**2 Programming**

In this assignment, we construct several SVM models with different kernels and slack variables to classify the Iris dataset.

The basic form of SVM is given below:

Where is the coefficient of different features, is the intercept of the hyperplane, and are the features and labels of the Iris data.

**2.1 (SVM without slack variables)**

1. The optimization problem

We first get the dual problem of the original problem stated above.

The dual Lagrange function is:

The primal and dual optimal solutions should satisfy KKT conditions:

Stationarity:

Feasibility:

Complementary slackness:

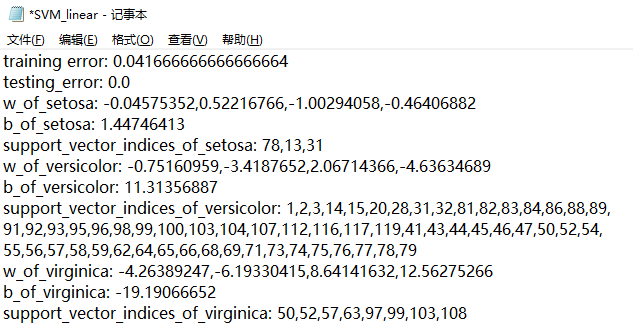
Then, the dual problem can be derived by substituting all the stationary conditions into the primal problem, finally we get:

Also, is given by:

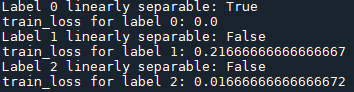
Since ,

2. Data processing and results

I use OneVsRestClassifier and sklearn.svm.SVC to do the following problems. Since sklearn doesn’t provide strict separation, we use C = 1e5, kernel = ‘linear’ to estimate the attributes and calculate errors. The result is shown below.



When determining which class is linearly separable, we first calculate the train loss by 1-training score by the sklearn, then if the train loss is 0, it is linearly separable. The statistics are shown below:



In conclusion, only label 0 (setosa) is linearly separable in the dataset.

**2.2 (SVM with slack variables)**

1. The optimization problem

For SVM with slack variables, we can simply modify it by adding a penalty term.

The dual Lagrange function is:

The primal and dual optimal solutions should satisfy KKT conditions:

Stationarity:

Feasibility:

Complementary slackness:

Then, the dual problem can be derived by substituting all the stationary conditions into the primal problem, finally we get:

Also, is given by:

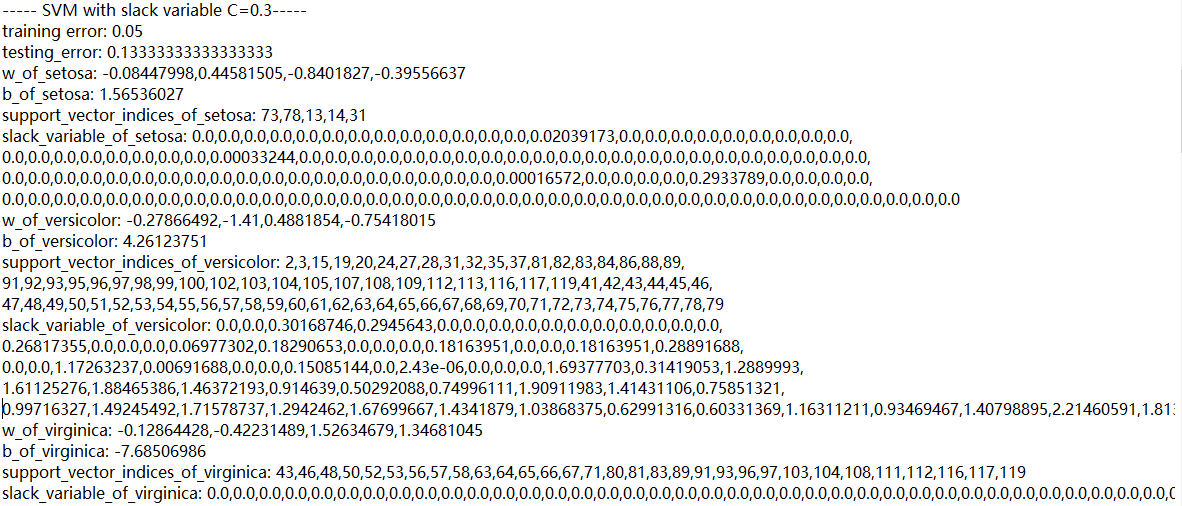
Since ,

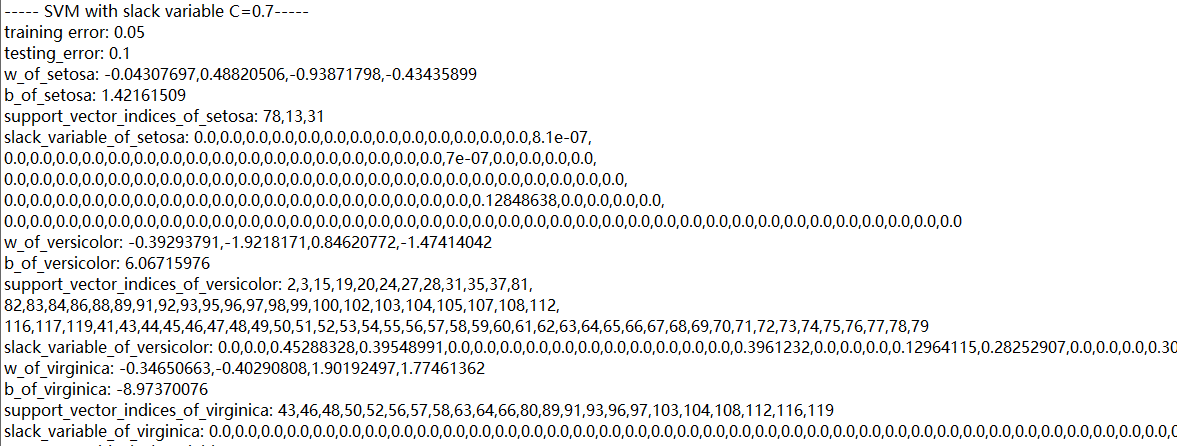
For the slack variables’ calculation, can return the value of the slack variables, where is the label {-1, +1}, is the result calculated by the input vectors and the estimated parameters and .

2. Data processing and results

We use a for loop to cyclically write into the output, when deriving the slack variables’ value, we use the decision\_function(X) provided by sklearn.svn.SVC to directly get the term and then modify the y\_train that it only contains labels of {-1, +1}, finally calculate the value of slack variables.

Below are two sample outputs C = 0.3 and C = 0.7:





**2.3 (SVM with kernels and slack variables)**

1. The optimization problem

The derivation of the dual problem of the kernel is similar to 2.2:

The related kernel functions are:

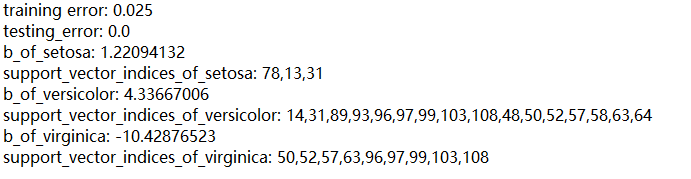
Polynomial kernel:

Radical Basis Function (RBF) kernel:

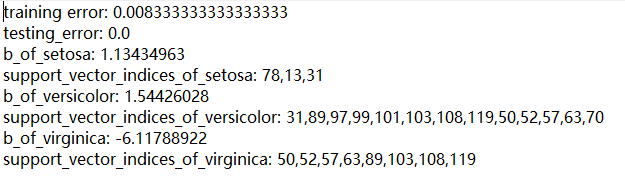
Sigmoidal kernel:

2. Data processing and results

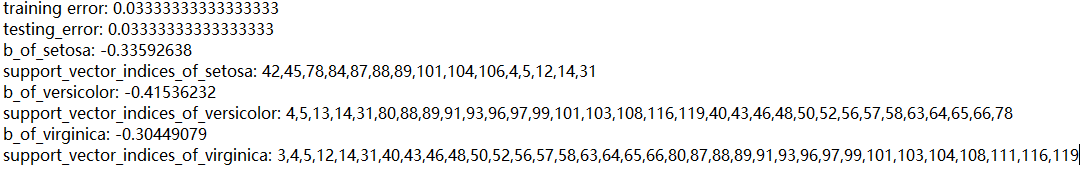
(a) In this scenario, we set C = 1, kernel = 'poly', degree = 2, gamma = 1, and get the following output:



(b) In this scenario, we set C = 1, kernel = 'poly', degree = 3, gamma = 1, and get the following output:



(c) In this scenario, we set C = 1, kernel = 'rbf', gamma = 0.5, and get the following output:



(d) In this scenario, we set C = 1, kernel = 'sigmoid', gamma = ‘auto’ (gamma = 0.25), and get the following output:

