The Chinese University of Hong Kong, Shenzhen

DDA3020

Machine Learning

Assignment2 Report

Author:
Shi Wenlan

Student Number:
119010265

November 4, 2022

DDA3020: Homework II

October 23, 2022

Homework due: **23:59, November 08, 2022**. The exercise numbers refer to Kevin P. Murphy's book "Machine Learning: A Probabilistic Perspective". The total score of this assignment is 15.

1. Regularizing separate terms in 2d logistic regression **(Exercise 8.7 of Murphy's book)** (1 point)
(1) Consider the data in Figure 1, where we fit the model

$$p(y = 1 \mid \mathbf{x}, \mathbf{w}) = \sigma\left(w_0 + w_1 x_1 + w_2 x_2\right).$$

Suppose we fit the model by maximum likelihood, i.e.,

$$J(\mathbf{w}) = -\ell\left(\mathbf{w}, \mathcal{D}_{\text{train}}\right),$$

where $\ell\left(\mathbf{w}, \mathcal{D}_{\text{train}}\right)$ is the log likelihood on the training set. Sketch a possible decision boundary corresponding to $\hat{\mathbf{w}}$. (Copy the figure first (a rough sketch is enough), and then superimpose your answer on your copy, since you will need multiple versions of this figure). Is your answer (decision boundary) unique? How many classification errors does your method make on the training set?

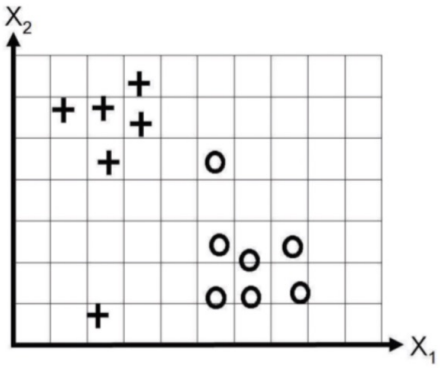


Figure 1: Data for logistic regression question

(2) Now suppose we regularize only the w_0 parameter, i.e., we minimize

$$J_0(\mathbf{w}) = -\ell\left(\mathbf{w}, \mathcal{D}_{\text{train}}\right) + \lambda w_0^2.$$

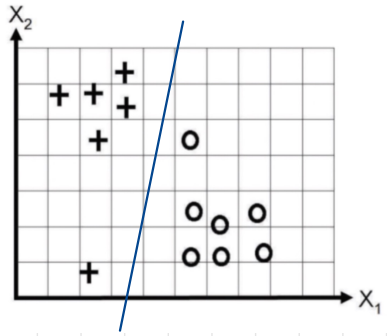
Suppose λ is a very large number, so we regularize w_0 all the way to 0, but all other parameters are unregularized. Sketch a possible decision boundary. How many classification errors does your method make on the training set? Hint: consider the behavior of simple linear regression, $w_0 + w_1 x_1 + w_2 x_2$ when $x_1 = x_2 = 0$.
(3) Now suppose we heavily regularize only the w_1 parameter, i.e., we minimize

$$J_1(\mathbf{w}) = -\ell\left(\mathbf{w}, \mathcal{D}_{\text{train}}\right) + \lambda w_1^2$$

Sketch a possible decision boundary. How many classification errors does your method make on the training set?
(4) Now suppose we heavily regularize only the w_2 parameter. Sketch a possible decision boundary. How many classification errors does your method make on the training set?

Solution:

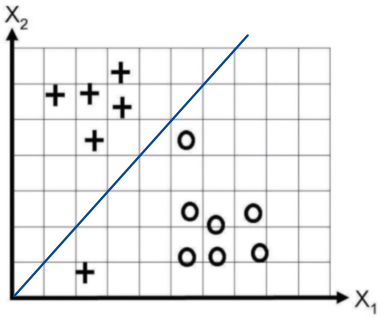
(1) A possible decision boundry is as following:



It is not unique.

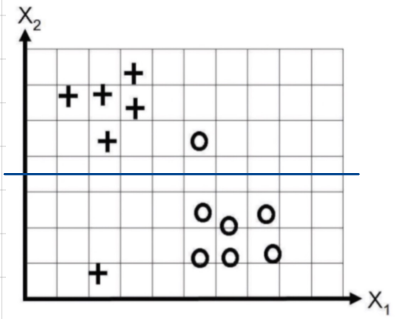
There is 0 classification error on the training set.

