CSE353 Assignment7 Report

Chengzhi Dong
ID 112890166

Due December 16, 2021

1 Introduction

For Assignment 7, I need to implement and apply the linear support vector machine onto the given dataset to obtain the (w, b). After I found the (w, b), I need to identify which training samples in the dataset are support vectors, and compute the largest margin my SVM achieved. Lastly, I need to visualize the training data, highlight the support vectors, plot the decision boundary line and the two lines along the support vectors.

2 Method

My algorithm which coded in Python implement and apply the linear support vector machine onto the given dataset to obtain the (w, b). The python libraries my code imported are numpy, matplotlib.pyplot, qpsolvers, and quadprog. Install quadprog packets using "!pip install quadprog" and "!pip install qpsolvers". I used the solve_qp() function from qpsolvers in Python to solve the Quadratic Programming optimization problem. I load the data from the text files ('X_LinearSeparable.txt' and 'Y_LinearSeparable.txt') into arrays.

For part A, I defined H, ϕ, A, c for $\min\limits_{q} \frac{1}{2} q^T H q + \phi^T q$ s.t. $Aq \leq c$.

$$H = \begin{bmatrix} 0.00001 & \vec{0}_{1\times d} \\ \vec{0}_{d\times 1} & I_{d\times d} \end{bmatrix} \qquad \phi = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}_{(d+1)\times 1}$$

$$A = \begin{bmatrix} \vdots \\ -y_n, -y_n x_n^T \\ \vdots \end{bmatrix}_{N\times(d+1)} \qquad c = \begin{bmatrix} -1 \\ \vdots \\ -1 \end{bmatrix}_{N\times 1}$$

Note: The $H_{0,0}$ should be 0; however the quadprog() function I used required matrix H to be positive-definite; there I set $H_{0,0} = 0.00001$ to mostly ignore b when $q^T Hq$.
Then, I called $solve_q p(H, Phi, A, c)$ to solve the Quadratic Programming optimization problem and obtain q where $q = \begin{bmatrix} b \\ w_1 \\ \vdots \\ w_d \end{bmatrix}$. By splitting q, I obtained b and $\vec{w}$.

For part B, I defined a function called "getSupportVector()" which finds the support vectors and return them as a list. The function takes W, B, X, Y as input. The function iterates the x and y data, and determine whether x_n is a support vector by checking "$y_n \times (W^T x_n + B) - 1 < 0.00001(errorBound)$. If x_n satisfy this condition, then the function will append x_n into the support vector list. At the end, the function returns the support vector list. In part B, I identify the support vectors by calling "$getSupportVector(w, b, x\_data, y\_data)$.

For part C, I computed the largest margin I achieved from my SVM by $Margin(w, b) \triangleq \frac{1}{||w||_2}$

For part D, I defined a function called "Visualization()" to visualize the training data, highlight the support vectors, plot the decision boundary line and the two lines along the support vectors. The function takes $W, B, X, Y, title$ as inputs. The function loop through the data and created positive, negative, support vector arrays for plotting. It also find the minimum and maximum x values and used "$w^T x + b = 0$", "$w^T x + b = 1$", and "$w^T x + b = -1$" to draw the line of decision boundary and the two lines along the support vectors.

