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:

$$min_{w,b}\ \frac{1}{2}\|w\|^2$$

$$s.t.\ 1 - y_i(w^T x_i + b) \leq 0, \forall i$$

Where w is the coefficient of different features, b is the intercept of the hyperplane, $\boldsymbol{x_i}$ and y_i 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:

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 + \sum_i^m \alpha_i(1 - y_i(w^T x_i + b))$$

   The primal and dual optimal solutions should satisfy KKT conditions:

   Stationarity:

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_i^m \alpha_i y_i x_i$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_i^m \alpha_i y_i = 0$$

   Feasibility:

$$\alpha_i \geq 0, 1 - y_i(w^T x_i + b) \leq 0, \forall i$$

   Complementary slackness:

$$\alpha_i(1 - y_i(w^T x_i + b)) = 0, \forall i$$

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

$$\max_\alpha \sum_i^m \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j,$$

$$s.t. \sum_i^m \alpha_i y_i = 0, \alpha_i \geq 0, \forall i$$

   Also, b is given by:

$$y_j(w^T x_i + b) = 1, \forall j \in S$$

$$\Rightarrow y_j\left(\sum_i^m \alpha_i y_i {x_i}^T x_j + b\right) = 1, \forall j \in S$$

Since ${y_j}^2 = 1$,

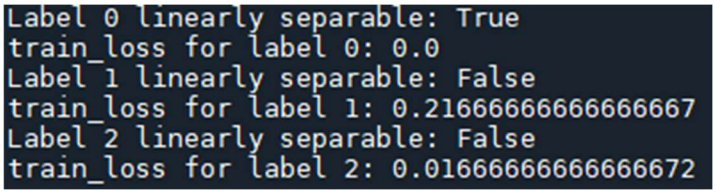
$$\sum_i^m \alpha_i y_i {x_i}^T x_j + b = y_j, \forall j \in S$$

$$\Rightarrow b = \frac{1}{|S|}\sum_{j\in S} y_j - \sum_i^m \alpha_j y_i {x_i}^T x_j$$

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:

$$\mathcal{L}(w, b, \alpha, \xi, \mu) = \frac{1}{2}\|w\|^2 + C\sum_i^m \xi_i + \sum_i^m [\alpha_i(1 - \xi_i - y_i(w^T x_i + b)) - \mu_i \xi_i]$$

The primal and dual optimal solutions should satisfy KKT conditions:

Stationarity:

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_{i}^{m} \alpha_i y_i x_i$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \Rightarrow \sum_{i}^{m} \alpha_i y_i = 0$$

$$\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \Rightarrow \alpha_i = C - \mu_i, \forall i$$

Feasibility:

$$\alpha_i \geq 0, 1 - \xi_i - y_i(w^T x_i + b) \leq 0, \xi_i \geq 0, \mu_i \geq 0, \forall i$$

Complementary slackness:

$$\alpha_i(1 - \xi_i - y_i(w^T x_i + b)) = 0, \mu_i \xi_i = 0, \forall i$$

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

$$\max_{\alpha} \sum_{i}^{m} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j,$$

$$s.t. \sum_{i}^{m} \alpha_i y_i = 0,\ 0 \leq \alpha_i \leq C, \forall i$$

Also, b is given by: $M = \{i | 0 < \alpha_i < C\}$

$$y_j(w^T x_i + b) = 1, \forall j \in M$$

$$\Rightarrow\ y_j \left(\sum_{i}^{m} \alpha_i y_i {x_i}^T x_j + b \right) = 1, \forall j \in M$$

Since ${y_j}^2 = 1$,

$$\sum_{i}^{m} \alpha_i y_i {x_i}^T x_j + b = y_j, \forall j \in M$$

$$\Rightarrow b = \frac{1}{|M|} \sum_{j \in M} y_j - \sum_{i}^{m} \alpha_j y_i {x_i}^T x_j$$

For the slack variables' calculation, $\max(0, 1 - y_i(w^T x_i + b))$ can return the value of the slack variables, where y_i is the label {-1, +1}, $w^T x_i + b$ is the result calculated by the input vectors x_i and the estimated parameters w and b.