(2) A possible decision boundry is as following:



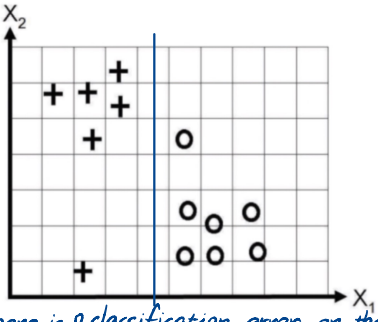
There is 1 classification error on the training set.

(3) A possible decision boundry is as following:



There are 2 classification errors on the training set.

(4) A possible decision boundry is as following:



There is 0 classification error on the training set.

2. Fitting an SVM classifier by hand **(Exercise 14.1 of Murphy's book)** (2 points)
Consider a dataset with 2 points in 1d: $(x_1 = 0, y_1 = -1)$ and $(x_2 = \sqrt{2}, y_2 = 1)$. Consider mapping each point to 3d using the feature vector $\phi(x) = [1, \sqrt{2}x, x^2]^\top$. (This is equivalent to using a second order polynomial kernel.) The max margin classifier has the form

$$\min \|\mathbf{w}\|^2 \quad \text{s.t.}$$
$$y_1\left(\mathbf{w}^T\phi\left(\mathbf{x}_1\right) + w_0\right) \geq 1$$
$$y_2\left(\mathbf{w}^T\phi\left(\mathbf{x}_2\right) + w_0\right) \geq 1$$

(1) Write down a vector that is parallel to the optimal vector $\mathbf{w}$.
(2) What is the value of the margin that is achieved by this $\mathbf{w}$? Hint: recall that the margin is the distance from each support vector to the decision boundary. Hint 2: think about the geometry of 2 points in space, with a line separating one from the other.
(3) Solve for $\mathbf{w}$, using the fact the margin is equal to $1/\|\mathbf{w}\|$.
(4) Solve for w_0 using your value for $\mathbf{w}$ and the optimization problem above. Hint: the points will be on the decision boundary, so the inequalities will be tight.
(5) Write down the form of the discriminant function $f(x) = w_0 + \mathbf{w}^\top\phi(x)$ as an explicit function of x.

Solution:

(1) Assume $w^T = [w_1\ w_2\ w_3]$

$$\begin{cases} -1\cdot(w_1 + w_0) \geq 1 \\ w_1 + 2w_2 + 2w_3 + w_0 \geq 1 \end{cases} \Rightarrow \begin{cases} w_0 + w_1 \leq -1 \\ w_0 + w_1 + 2w_2 + 2w_3 \geq 1 \end{cases}$$

$\because \min \|w\|^2 = w_1^2 + w_2^2 + w_3^2$

$\therefore w_0 = -1, w_1 = 0, w_2 = \frac{1}{2}, w_3 = \frac{1}{2}$

$w = [0\ \frac{1}{2}\ \frac{1}{2}]^T$

(2) $P_1: (1, 0, 0)$ $P_2: (1, 2, 2)$

$\text{margin} = \frac{1}{2}\sqrt{(1-1)^2 + (2-0)^2 + (2-0)^2} = \sqrt{2}$

(3) $\frac{1}{\|w\|} = \sqrt{2}$ $\therefore \|w\| = \frac{1}{\sqrt{2}}$ $\therefore w = [0\ \frac{1}{2}\ \frac{1}{2}]^T$

(4) $\begin{cases} w_0 + 0 \leq -1 \\ w_0 + 0 + 1 + 1 \geq 1 \end{cases}$ $\therefore w_0 = -1$

(5) $f(x) = -1 + \frac{\sqrt{2}}{2}x + \frac{1}{2}x^2$

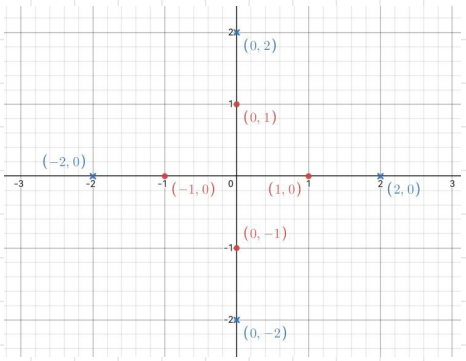
3. Given a binary data set: (2 points)

$$\text{Class -1:} \begin{bmatrix} (1 & 0) \\ (0 & 1) \\ (-1 & 0) \\ (0 & -1) \end{bmatrix} \qquad \text{Class } +1: \begin{bmatrix} (2 & 0) \\ (0 & 2) \\ (-2 & 0) \\ (0 & -2) \end{bmatrix}$$

