Classification Methods

This report consists of the description of following tasks:

1. Empirical study on random forest
2. Empirical study on SVM
3. Empirical study on neural network model

Environment



|  |  |
| --- | --- |
| RAM | 16GB |
| CPU | Intel i7-6700HQ |
| System | Windows 10 64 bit |

Run time

|  |  |
| --- | --- |
| Program | Run time(s) |
| Random forest model | 2.0167622566223145 |
| SVM with RBF kernel | 5974.0660808086395 |
| Neural network model | 382.55888175964355 |

Empirical study on Random forest model 2.0167622566223145seconds

train accruacy score is 99.49%

test accruacy score is 82.2%

Classification report for classifier RandomForestClassifier(bootstrap=True, class\_weight=None, criterion='gini',

max\_depth=None, max\_features='auto', max\_leaf\_nodes=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None,

min\_samples\_leaf=1, min\_samples\_split=2,

min\_weight\_fraction\_leaf=0.0, n\_estimators=10, n\_jobs=None,

oob\_score=True, random\_state=10, verbose=0, warm\_start=False):

precision recall f1-score support

0 0.76 0.79 0.77 107

1 0.98 0.95 0.97 105

2 0.68 0.79 0.73 111

3 0.76 0.84 0.80 93

4 0.82 0.70 0.76 115

5 0.92 0.87 0.89 87

6 0.59 0.52 0.55 97

7 0.87 0.89 0.88 95

8 0.98 0.97 0.97 95

9 0.91 0.93 0.92 95

micro avg 0.82 0.82 0.82 1000

macro avg 0.83 0.83 0.82 1000

weighted avg 0.82 0.82 0.82 1000

Confusion matrix:

[[ 84 0 3 7 0 0 13 0 0 0]

[ 0 100 0 5 0 0 0 0 0 0]

[ 3 0 88 1 9 0 10 0 0 0]

[ 6 1 0 78 1 0 5 0 2 0]

[ 0 0 21 7 81 0 6 0 0 0]

[ 0 0 0 0 0 76 0 8 0 3]

[ 17 0 18 4 8 0 50 0 0 0]

[ 0 0 0 0 0 4 0 85 0 6]

[ 0 1 0 0 0 1 1 0 92 0]

[ 0 0 0 0 0 2 0 5 0 88]]

Empirical study on SVM 5974.0660808086395seconds

The best classifier is: SVC(C=1.0, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape='ovr', degree=3, gamma=0.001, kernel='rbf',

max\_iter=-1, probability=False, random\_state=None, shrinking=True,

tol=0.001, verbose=False)

param\_gamma: [0.001 0.01 0.1 1]

mean\_test\_score: [0.1027 0.1027 0.1027 0.1027]

std\_test\_score: [0.00013569 0.00013569 0.00013569 0.00013569]

mean\_train\_score: [1. 1. 1. 1.]

std\_train\_score [0. 0. 0. 0.]

Therefore, optimal γ = 0.001

train accruacy score is 100%

test accruacy score is 10.5%

Classification report for classifier SVC(C=1.0, cache\_size=200, class\_weight=None, coef0=0.0,

decision\_function\_shape='ovr', degree=3, gamma=0.001, kernel='rbf',

max\_iter=-1, probability=False, random\_state=None, shrinking=True,

tol=0.001, verbose=False):

precision recall f1-score support

0 0.00 0.00 0.00 107

1 0.10 1.00 0.19 105

2 0.00 0.00 0.00 111

3 0.00 0.00 0.00 93

4 0.00 0.00 0.00 115

5 0.00 0.00 0.00 87

6 0.00 0.00 0.00 97

7 0.00 0.00 0.00 95

8 0.00 0.00 0.00 95

9 0.00 0.00 0.00 95

micro avg 0.10 0.10 0.10 1000

macro avg 0.01 0.10 0.02 1000

weighted avg 0.01 0.10 0.02 1000

Confusion matrix:

[[ 0 107 0 0 0 0 0 0 0 0]

[ 0 105 0 0 0 0 0 0 0 0]