3 Experiment

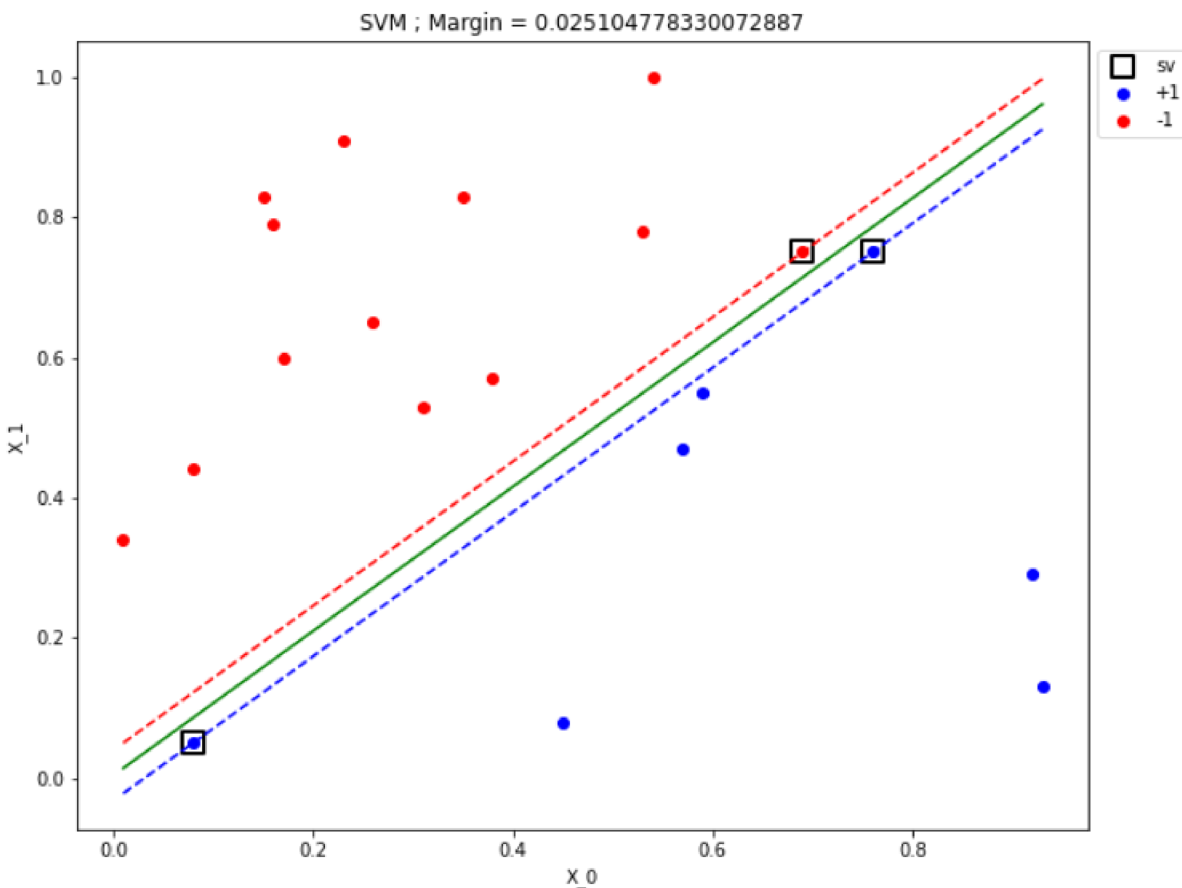
Below are my results:

```
Part A: w = [ 28.57142857 -27.75510204]; b = 0.10204081632652828

Part B: Support Vectors: [[0.76, 0.75], [0.08, 0.05], [0.69, 0.75]]

Part C: Margin = 0.025104778330072887

Part D:
```



As the results shown, I got $w = [28.5714, -27.7551]$ and $b = 0.1020$.
The support vectors are $[0.76, 0.75]$, $[0.08, 0.05]$, and $[0.69, 0.75]$.
The largest margin I achieved from my SVM is 0.0251.
For the visualization, the data samples with label "+1" are plotted as blue dots, and the data samples with label "-1" are plotted as red dots. The line of decision boundary is the green solid line. The line along the -1 support vector is the red dashed line. The line along the $+1$ support vectors is the blue dashed line.
Based on the visualization, my SVM did a great job on classifying the data with

a largest margin.

4 Discussion

When calling the $solve\_qp(H, \phi, A, c)$ function, I keep getting an error message saying matrix H is not positive definite. The first row and column of H is originally all zeros which makes it not positive definite. To fix this error, I change the value of $H_{0,0}$ from 0 to 0.00001 to make H positive definite and do its best to ignore b in $q^T Hq$. The result seems to be correct so I think the solution works.

When identifying the support vector, the $y_n \times (w^T x_n + b)$ is not exactly equal to 1 for the support vectors; therefore, I set a error bound of 0.00001 and as long as $(y_n \times (w^T x_n + b)) - 1$ is approximately zero within the error bound, x_n is a support vector.