(1) Can you find a svm classifier(without slack variable) for this data set? explain why; (1 point)
(2) Use SVM by expanding the original feature vector $\mathbf{x} = [x_1; x_2]$ to $\phi(\mathbf{x}) = [x_1^2; x_2^2]$, find the svm of this given data set and predict the label of $[1; 2]$. (1 point)

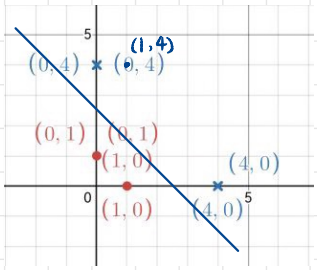
Solution:

(1)



No, because as the figure shows, it is not linearly separable.

(2) Class -1: $\begin{bmatrix} (1 & 0) \\ (0 & 1) \\ (1 & 0) \\ (0 & 1) \end{bmatrix}$ Class +1 $\begin{bmatrix} (4 & 0) \\ (0 & 4) \\ (4 & 0) \\ (0 & 4) \end{bmatrix}$



The SVM is $f(x) = x_1^2 + x_2^2 - \frac{5}{2}$

$\begin{cases} \text{class } +1 \text{ if } f(x) \geq 0 \\ \text{class } -1 \text{ if } f(x) < 0 \end{cases}$

$\because f([1\ 2]) = 1^2 + 2^2 - \frac{5}{2} = \frac{5}{2} > 0$

$\therefore [1, 2]$ is predicted to be class +1

4. Show that the value γ of the margin width for the maximum-margin hyperplane is given by

$$\frac{1}{\gamma^2} = \sum_{n=1}^{N} \alpha_n,$$

where $\{\alpha_n\}$ are given by the following optimization problem (2 points)

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & \sum_{n=1}^{N} \alpha_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} \alpha_n \alpha_m y_n y_m \mathbf{x}_n^\top \mathbf{x}_m \\ s.t. \quad & \sum_{n=1}^{N} \alpha_n y_n = 0 \\ & \alpha_n \geq 0 \quad \forall n = 1, 2, ..., N \end{aligned}$$

Hint: consider the minimum value of the Lagrange function.

Solution:

The objective function of support vector machine is: $\min_{w,b} \frac{1}{2}\|w\|^2$ s.t. $y_i(w^T x_i + b) \geq 1, \forall i$

It can be transformed to $\min_{w,b} \frac{1}{2}\|w\|^2$ s.t. $1 - y_i(w^T x_i + b) \leq 0, \forall i$

Its Lagrange function is $L(w,b,\alpha) = \frac{1}{2}\|w\|^2 + \sum_i^N \alpha_i (1 - y_i(w^T x_i + b))$ where α is dual variable.

According stationary condition: $\begin{cases} \frac{\partial L}{\partial w} = 0 \Rightarrow w = \sum_i^N \alpha_i y_i x_i \\ \frac{\partial L}{\partial b} = 0 \Rightarrow \sum_i^N \alpha_i y_i = 0 \end{cases}$

$$\begin{aligned} \Rightarrow L(w,b,\alpha) &= \frac{1}{2}\|w\|^2 + \sum_i^N \alpha_i - \sum_i^N \alpha_i y_i (\sum_j \alpha_j y_j x_j)^T x_i - \sum_i^N \alpha_i y_i b \\ &= \frac{1}{2}\|w\|^2 + \sum_i^N \alpha_i - \sum_{i,j}^N \alpha_i \alpha_j y_i y_j x_i^T x_j - b \sum_i^N \alpha_i y_i \\ &= \frac{1}{2}\|w\|^2 + \sum_i^N \alpha_i - \|w\|^2 - b \cdot 0 \\ &= \sum_i^N \alpha_i - \frac{1}{2}\|w\|^2 \\ &= \sum_i^N \alpha_i - \frac{1}{2} \sum_{i,j}^N \alpha_i \alpha_j y_i y_j x_i^T x_j \end{aligned}$$

According to feasibility condition: $\alpha_i \geq 0$

$\Rightarrow$ We get the dual problem presented in the exercise problem.

$\because$ The margin $\gamma = \frac{1}{\|w\|}$

$\therefore \frac{1}{\gamma^2} = \|w\|^2$

$\because$ According to Strong Duality Theorem $\max L(w,b,\alpha) = \min \frac{1}{2}\|w\|^2$ when getting optimal solution

$\therefore \sum_i^N \alpha_i - \frac{1}{2}\|w\|^2 = \frac{1}{2}\|w\|^2$

$\therefore \|w\|^2 = \sum_i^N \alpha_i$

$\therefore \frac{1}{\gamma^2} = \sum_{n=1}^N \alpha_n$

Programming

Task description In this problem you are asked to write a program that construct support vector machine models with different kernel functions and slack variable.

Datasets You are provided with the training and testing datatset (see *train.txt* and *test.txt*), including 120 training data and 30 testing data, respectively. It covers 3 classes, corresponding to setosa, versicolor, virginica. They are derived from the Iris dataset (`https://archive.ics.uci.edu/ml/datasets/iris`), contains 3 classes of 50 instances each, where each class refers to a type of iris plant. Your task is to classify each iris plant as one of the three possible types.

What you should do You should use the SVM function from python sklearn package, which provides different form of SVM function you can use. For multiclass SVM you should use one vs rest strategy. You are recommended to use sklearn.svm.svc() function. You can use numpy for the vector manipulation. For technical report you should state clearly the optimization problem you are solving, how did you derive it, the meaning of different values in the formulation, and some results suitable for presenting in the report (e.g. training error, testing error). The basic form of SVM is given and you don't need to derive this

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$
$$\text{s.t. } 1 - y_i\left(\mathbf{w}^\top \mathbf{x}_i + b\right) \leq 0, \forall i$$

Solution:

The problem to be solved is: construct a support vector machine model to predict the class of the iris based on 4 features

i.e. sepal length, sepal width, petal length, petal width (cm)

↓

SVM model

↓

class of iris (Setosa/Versicolour/Virginica)

$\min_{w_0,b_0} \frac{1}{2}\|w_0\|^2$ s.t.

Assume x_i is the feature data of the i-th entry, y_i is the class of the i-th entry.

and $x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{i3} \\ x_{i4} \end{bmatrix} = \begin{bmatrix} \text{sepal length of the i-th iris (cm)} \\ \text{sepal width of the i-th iris (cm)} \\ \text{petal length of the i-th iris (cm)} \\ \text{petal width of the i-th iris (cm)} \end{bmatrix}$ $y_i = \begin{cases} 0 & \text{if the i-th iris is Setosa} \\ 1 & \text{if the i-th iris is Versicolour} \\ 2 & \text{if the i-th iris is Virginica} \end{cases}$

One Vs. Rest Strategy: this needs 3 classifiers - clf0, clf1, clf2 to determine the final prediction result.

Notes: The "decision_function_shape = 'ovr'" doesn't take effect in the sklearn.svm.SVC() interface.

sklearn.svm.SVC() interface only uses One Vs. One Strategy.

So I would use sklearn.svm.SVC() interface as the binary classifier to implement One Vs Rest Strategy.

Training:

clf0: training feature: x_i in "train.txt"

training target: $y_{0i} = \begin{cases} -1 & \text{if the i-th iris is Setosa} \\ 1 & \text{if the i-th iris is Versicolour/Virginica} \end{cases}$ in "train.txt"

clf1: training feature: x_i in "train.txt"

training target: $y_{1i} = \begin{cases} -1 & \text{if the i-th iris is Versicolour} \\ 1 & \text{if the i-th iris is Setosa/Virginica} \end{cases}$ in "train.txt"

clf2: training feature: x_i in "train.txt"

training target: $y_{2i} = \begin{cases} -1 & \text{if the i-th iris is Virginica} \\ 1 & \text{if the i-th iris is Versicolour/Setosa} \end{cases}$ in "train.txt"

Prediction:

Given x_i: https://www.cs.colorado.edu/~mozer/Teaching/syllabi/6622/papers/Platt1999.pdf

$x_i \to$ clf0 $\to$ [The possibility that the i-th iris is Setosa P_0, $1-P_0$]

$x_i \to$ clf1 $\to$ [The possibility that the i-th iris is Versicolour P_1, $1-P_1$]

$x_i \to$ clf2 $\to$ [The possibility that the i-th iris is Virginica P_2, $1-P_2$]

The probabilities are calibrated using Platt scaling:

⇐ logistic regression on the SVM scores, fit by an additional cross-validation on the training data.

