Machine Learning Homework 3

Basic Idea:

In this project, we are implementing a classifier(SVM), to learn a module from existing data set, and make prediction to another data set.

There are all together 14 features in the origin data set, each feature is divided by comma. Each feature has its own range, numerical or alphabetical. Our goal is to use these data to train a rule, which will classify the data set best.

1. Preprocess the data set:

The given data set has numbers and strings in different scale, and even some missing value. So the first step is to preprocess these data, deal with the missing value, encode the string-features so that they can be represented in formula. In my program, OneHotEncoder was used to encode string-features. They these features should be compressed into a same range, so that the value of ‘age, 20’ doesn’t mean half the importance of ‘work hour 40’.

1. Choose a SVM training module and run the SVM algorithm using K-folds.

In this project, I used three kernel function to train the rule: polynomial kernel, linear kernel, and RBF kernel. Each kernel has different meaning and different parameters needed. The detailed parameter and performance is shown below.

K-folds approach is used to prevent from over fitting. The best predicting module will be chosen to predict new data set.

1. Predict new data set using learnt classifier:

With the best existed classifier, new data set can be predicted. Data samples will be preprocessed first, using the same way with labeled data set. And then by using the best SVM module we just found, labels are given out, for each data sample.

|  |  |  |
| --- | --- | --- |
| **Linear Kernel’s performance** | | |
| **Sequence Number** | **C** | **Accuracy (%)** |
| 1 | 1.0 | 84.86 |
| 2 | 0.0625 | 84.88 |
| 3 | 0.03 | 84.90 |
| 4 | 0.01 | 84.94 |
| 5 | 0.00625 | 84.88 |
| 6 | 0.003 | 84.71 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Polynomial Kernel’s performance** | | | | |
| **Sequence Number** | **C** | **Degree** | **Gamma** | **Accuracy (%)** |
| 1 | 1.0 | 1 | 1.0 | 84.86 |
| 2 | 0.01 | 2 | 0.0095 (1/105) | 76.54 |
| 3 | 0.01 | 3 | 0.0095 (1/105) | 76.54 |
| 4 | 0.5 | 1 | 0.0095 (1/105) | 84.84 |
| 5 | 0.5 | 2 | 0.0095 (1/105) | 83.19 |
| 6 | 1.0 | 1 | 0.0095 (1/105) | 84.95 |
| 7 | 1.5 | 1 | 0.0095 (1/105) | 84.92 |
| 8 | 1.5 | 2 | 0.0095 (1/105) | 84.08 |

|  |  |  |  |
| --- | --- | --- | --- |
| **RBF kernel’s performance** | | | |
| **Sequence Number** | **C** | **Gamma** | **Accuracy (%)** |
| 1 | 1.0 | 1.0 | 81.63 |
| 2 | 16 | 0.0095 (1/105) | 85.27 |
| 3 | 64 | 0.0095 (1/105) | 85.50 |
| 4 | 100 | 0.0095 (1/105) | 85.51 |
| 5 | 128 | 0.0095 (1/105) | 85.49 |
| 6 | 100 | 0.01 | 85.51 |
| 7 | 100 | 0.012 | 85.50 |