OPTIMIZATION READING GROUP

Summer School on Optimization

LINEAR SUPPORT VECTOR MACHINES

Group B: Nguyen Tien Hung

Tran Dinh Dai Quan

Tran Hoang Phi

Supervisor: Prof. Tran Thai An Nghia

September - 2021

Project 3.2 - Linear support vector machines

September 19, 2021

Abstract

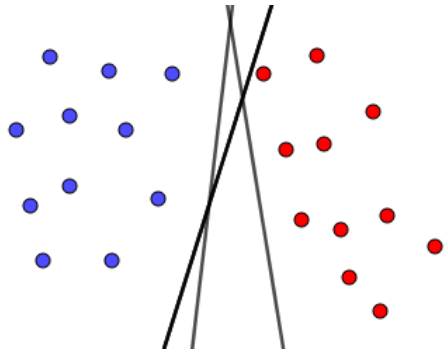
We train linear support vector machine models to solve two binary classification problems with two real data sets: leukemia and colon-cancer. The subgradient method will be studied and then applied to solve the above problems while their dual problems will be also solved. To demonstrate and analyze the result, we use a solver of Python and compare all results obtained from the above methods.

Keywords: Classification · programming · machine learning · support vector machines · subgradient method · dual problem.

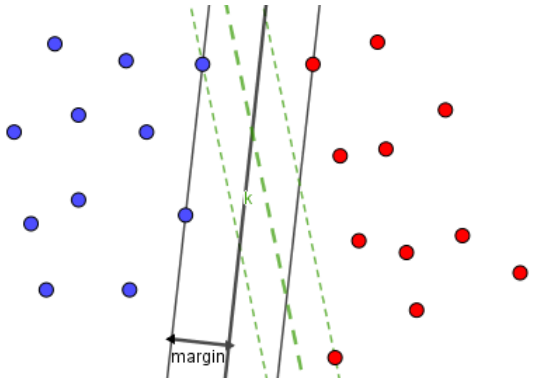
1 Introduction

1.1 Two-class division problem

Assuming that there are two different classes described by points in a multi-dimensional space. These two classes are linearly separable, which means there exists a hyperplane that divides exactly those two classes. Our mission is to find a hyperplane that divides those two classes, i.e. all points of a class lie on the same side of that hyperplane and opposite to all points of the other class.



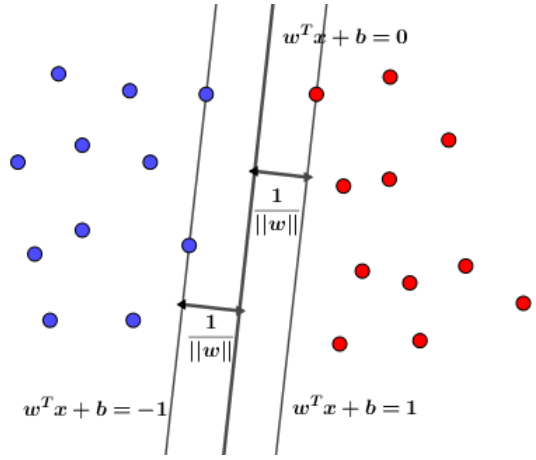
The question is: out of the multitude of hyperplanes, which one is the best by some standard? As you can see in the picture above, there are two lines that are quite skewed towards the red circle class. This can make the red class unhappy because the territory seems to be encroached too much. Is there a way to find the dividing line that separates two classes best?



We need a dividing line such that the distance from the nearest point of each class to the dividing line is the same. This equal distance is called the margin. We continue to consider the picture above when the distance from the dividing line to the nearest points of each class is the same. Considering two ways of dividing, dividing by solid black line and green dashed line, which line is the better line? Obviously it should be the solid black line because it creates a wider margin. Wider margins are better because the division between the two classes is clearer.
The optimization problem in **Support Vector Machine (SVM)** is the problem of finding the hyperplane so that the margin is the largest.

1.2 Construct an optimization problem for SVM

Assume that the data pairs of the training set are $(\mathbf{x_1}, y_1)$, $(\mathbf{x_2}, y_2), \ldots, (\mathbf{x_N}, y_N)$. With $\mathbf{x_i} \in \mathbb{R}^d$ represents the input of a data point and y_i is the label of that data point. d is the dimension of the data and N is the number of data points. Assume that the label of each data point is defined by $y_i = 1$ (class 1) or $y_i = -1$ (class 2).



Assume that the blue points belong to class 1, red points belong to class -1 and the hyperplane $\mathbf{w^T x} + \mathbf{b} = 0$ is the hyperplane that separates two classes. Furthermore, class 1 is on the positive side, class -1 is on the negative side of the hyperplane. We need to find the coefficients $\mathbf{w}$ and $\mathbf{b}$.
By choosing the appropriate w and b, we can assume $\mathbf{w^T x^i} + b = 1$ or $\mathbf{w^T x^i} + b = -1$ with the points which are closest to the hyperplane as in the picture above. So we can suppose that:

$$w^T x_i + b \leq -1 \text{ if } x_i \text{ belongs to blue.}$$

$$w^T x_i + b \geq 1 \text{ if } x_i \text{ belongs to red.}$$

Use a decision function namely:

$$y_i = -1 \text{if } x_i \text{ belongs to blue} 1 \text{if } x_i \text{ belongs to red.}$$

We have the constraints: $y_i(w^T x_i + b) \geq 1$.