If $\max(P_0, P_1, P_2) = P_0$, then predict that the i-th iris is Setosa
If $\max(P_0, P_1, P_2) = P_1$, then predict that the i-th iris is Versicolour
If $\max(P_0, P_1, P_2) = P_2$, then predict that the i-th iris is Virginica
Check the prediction result and y_i to calculate prediction error.

1. (2 points) Calculate using standard SVM model (linear separator). Fit your algorithm on the training dataset, then validate your algorithm on testing dataset. Compute the misclassification error of training and testing datasets, the weight vector $\mathbf{w}$, the bias b, and the indices of support vectors(start with 0). Write output to file **SVM_linear.txt**. Note that the sklearn package doesn't provide a function with strict separation so we will simulate this using $C = 1e5$. You should print out the coefficient for each different class separately. The output format should be like this

```
${training_error}
${testing_error}
${w_of_setosa}
${b_of_setosa}
${support_vector_indices_of_setosa}
${w_of_versicolor}
${b_of_versicolor}
${support_vector_indices_of_versicolor}
${w_of_virginica}
${b_of_virginica}
${support_vector_indices_of_virginica}
```

Partial Output:

SVM_linear.txt

```
1  0.10833333333333334
2  0.13333333333333333
3  0.04625853808554219, -0.5211827995531008, 1.0
4  -1.4528444969775751
5  13, 31, 78
6  0.3676399048773078, 4.575262242928147, -1.403
7  -13.771682929874231
8  41, 44, 45, 46, 47, 50, 54, 55, 56, 57, 58, 5
9  4.202064068056643, 7.176284283883007, -8.7498
10 20.255859171538205
11 96, 99, 103, 108, 50, 52, 57, 63
```

where each line contains one variable. The training error and testing error count the total error instead of error for each distinct class, the error is $\frac{\text{wrong prediction}}{\text{number of data}}$. If we view the one vs all strategy as combining the multiple different SVM, each one being a separating hyperplane for one class and the rest of the points, then the w, b and support vector indices for that class is the corresponding parameters for the SVM separating this class and the rest of the points. If a variable is of vector form, say $\alpha = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$, then you should write each entry in the same line with comma separated, e.g.

```
1,2,3
```

You should also mention in your report on which classes are linear separable with SVM without slack and how you find it out.

(1) Optimization problem: Assume $w_0 = [w_{00}, w_{01}, w_{02}, w_{03}]^T$, $w_1 = [w_{10}, w_{11}, w_{12}, w_{13}]^T$, $w_2 = [w_{20}, w_{21}, w_{22}, w_{23}]^T$

clf0: $\max_{w_0, b_0} \min_i \frac{y_{0i}(w_0^T x_i + b_0)}{\|w_0\|}$

Consider a fixed scale such that $y_{i*}(w_0^T x_{i*} + b) = 1$ where x_{i*} is the support vector point closest to the hyperplane

Then for all data $y_{0i}(w_0^T x_i + b_0) \geq 1$

$\Rightarrow \min_{w_0, b_0} \frac{1}{2}\|w_0\|^2$

s.t. $y_{0i}(w_0^T x_i + b_0) \geq 1, \forall i$

Similarly:

clf1: $\min_{w_1, b_1} \frac{1}{2}\|w_1\|^2$

s.t. $y_i(w_1^T x_i + b_1) \geq 1, \forall i$

clf2: $\min_{w_2, b_2} \frac{1}{2}\|w_2\|^2$

s.t. $y_i(w_2^T x_i + b_2) \geq 1, \forall i$

How to get support vectors w, b: Take clf0 as an example. (clf1 and clf2 are similar)

The dual problem: (See Written Problem 4 for derivation)

$$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_{0i} y_{0j} x_i^T x_j$$
$$\text{s.t. } \sum_i \alpha_i y_{0i} = 0,\ \alpha_i \geq 0\ \forall i$$

the primal solution $w_0 = \sum_i \alpha_i y_{0i} x_i$

$$b_0 = \frac{1}{|S|}\sum_{j \in S}\left(y_{0j} - \sum_i \alpha_i y_{0i} x_i^T x_j^T\right) \text{ where } S = \{i \mid \alpha_i > 0\}$$

It can be solved by any off-the-shelf optimization solver.

