduction is to transform data from high dimensions to low dimensions. Having a higher dimensional dataset requires large storage space, and training models on higher dimensional data is usually computationally costly and hard to control. Moreover, raw data are often sparse as a consequence of the curse of dimensionality, which makes it difficult to reveal the underlying pattern of the data.

A good dimension reduction method should be effective in selecting features. For example a simple quality filtering would not be a good method for the data that we are using because the dataset have been cleaned and there's no missing values, which means simple quality filtering is unlikely to filter out any features for us. A good method should also retains the meaningful properties of the original data, ideally close to its intrinsic dimension, and filter out irrelevant features.

1.4 Dimension reduction methods chosen

The first method I chose is Wrapper feature selection with Greedy Forward Feature Construction. The method starts by fitting the model with only one feature in the dataset. After evaluating the models for each feature, it picks the feature that gives the best model. Then the method iteratively add features to the training data such that at each step the new feature gives the best result among the remaining features.

The second method I chose is Embedded methods which is intrinsically performed by a decision tree. By pruning the tree or restrict the number of split of the tree, the model automatically choose the features that maximize decrease in Gini Impurity or Variance.

1.5 Binary classification problems

I chose 'O' and 'Q' as the third pair of classification problem. I would expect this pair to be hardest to classify as these two letters looks very similar. Following the same logic, I would expect 'M' and 'Y' pair to be the easiest to classify.

1.6 EC Multi-class classifier

When using binary classifiers, we can select the relevant features and tune parameter for each classifier, because the relevant features for discriminating class A from class B can be different from the relevant features for discriminating A from C. In contrast, when doing multi-class classifier, we might need to incorporate more features at the cost of computational efficiency and risk of overfitting for better accuracy.

The advantage of using a multi-class classifier is that we only need to train and use a single classification model instead of a bunch binary classifiers, which make interpretation of the results easier.

2 Results

2.1 K-nearest neighbors classifier

2.1.1 Model description

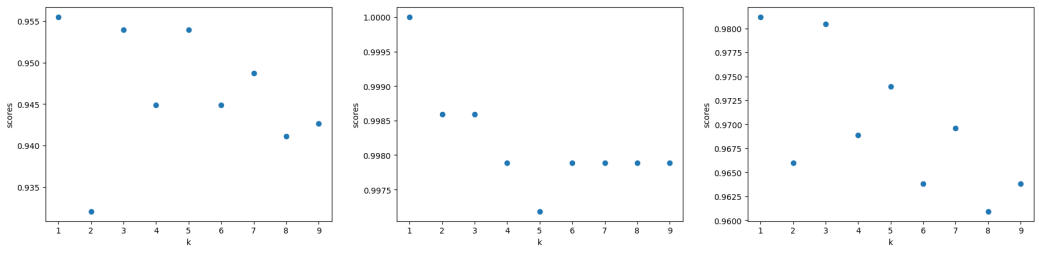
K-nearest neighbors classifier classifies a new sample with the the majority label of the k training samples that are closest to this sample.

The advantage of knn is that there's no training period, so that the model usually performs much faster than other algorithms. Also, knn is easy to interpret and requires very few parameters.

The disadvantage of knn is that calculating distances in large dataset can be very costly, and it does not work well with high dimensionality. It is also sensitive to noise and missing data.

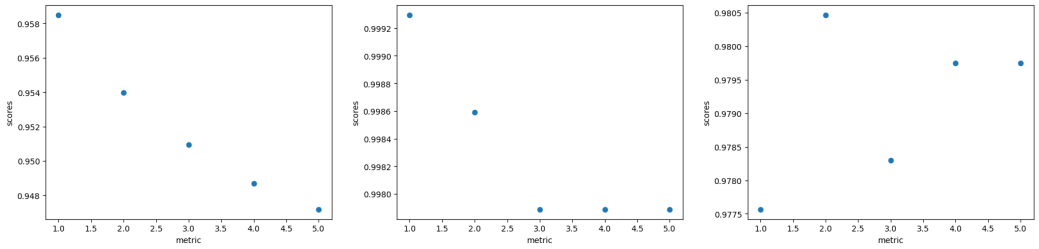
2.1.2 Figure of CV results

The first parameter I tuned for K-nearest neighbors classifier is k from 1 to 9. The results are the following. The three plots are for HK, MY, and OQ respectively.



