CS 519 Project 3 Report

Xiaonan Zhu

In this project, I utilized five sciklit-learn classifiers to two datasets, digits and a time-series data set, subject1_ideal.log, from REALDISP Activity Recognition Dataset.

1. Preceptron
   (1) digits dataset
   When Learning rate is 0.1, Iteration is 5,
   Misclassified samples: 53.
   The accuracy is 90.19%.
   The running time is 0.4498 seconds.

   When Learning rate is 0.1, Iteration is 20,
   Misclassified samples: 37.
   The accuracy is 93.15%.
   The running time is 0.4597 seconds.

   When Learning rate is 0.01, Iteration is 5,
   Misclassified samples: 53.
   The accuracy is 90.19%.
   The running time is 0.4398 seconds.

   When Learning rate is 0.01, Iteration is 40,
   Misclassified samples: 37.
   The accuracy is 93.15%.
   The running time is 0.4797 seconds.

   (2) subject1_ideal
   When Learning rate is 0.1, Iteration is 5,
   Misclassified samples: 1691.
   The accuracy is 96.85%.
   The running time is 11.7240 seconds.

   When Learning rate is 0.1, Iteration is 20,
   Misclassified samples: 1334.
   The accuracy is 97.52%.
   The running time is 27.3642 seconds.

   When Learning rate is 0.01, Iteration is 5,
   Misclassified samples: 1691.
   The accuracy is 96.85%.
   The running time is 11.6482 seconds.

When Learning rate is 0.01, Iteration is 40,
Misclassified samples: 1331.
The accuracy is 97.52%.
The running time is 52.6101 seconds.

Analysis: The performance of Preceptron is good for both two datasets. Smaller learning rate and more iterations can improve predictions.

2. SVM
   (1) digits dataset

When gamma is 0.8, C is 0.5,
Misclassified samples: 485.
The accuracy is 10.19%.
The running time is 0.7995 seconds.

When gamma is 0.8, C is 1,
Misclassified samples: 461.
The accuracy is 14.63%.
The running time is 0.8196 seconds.

When gamma is 0.8, C is 5,
Misclassified samples: 451.
The accuracy is 16.48%.
The running time is 0.8095 seconds.

When gamma is 0.2, C is 0.5,
Misclassified samples: 353.
The accuracy is 34.63%.
The running time is 0.8595 seconds.

When gamma is 0.2, C is 1,
Misclassified samples: 110.
The accuracy is 79.63%.
The running time is 0.7596 seconds.

When gamma is 0.2, C is 5,
Misclassified samples: 101.
The accuracy is 81.30%.
The running time is 0.7896 seconds.

When gamma is 0.02, C is 0.5,
Misclassified samples: 11.
The accuracy is 97.96%.
The running time is 0.5797 seconds.

When gamma is 0.02, C is 1,
Misclassified samples: 6.
The accuracy is 98.89%.
The running time is 0.5797 seconds.

When gamma is 0.02, C is 5,
Misclassified samples: 9.
The accuracy is 98.33%.
The running time is 0.5797 seconds.

(2) subject1_ideal
When gamma is 0.02, C is 5
The SVM is very very slow to time-series dataset. My code is running over 30 hours.

Analysis: We can see that gamma and C will affect prediction a lot. The smaller gamma and the bigger C, the better the prediction. But predictions are bad if gamma is relatively big.

3. DecisionTree
(1) digits dataset
When max_depth=4, min_samples_leaf=1
Misclassified samples: 249.
The accuracy is 53.89%.
The running time is 0.4198 seconds.

When max_depth=4, min_samples_leaf=5,
Misclassified samples: 248.
The accuracy is 54.07%.
The running time is 0.4198 seconds.

When max_depth=40, min_samples_leaf=1,
Misclassified samples: 79.
The accuracy is 85.37%.
The running time is 0.4398 seconds.