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

Thus the data points with $\alpha_i > 0$ are support vectors.

```
Predict: 2, True: 1
Predict: 2, True: 1
Predict: 2, True: 1
Predict: 2, True: 1
Predict: 2, True: 1
Predict: 2, True: 1
Predict: 2, True: 1
Predict: 2, True: 1
Predict: 2, True: 1
Predict: 2, True: 1
Predict: 1, True: 2
Predict: 1, True: 2
Predict: 1, True: 2
Predict: 2, True: 1
Predict: 2, True: 1
Predict: 2, True: 1
Predict: 2, True: 1
```

By analyzing the data with wrong prediction,
Class 0 (i.e. Setosa) is linear separable with other two classes.
Because class Setosa is never confused with the other two classes.

2. (3 points) Calculate using SVM with slack variables. For each $C = 0.1 \times t, t = 1, 2, ..., 10$, fit your algorithm on the training dataset, then validate your algorithm on testing dataset. Compute the misclassification error of training and testing datasets, the weight vector $\mathbf{w}$, the bias b, the indices of support vectors, and the slack variable $\boldsymbol{\xi}$. Write output to file **SVM_slack.txt**. The format is

```
${training_error}
${testing_error}
${w_of_setosa}
${b_of_setosa}
${support_vector_indices_of_setosa}
${slack_variable_of_setosa}
${w_of_versicolor}
${b_of_versicolor}
${support_vector_indices_of_versicolor}
${slack_variable_of_versicolor}
${w_of_virginica}
${b_of_virginica}
${support_vector_indices_of_virginica}
${slack_variable_of_virginica}
```

(2) Optimization problem:

clf0: $\min_{w_0, b_0, \xi_0} \frac{1}{2}\|w_0\|^2 + C\sum_i \xi_{0i}$
$\text{s.t. } 1 - \xi_{0i} - y_{0i}(w_0^T x_i + b_0) \leq 0,\ -\xi_{0i} \leq 0, \forall i$

clf1: $\min_{w_1, b_1, \xi_1} \frac{1}{2}\|w_1\|^2 + C\sum_i \xi_{1i}$
$\text{s.t. } 1 - \xi_{1i} - y_{1i}(w_1^T x_i + b_1) \leq 0,\ -\xi_{1i} \leq 0, \forall i$

clf2: $\min_{w_2, b_2, \xi_2} \frac{1}{2}\|w_2\|^2 + C\sum_i \xi_{2i}$
$\text{s.t. } 1 - \xi_{2i} - y_{2i}(w_2^T x_i + b_2) \leq 0,\ -\xi_{2i} \leq 0, \forall i$

where ξ_0, ξ_1, ξ_2 are slack variables, and C is the penalty parameter.

How to get support vectors w, b: Take clf0 as an example. (clf1 and clf2 are similar)

Its Lagrange function is $\mathcal{L}(w_0, b_0, \xi_0, \alpha, \mu) = \frac{1}{2}\|w_0\|^2 + C\sum_i \xi_{0i} + \sum_i [\alpha_i(1 - \xi_i - y_i(w_0^T x_i + b_0)) + \mu_i(-\xi_{0i})]$ and $\alpha_i, \mu_i \geq 0, \forall i$

According to stationary condition: $\frac{\partial \mathcal{L}}{\partial w_0} = 0 \Rightarrow w_0 = \sum_i \alpha_i y_{0i} x_i$

$$\frac{\partial \mathcal{L}}{\partial b_0} = 0 \Rightarrow \sum_i \alpha_i y_{0i} = 0$$

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

Thus $\mathcal{L}(\alpha,\mu) = \frac{1}{2}\|w_0\|^2 + \sum_i[\alpha_i(1-y_{0i}(w_0^T x_i + b_0))] + \sum_i \underbrace{(C-\alpha_i-\mu_i)}_{=0}\xi_{0i}$

$= \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_{0i}y_{0j}x_i x_j$

The dual problem is: $\max_{\alpha} \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_{0i}y_{0j}x_i x_j$ s.t. $\sum_i \alpha_i y_{0i} = 0,\ 0 \le \alpha_i \le C,\ \forall i$

the primal solution $w_0 = \sum_i \alpha_i y_{0i} x_i$

$b_0 = \frac{1}{|M|}\sum_{j\in M}(y_{0j} - \sum_i \alpha_i y_{0i} x_i^T x_j^T)$ where $M = \{i \mid 0 < \alpha_i < C\}$

It can be solved by any off-the-shelf optimization solver.