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 $w^T x_i + b$ 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:

```
----- SVM with slack variable C=0.3-----
training error: 0.05
testing_error: 0.13333333333333333
w_of_setosa: -0.08447998,0.44581505,-0.8401827,-0.39556637
b_of_setosa: 1.56536027
support_vector_indices_of_setosa: 73,78,13,14,31
slack_variable_of_setosa: 0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.02039173,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.00033244,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.00016572,0.0,0.0,0.0,0.0,0.0,0.0,0.2933789,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
w_of_versicolor: -0.27866492,-1.41,0.4881854,-0.75418015
b_of_versicolor: 4.26123751
support_vector_indices_of_versicolor: 2,3,15,19,20,24,27,28,31,32,35,37,81,82,83,84,86,88,89,
91,92,93,95,96,97,98,99,100,102,103,104,105,107,108,109,112,113,116,117,119,41,42,43,44,45,46,
47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79
slack_variable_of_versicolor: 0.0,0.0,0.0,0.30168746,0.2945643,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.26817355,0.0,0.0,0.0,0.0,0.06977302,0.18290653,0.0,0.0,0.0,0.0,0.18163951,0.0,0.0,0.0,0.18163951,0.28891688,
0.0,0.0,0.0,1.17263237,0.00691688,0.0,0.0,0.0,0.15085144,0.0,0.0,2.43e-06,0.0,0.0,0.0,0.0,0.0,1.69377703,0.31419053,1.2889993,
1.61125276,1.88465386,1.46372193,0.914639,0.50292088,0.74996111,1.90911983,1.41431106,0.75851321,
0.99716327,1.49245492,1.71578737,1.2942462,1.67699667,1.4341879,1.03868375,0.62991316,0.60331369,1.16311211,0.93469467,1.40798895,2.21460591,1.81
w_of_virginica: -0.12864428,-0.42231489,1.52634679,1.34681045
b_of_virginica: -7.68506986
support_vector_indices_of_virginica: 43,46,48,50,52,53,56,57,58,63,64,65,66,67,71,80,81,83,89,91,93,96,97,103,104,108,111,112,116,117,119
slack_variable_of_virginica: 0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
```

```
----- SVM with slack variable C=0.7-----
training error: 0.05
testing_error: 0.1
w_of_setosa: -0.04307697,0.48820506,-0.93871798,-0.43435899
b_of_setosa: 1.42161509
support_vector_indices_of_setosa: 78,13,31
slack_variable_of_setosa: 0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,8.1e-07,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,7e-07,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.12848638,0.0,0.0,0.0,0.0,0.0,0.0,
0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
w_of_versicolor: -0.39293791,-1.9218171,0.84620772,-1.47414042
b_of_versicolor: 6.06715976
support_vector_indices_of_versicolor: 2,3,15,19,20,24,27,28,31,35,37,81,
82,83,84,86,88,89,91,92,93,95,96,97,98,99,100,102,103,104,105,107,108,112,
116,117,119,41,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79
slack_variable_of_versicolor: 0.0,0.0,0.0,0.45288328,0.39548991,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.3961232,0.0,0.0,0.0,0.0,0.0,0.12964115,0.28252907,0.0,0.0,0.0,0.0,0.0,0.3
w_of_virginica: -0.34650663,-0.40290808,1.90192497,1.77461362
b_of_virginica: -8.97370076
support_vector_indices_of_virginica: 43,46,48,50,52,56,57,58,63,64,66,80,89,91,93,96,97,103,104,108,112,116,119
slack_variable_of_virginica: 0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
```

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:

$$\max_\alpha \sum_i^m \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

$$s.t. \sum_i^m \alpha_i y_i = 0, \alpha_i \geq 0, \forall i$$

The related kernel functions are:

Polynomial kernel:

$$k(x, x_i) = (1 + \frac{x^T x_i}{\sigma^2})^p, p > 0$$

Radical Basis Function (RBF) kernel:

$$k(x, x_i) = \exp\left\{-\frac{\|x - x_i\|^2}{2\sigma^2}\right\}$$

Sigmoidal kernel:

$$k(x, x_i) = \frac{1}{1 + exp^{-\frac{x^T x_i + b}{\sigma^2}}}$$

2. Data processing and results

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

```
training error: 0.025
testing_error: 0.0
b_of_setosa: 1.22094132
support_vector_indices_of_setosa: 78,13,31
b_of_versicolor: 4.33667006
support_vector_indices_of_versicolor: 14,31,89,93,96,97,99,103,108,48,50,52,57,58,63,64
b_of_virginica: -10.42876523
support_vector_indices_of_virginica: 50,52,57,63,96,97,99,103,108
```

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

```
training error: 0.008333333333333333
testing_error: 0.0
b_of_setosa: 1.13434963
support_vector_indices_of_setosa: 78,13,31
b_of_versicolor: 1.54426028
support_vector_indices_of_versicolor: 31,89,97,99,101,103,108,119,50,52,57,63,70
b_of_virginica: -6.11788922
support_vector_indices_of_virginica: 50,52,57,63,89,103,108,119
```

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

```
training error: 0.03333333333333333
testing_error: 0.03333333333333333
b_of_setosa: -0.33592638
support_vector_indices_of_setosa: 42,45,78,84,87,88,89,101,104,106,4,5,12,14,31
b_of_versicolor: -0.41536232
support_vector_indices_of_versicolor: 4,5,13,14,31,80,88,89,91,93,96,97,99,101,103,108,116,119,40,43,46,48,50,52,56,57,58,63,64,65,66,78
b_of_virginica: -0.30449079
support_vector_indices_of_virginica: 3,4,5,12,14,31,40,43,46,48,50,52,56,57,58,63,64,65,66,80,87,88,89,91,93,96,97,99,101,103,104,108,111,116,119
```

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