When max_depth=40, min_samples_leaf=5,
Misclassified samples: 91.
The accuracy is 83.15%.
The running time is 0.4198 seconds.

(2) subject1_ideal
When max_depth=4, min_samples_leaf=1
Misclassified samples: 10181.
The accuracy is 81.05%.
The running time is 14.5997 seconds.

When max_depth=4, min_samples_leaf=5,
Misclassified samples: 10181.
The accuracy is 81.05%.
The running time is 14.6740 seconds.

When max_depth=40, min_samples_leaf=1,
Misclassified samples: 299.
The accuracy is 99.44%.
The running time is 53.0190 seconds.

When max_depth=40, min_samples_leaf=5,
Misclassified samples: 426.
The accuracy is 99.21%.
The running time is 54.1449 seconds.

Analysis: The performance of decision tree on subject1_ideal is much better than digits dataset, because the sample size of subject1_ideal is very large. From digits dataset, we can see that if sample size is small, max_depth affects predictions.

4. KNN
   (1) digits dataset
   When n_neighbors=1, p=2,
   Misclassified samples: 12.
   The accuracy is 97.78%.
   The running time is 0.4598 seconds.

   When n_neighbors=5, p=2,
   Misclassified samples: 10.
   The accuracy is 98.15%.
   The running time is 0.4598 seconds.

   When n_neighbors=10, p=2,
   Misclassified samples: 11.
   The accuracy is 97.96%.
   The running time is 0.4597 seconds.

   When n_neighbors=1, p=1,
   Misclassified samples: 10.
   The accuracy is 98.15%.
   The running time is 0.4497 seconds.

   When n_neighbors=5, p=1,
   Misclassified samples: 15.
   The accuracy is 97.22%.

The running time is 0.4598 seconds.

When n_neighbors=10, p=1,
Misclassified samples: 18.
The accuracy is 96.67%.
The running time is 0.4597 seconds.

(2) subject1_ideal
When n_neighbors=5, p=2,
Misclassified samples: 278.
The accuracy is 99.48%.
The running time is 15790.8811 seconds.

Analysis: n_neightbors and p don't affect predictions a lot. But for time-series dataset, the running time is long.

5. Log Regression
   (1) digits dataset
   When C=0.01,
   Misclassified samples: 32.
   The accuracy is 94.07%.
   The running time is 0.4797 seconds.

   When C=1
   Misclassified samples: 20.
   The accuracy is 96.30%.
   The running time is 0.5697 seconds.

   When C=100
   Misclassified samples: 24.
   The accuracy is 95.56%.
   The running time is 0.7159 seconds.

   (2) subject1_ideal
   When C=0.01
   Misclassified samples: 1846.
   The accuracy is 96.56%.
   The running time is 198.7072 seconds.

   When C=1
   Misclassified samples: 558.
   The accuracy is 98.96%.
   The running time is 459.2291 seconds.

When C=100

Misclassified samples: 180.

The accuracy is 99.66%.

The running time is 861.8685 seconds.

Analysis: For logistic regression, the bigger C the better the prediction. But for time-series dataset, the running time is long.

For Decision Tree, two strategies to pre-prune or post-prune the tree could be

(1) set max_leaf_nodes (pre-prune) and (2) set max_depth (pre-prune)

They can be implemented by codes in source file from line 348 to line 363

```
# Use BestFirst if max_leaf_nodes given; use DepthFirst otherwise
        if max_leaf_nodes < 0:
            builder = DepthFirstTreeBuilder(splitter, min_samples_split,
                                            min_samples_leaf,
                                            min_weight_leaf,
                                            max_depth,
                                            self.min_impurity_decrease,
                                            min_impurity_split)
        else:
            builder = BestFirstTreeBuilder(splitter, min_samples_split,
                                           min_samples_leaf,
                                           min_weight_leaf,
                                           max_depth,
                                           max_leaf_nodes,
                                           self.min_impurity_decrease,
                                           min_impurity_split)
```