Report

CHEN Hang

This lab is to build the SVM model with Kernel function which has the best prediction accuracy. The dataset is a Covertype dataset, it has 7 different labels. We need to divide such dataset into training and test sets considering 70% of data for training and 30% for testing. In short, as we have 7 classes, the goal is to find 7 kernels for 7 different SVM instances. Each kernel will separate its main class from the others.

Because this dataset is so large that my computer could not run the program for all data, I only take 2000 samples from this dataset for testing.

There are 7 labels of the dataset, so we need to build One-Versus-Rest SVMs, in the training, the samples of a certain category are classified into one class, and the other remaining samples are classified into another class. Here I wrote a loop for each label:

```
for (class in 1:7){
        print(paste("The model for ", as.character(class), sep = " "))
        best_model = get_model(class, dataset)
}
```

For each loop, we deal with the data, when testing a label, we put the other labels into "-1" to sparate them.

```
get_single_dataset <- function(class, dataset){
    new_dataset = dataset
    col = ncol(dataset)
    for(i in 1 : nrow(new_dataset)) {
        if(new_dataset[i, col] != class) {
            new_dataset[i, col] = -1
        }
    }
    return(new_dataset)
}
```

Next step is building the SVM models with different Kernels(Polynomial and Gaussian).

As we have got the range of parameters from the course, for Polynomial: degree from 1 to 4, gamma from 0 to 2, coef from 0.01 to 2; for Gaussian: gamma from 0 to 2. So I wrote 3 loops for traversing these parameters and build the corresponding mode, then predict the result and calculate the accuracy.

For Polynomial model:

```
for(d in 1:4){
        for(g in seq(from = 0, to = 2, length.out = 10)) {
            for(c in seq(from = 0.01, to = 2, length.out = 10)) {
            ......
            }
        }
}
```

For Gaussian model:

```
for(g in seq(from = 0, to = 2, length.out = 10)) {
    ...
}
```

Here we will get 2 model, one is the Polynomial Kernel with its highest accuracy and another one is Gaussian Kernel with its highest accuracy. After that, we compare them and choose better one as the final model.

Normally at the end, the program will print the best Kernel model, its parameters and the accuracy. But my computer is so slow that I can't have the final result.

There is the result that I have test other smaller dataset before(but obviously the result is wrong):

For all 7 classes, the best Kernel are Polynomial, and the parameter is:

Label	Degree	Gamma	Coef
1	1	0.0100	0.01
2	1	1.5000	0.01
3	1	1.0000	0.01
4	1	1.0000	0.01
5	1	1.0000	0.01
6	1	1.5000	0.01
7	1	0.0100	0.01