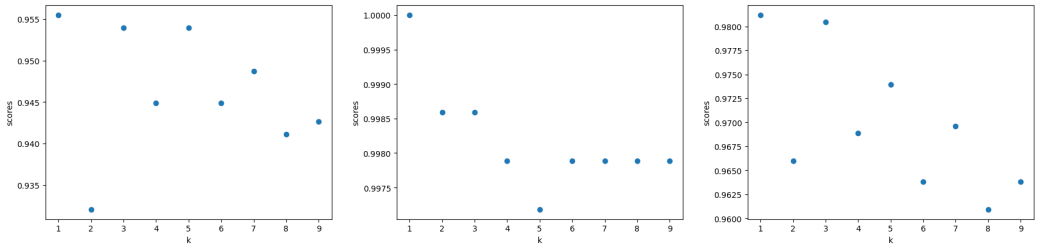
2.1.3 Figure of CV results for additional hyperparameter

The second parameter I tuned for K-nearest neighbors classifier is metric. I tested l_p norms for $p = 1, 2, 3, 4, 5$. The results are the following. The three plots are for HK, MY, and OQ respectively.

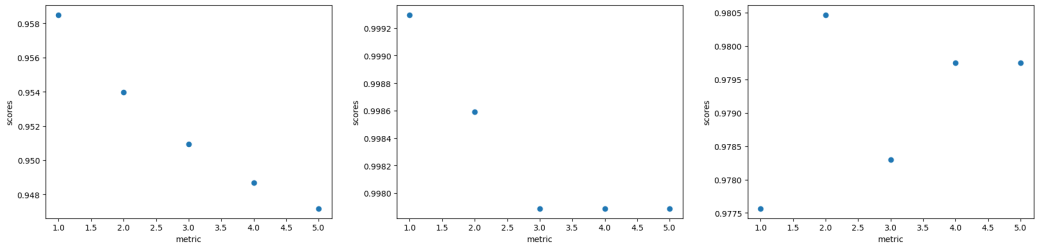


2.1.4 Figure of CV results with dimension reduction

Validation result after dimension reduction are the following. The first parameter tuned is k from 1 to 9. The three plots are for HK, MY, and OQ respectively.



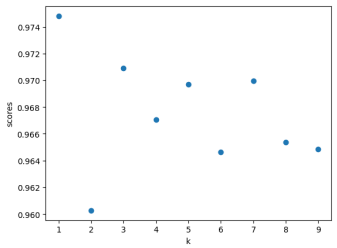
The second parameter tuned is metric (l_p norm for $p = 1, \cdots, 5$). The three plots are for HK, MY, and OQ respectively.



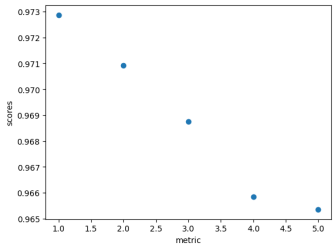
2.1.5 Multi-class classification

Validation result for multi-class classification are the following.

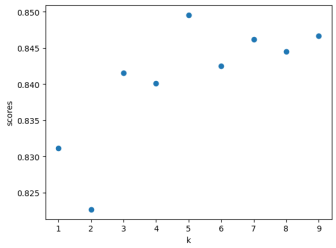
Tuning for k without dimension reduction:



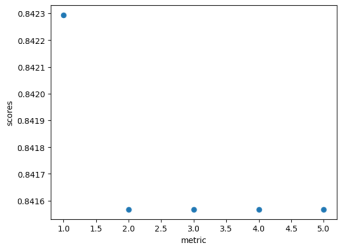
Tuning for metric without dimension reduction:



Tuning for k with dimension reduction:



Tuning for metric without dimension reduction:



2.2 Decision Tree

2.2.1 Model description

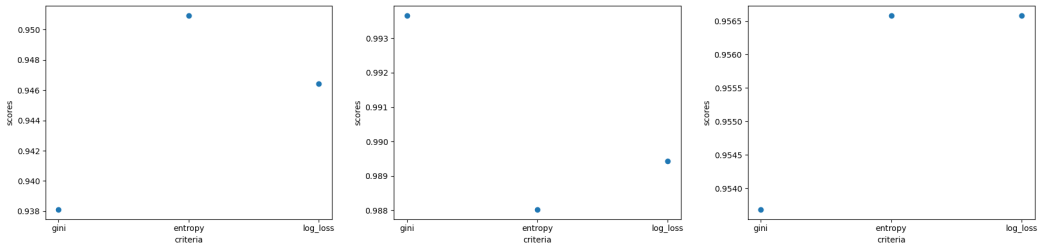
Decision tree represent the decision model in as a tree, where each internal node of the tree denotes an attribute and each leaf node denotes a class label. The tree split on some features of the training data to achieve maximal accuracy at the leaf nodes at each level.

The advantage of decision tree is that it requires less effort for data preparation such as normalization during pre-processing.

The disadvantage of decision tree is that calculation can be far more complex compared to other algorithms, so training decision tree is relatively expensive.

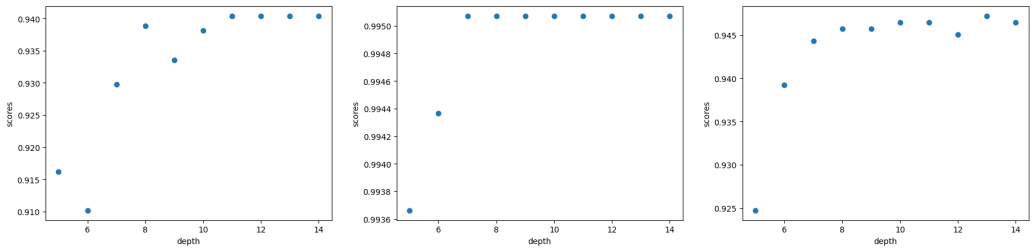
2.2.2 Figure of CV results

The first parameter I tuned for Decision Tree classifier is the splitting criteria. The results are the following. The three plots are for HK, MY, and OQ respectively.



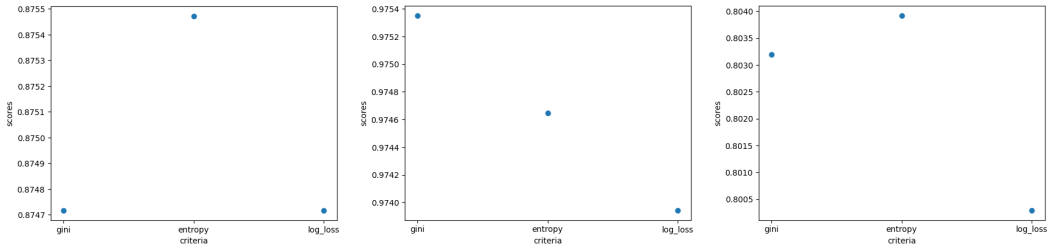
2.2.3 Figure of CV results for additional hyperparameter

The second parameter I tuned for Decision Tree classifier is max depth of the tree, from 5 to 14. The results are the following. The three plots are for HK, MY, and OQ respectively.

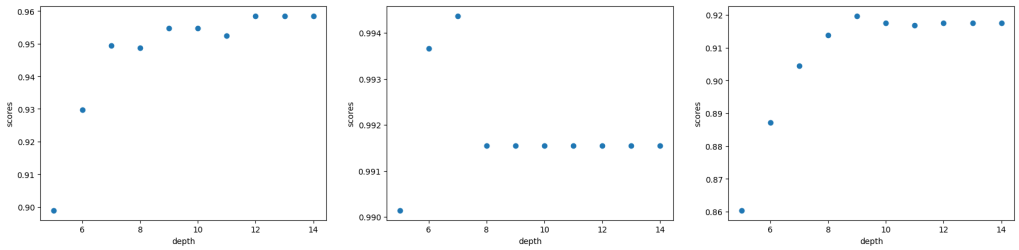


2.2.4 Figure of CV results with dimension reduction

Validation result after dimension reduction are the following. The first parameter tuned is splitting criteria. The three plots are for HK, MY, and OQ respectively.