According to Complementary slackness $\alpha_i(1-\xi_{0i} - y_i(w^T x_i + b)) = 0,\ \mu_i \xi_{0i} = 0,\ \forall i$

If $0 < \alpha_i < C$, then $\mu_i = C - \alpha_i > 0$, then $\xi_{0i} = 0$

Thus the data points with $0 < \alpha_i < C$ are support vectors.

How to get slack variables: Take clf0 as an example. (clf1 and clf2 are similar)

$1 - \xi_{0i} - y_{0i}(w_0^T x_i + b_0) \le 0$

$\xi_{0i} \ge 1 - y_{0i}(w_0^T x_i + b_0)$ and $\xi_{0i} \ge 0$

$\therefore$ hyperplane -1: $1 - (-1)(w_0^T x_i + b_0) = 0 \Rightarrow$ distance $= \frac{1 + (w_0^T x_i + b_0)}{\|w_0\|} \Rightarrow$ slack $= \max(0, \text{distance})$

$1 + (w_0^T x_i + b_0) = 0$

hyperplane 1: $1 - (w_0^T x_i + b_0) = 0 \Rightarrow$ distance $= \frac{1 - (w_0^T x_i + b_0)}{\|w_0\|} \Rightarrow$ slack $= \max(0, \text{distance})$

Partial result:

```
C = 0.1
0.041666666666666664
0.13333333333333333
0.1470921538569829, -0.3508769716586302, 0.7189940004257297, 0.3420055299780926
-1.8921299070140811
10, 13, 14, 15, 31, 34, 40, 44, 59, 73, 78
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.10588867650006369, 0.06280481685787505, 0.0107
0.15567372696595771, 0.5856476364282258, -0.1890930005328131, 0.26784990875184467
-1.432728294508877
40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
0, 0, 0.24165075773201944, 0.2722879824254615, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.21247
-0.09194028495244216, 0.08716418394555608, -1.1056716321088855, -0.899104468897998
7.2229261139718295
80, 81, 83, 84, 86, 89, 91, 93, 96, 97, 103, 104, 107, 108, 109, 111, 112, 115, 116, 11
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
C = 0.2
0.03333333333333333
0.1
0.15469538060468466, -0.3915752519694864, 0.7651819374495733, 0.3543591751898917
-1.8964214397668673
13, 14, 31, 34, 59, 73, 78
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.08309204734597578, 0.04188582822600951, 0, 0,
0.31134745426973054, 1.1712952727853336, -0.3781860004788849, 0.5356998178770692
-3.8654565932822953
40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
0, 0, 0.24165075918166684, 0.27228798412542216, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.2124
0.07164491211993343, 0.23999999999999905, -1.3760313444397667, -1.1621932161599176
7.4575903494724525
80, 81, 83, 89, 91, 93, 96, 97, 103, 104, 107, 108, 111, 112, 116, 117, 119, 43, 45, 46
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

```
C = 1.0
0.058333333333333334
0.06666666666666667
0.04625853808554219, -0.5211827995531008, 1.0030446153124943, 0.46412978496693447
-1.4528444969775751
13, 31, 78
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.00033655614303910547, 0, 0, 0, 0, 0, 0, 0, 0,
0.41222947723850645, 1.9282728817251718, -0.9914934898325258, 1.7806923790505798
-6.007068738775105
41, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63,
0, 0, 0.15994106575529515, 0.1277344855170574, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.13800
0.3425109257207335, 0.3395682441796591, -2.0522147134948265, -2.222362477356284
10.631612459667224
80, 89, 91, 93, 96, 97, 103, 104, 108, 116, 119, 43, 46, 48, 50, 52, 56, 57, 63, 64, 66
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

It shows that reasonable slackness can improve prediction accuracy.