[ 0 111 0 0 0 0 0 0 0 0]

[ 0 93 0 0 0 0 0 0 0 0]

[ 0 115 0 0 0 0 0 0 0 0]

[ 0 87 0 0 0 0 0 0 0 0]

[ 0 97 0 0 0 0 0 0 0 0]

[ 0 95 0 0 0 0 0 0 0 0]

[ 0 95 0 0 0 0 0 0 0 0]

[ 0 95 0 0 0 0 0 0 0 0]]

Empirical study on neural network model: 382.55888175964355seconds

The best classifier is: MLPClassifier(activation='relu', alpha=0.0001, batch\_size='auto', beta\_1=0.9,

beta\_2=0.999, early\_stopping=False, epsilon=1e-08,

hidden\_layer\_sizes=50, learning\_rate='constant',

learning\_rate\_init=0.001, max\_iter=200, momentum=0.9,

n\_iter\_no\_change=10, nesterovs\_momentum=True, power\_t=0.5,

random\_state=None, shuffle=True, solver='sgd', tol=0.0001,

validation\_fraction=0.1, verbose=False, warm\_start=False)

param\_hidden\_layer\_sizes: [1 5 10 20 50]

mean\_test\_score: [0.1023 0.1027 0.1208 0.1477 0.2037]

std\_test\_score: [0.00031336 0.00013569 0.03763776 0.06525324 0.09891392]

mean\_train\_score: [0.10234999 0.10275001 0.12089762 0.15152389 0.212629 ]

std\_train\_score [3.25758675e-04 3.54084835e-05 3.77889389e-02 7.03958639e-02

1.10755082e-01]

Therefore, the optimal H = 50

train accruacy score is 0.3209

test accruacy score is 0.292

Classification report for classifier MLPClassifier(activation='relu', alpha=0.0001, batch\_size='auto', beta\_1=0.9,

beta\_2=0.999, early\_stopping=False, epsilon=1e-08,

hidden\_layer\_sizes=50, learning\_rate='constant',

learning\_rate\_init=0.001, max\_iter=200, momentum=0.9,

n\_iter\_no\_change=10, nesterovs\_momentum=True, power\_t=0.5,

random\_state=10, shuffle=True, solver='sgd', tol=0.0001,

validation\_fraction=0.1, verbose=False, warm\_start=False):

precision recall f1-score support

0 0.00 0.00 0.00 107

1 0.14 1.00 0.25 105

2 0.00 0.00 0.00 111

3 0.00 0.00 0.00 93

4 0.00 0.00 0.00 115

5 0.92 0.76 0.83 87

6 0.00 0.00 0.00 97

7 0.79 0.32 0.45 95

8 0.00 0.00 0.00 95

9 0.65 0.96 0.78 95

micro avg 0.29 0.29 0.29 1000

macro avg 0.25 0.30 0.23 1000

weighted avg 0.23 0.29 0.21 1000

Confusion matrix:

[[ 0 106 0 0 0 0 0 1 0 0]

[ 0 105 0 0 0 0 0 0 0 0]

[ 0 111 0 0 0 0 0 0 0 0]

[ 0 93 0 0 0 0 0 0 0 0]

[ 0 115 0 0 0 0 0 0 0 0]

[ 0 3 0 0 0 66 0 6 0 12]

[ 0 97 0 0 0 0 0 0 0 0]

[ 0 21 1 0 0 4 3 30 0 36]

[ 0 95 0 0 0 0 0 0 0 0]

[ 0 1 0 0 0 2 0 1 0 91]]

Experiment results

1. Run time

Random forest is the fastest, the second is neural network, the slowest is SVM with RBF kernel

1. Accuracy

Random Forest provide the best accuracy result with 99.49% and 82.2% on training set and test set

Neural Network gives a unsatisfied result of 32.09% and 29.2% on training set and test set.

SVM gives the lowest accuracy on test set and seems to overfit on the training set

Reference

<http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html><http://ogrisel.github.io/scikit-learn.org/sklearn-tutorial/auto_examples/svm/plot_svm_parameters_selection.html>