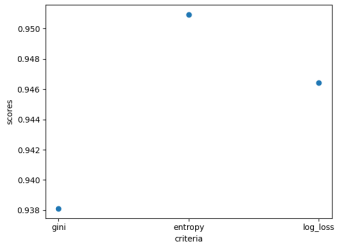
The second parameter tuned is max depth from 5 to 14. The three plots are for HK, MY, and OQ respectively.



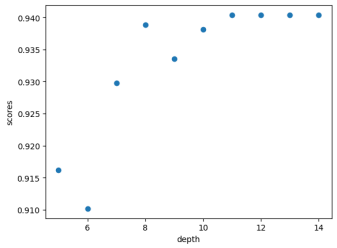
2.2.5 Multi-class classification

Validation result for multi-class classification are the following.

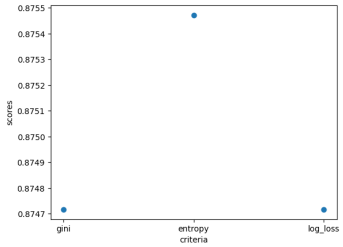
Tuning for splitting criteria without dimension reduction:



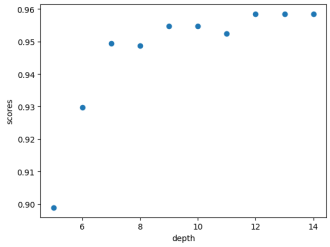
Tuning for max depth without dimension reduction:



Tuning for splitting criteria with dimension reduction:



Tuning for max depth with dimension reduction:



3 Discussion

3.1 Performance comparison of different classifiers

The parameters picked for each model are those who gave highest scores in the cross validation part. Overall, the performance of knn classifier is better than decision tree, and knn also takes less time to train. Among the three pairs of classification problem, “MY” pair seems to be the easiest to classify for both models; “HK” pair is better classified by knn than by decision tree, and the

converse holds for "OQ" pair. Knn also does better for multi-class classification problem both in terms of time and accuracy.

	knn times	knn scores	Decision Tree times	Decision Tree scores
HK	0.008013	0.939189	0.013963	0.939189
MY	0.005989	1.000000	0.010969	0.974684
OQ	0.007874	0.974026	0.010969	0.922078
full	0.013115	0.973856	0.037825	0.934641

3.2 Performance comparison after dimension reduction

After dimension reduction, the run time of decision tree reduced significantly, whereas the run time for knn did not reduce. As a result, the run time for decision tree is now faster than knn. However, knn have higher accuracy for all three binary classification pairs and for multi-class classification. "MY" pair remains to be the easiest to classify. The effect of reduced dimension on decreased scores is more evident on decision tree than on knn. Also, the score decreased more significantly for the multi-class classification models than for the binary classification models.

	knn times	knn scores	Decision Tree times	Decision Tree scores
HK	0.014959	0.925676	0.007012	0.878378
MY	0.012599	0.987342	0.004995	0.943038
OQ	0.008978	0.922078	0.005954	0.805195
full	0.020026	0.867102	0.012936	0.657952

3.3 Lessons learned

I would choose knn classifier for this problem. Even though it takes slightly longer time to run than decision tree classifier, the accuracy is significantly higher in most cases.

Dimension reduction affects the performance of decision tree more, both in terms of time and accuracy. After dimension reduction, the run times decreases as well as the accuracy. The effect is most significant for the multi-class classification problem, where the accuracy for knn dropped from 97% to 86% and accuracy of decision tree dropped from 93% to 65%.

Given a new dataset for same task, it would be more efficient if we skip the dimension reduction step and use whatever feature we chose this time. Since we are given the same task, it is likely that the features most relevant to this task remains the same for the new dataset. Thus we do not need to re-do the feature selection.