3. (3 points) Implement SVM with kernel functions and slack variables. You should experiment with different kernel functions in this task:

   (a) A 2nd-order polynomial kernel, write output to **SVM_poly2.txt**

   (b) A 3rd-order polynomial kernel, write output to **SVM_poly3.txt**

   (c) Radial Basis Function kernel with $\sigma = 1$, write output to **SVM_rbf.txt**

   (d) Sigmoidal kernel with $\sigma = 1$, write output to **SVM_sigmoid.txt**

   During these tasks we set $C = 1$. The output format is

```
${training_error}
${testing_error}
${b_of_setosa}
${support_vector_indices_of_setosa}
${b_of_versicolor}
${support_vector_indices_of_versicolor}
${b_of_virginica}
${support_vector_indices_of_virginica}
```

(3) The dual problem of SVM with kernel function $k(x_i, x_j)$ and slack variables:

clf0: $\max_{\alpha_0} \sum_i \alpha_{0i} - \frac{1}{2}\sum_{i,j} \alpha_{0i}\alpha_{0j} y_{0i} y_{0j} k(x_i, x_j)$ s.t. $\sum_i \alpha_{0i} y_{0i} = 0,\ 0 \le \alpha_{0i} \le C,\ \forall i$

clf1: $\max_{\alpha_1} \sum_i \alpha_{1i} - \frac{1}{2}\sum_{i,j} \alpha_{1i}\alpha_{1j} y_{0i} y_{0j} k(x_i, x_j)$ s.t. $\sum_i \alpha_{1i} y_{0i} = 0,\ 0 \le \alpha_{1i} \le C,\ \forall i$

clf2: $\max_{\alpha_2} \sum_i \alpha_{2i} - \frac{1}{2}\sum_{i,j} \alpha_{2i}\alpha_{2j} y_{0i} y_{0j} k(x_i, x_j)$ s.t. $\sum_i \alpha_{2i} y_{0i} = 0,\ 0 \le \alpha_{2i} \le C,\ \forall i$

Default: $C=1$, degree $=3$, gamma $=1$, coef0 $=0$

2-nd order polynomial kernel: $k(x_i, x_j) = (1 + x_i^T x_j)^2$

3-rd order polynomial kernel: $k(x_i, x_j) = (1 + x_i^T x_j)^3$

Radial Basis Function kernel: $k(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2}\right)$

Sigmoidal kernal: $k(x_i, x_j) = \frac{1}{1 + \exp^{-x_i^T x_j}}$

How to get b, support vectors:

$b_0 = \frac{1}{|M_0|}\sum_{j \in M_0}\left(y_{0j} - \sum_i \alpha_{0i} y_{0i} k(x_i, x_j)\right)$ where $M_0 = \{i \mid 0 < \alpha_{0i} < C\}$, data points with $0 < \alpha_{0i} < C$ are support vectors.

$b_1 = \frac{1}{|M_1|}\sum_{j \in M_1}\left(y_{1j} - \sum_i \alpha_{1i} y_{1i} k(x_i, x_j)\right)$ where $M_1 = \{i \mid 0 < \alpha_{1i} < C\}$, data points with $0 < \alpha_{1i} < C$ are support vectors.

$b_2 = \frac{1}{|M_2|}\sum_{j \in M_2}\left(y_{2j} - \sum_i \alpha_{2i} y_{2i} k(x_i, x_j)\right)$ where $M_2 = \{i \mid 0 < \alpha_{2i} < C\}$, data points with $0 < \alpha_{2i} < C$ are support vectors.

Partial result:

SVM_poly2.txt
```
0.03333333333333333
0.0
-1.223190399423636
13, 31, 78
-4.331066745260715
48, 50, 52, 57, 58, 63, 64, 14, 31, 89, 93, 96, 97, 99, 103, 108
10.42386320816419
96, 97, 99, 103, 108, 50, 52, 57, 63
```

SVM_poly3.txt
```
0.008333333333333333
0.0
-1.1357583700277185
13, 31, 78
-1.5439222910710315
50, 52, 57, 63, 70, 31, 89, 97, 99, 101, 103, 108, 119
6.131746230662894
89, 103, 108, 119, 50, 52, 57, 63
```

SVM_rbf.txt
```
0.025
0.03333333333333333
0.3877218948889365
3, 4, 5, 10, 12, 14, 31, 34, 40, 42, 44, 45, 48, 64, 65, 78, 84, 87, 88, 89, 99, 101, 1
0.3818508438577119
40, 42, 43, 46, 48, 50, 52, 56, 57, 63, 64, 65, 66, 78, 3, 4, 5, 10, 12, 13, 14, 31, 34
0.27499976969671747
80, 87, 88, 89, 91, 93, 96, 97, 99, 101, 103, 104, 106, 108, 111, 116, 119, 3, 4, 5, 10
```

SVM_sigmoid.txt
```
0.0
0.0
1.0
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 2
1.0
40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61,
1.0
80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 10
```

The Sigmoid version shows the best performance.