We can easily prove that the distances between the hyperplane is $\frac{1}{||w||}$. We want to have a wider margin, so we have the problem:

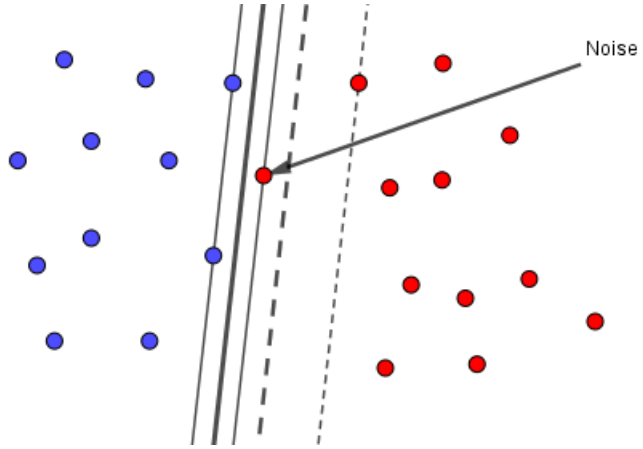
$$\max_{w,b} \frac{2}{||w||} \text{ subject to } y_i(w^T x_i + b) \geq 1.$$

This problem is "equivalent" to

$$\min_{w,b} \frac{||w||^2}{2} \text{ subject to } 1 - y_i(w^T x_i + b) \leq 0.$$

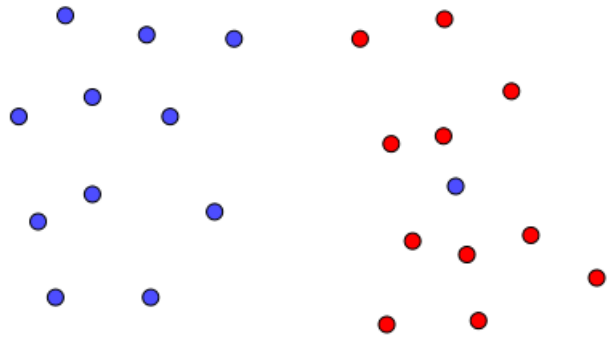
1.3 Soft margin SVM

Consider the example in picture below:



We can see that SVM is not working efficiently. The data set is still linearly separable but there is a noise point of the red class that is too close to the blue class. In this case, if we use pure SVM, it will create a very small margin. In addition, the classification hyperplane is located too close to the blue class and far from the red class. Whereas, if we sacrifice this noise, we will get a much better margin depicted by the dashed lines. The margin created by the dividing line and the dashed line is also known as the soft margin. Also in this way, pure SVM is also known as Hard Margin SVM.

We continue to consider this situation:



We can see that the data is not linearly separable but near linearly separable. In this case, if we use pure SVM, then obviously the optimization problem is infeasible, because feasible set is an empty set. Therefore, the SVM optimization problem becomes unsolved. However, if we are willing to sacrifice the blue point in the region of the red points, we can still find a good hyperplane. This problem then become the Soft Margin SVM.
The optimization problem for Soft Margin SVM has many different approaches. One of them is to construct an unconstrained optimization problem. For a larger margin in Soft Margin SVM, we need to sacrifice a few data points by allowing them to fall into the "unsafe zone".
Let η_i such that $1 - y_i(w^T x_i + b) \leq \eta_i$ and $\eta_i \geq 0$. So $\eta_i \geq \max\left\{0, 1 - y_i(w^T x_i + b)\right\}$.
Of course, we must limit this sacrifice, otherwise we can create a very large margin by sacrificing most of the points in the data set. So the objective function should be a combination of maximizing margin and minimizing sacrifice and then we have the problem:

$$\min_{w,b} \frac{||w||^2}{2} + C\sum_{i=1}^{m} \eta_i \quad \text{s.t} \quad 1 - y_i(w^T x_i + b) \leq \eta_i 0 \leq \eta_i.$$

with C is a positive constant.
Without loss of generality, we can set $\eta_i = \max\left\{0, 1 - y_i(w^T x_i + b)\right\}$ to have the problem:

$$\min_{w,b} \frac{||w||^2}{2} + C\sum_{i=1}^{m} \max\left\{0, 1 - y_i(w^T x_i + b)\right\}.$$

The question here is: how to solve the above problem? We consider the task below.

2 Task

You consider a binary classification problem, *i.e.*, $y_i \in \{-1, 1\}$. The classification rule of linear SVM is defined by a decision function $h_{w,b}(x^i) = w^T x^i + b$, namely

$$\hat{y} = \begin{cases} 1 & \text{if } h_{w,b}(x) \geq 0, \\ -1 & \text{otherwise.} \end{cases} \tag{1}$$

Training a linear SVM model means finding the value of w and b that makes the margin as wide as possible while avoiding margin violations (hard margin) or limiting them (soft margin):

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \max(0, 1 - y_i(w^T x^i + b)) + R(w), \tag{2}$$

where C is a hyperparameter.

Task:

- Use a solver to train (2) with $R = 0$ and $R = \alpha\|\cdot\|_1$, and test on two data sets: leukemia and colon-cancer.
- Use the subgradient method to solve this problem.
- Write and solve the dual problem of (2) with $R = 0$, and evaluate on the data sets.

The data sets can be download from `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`.

3 Basic Definition and Preliminaries

Definition 3.1 (Subdifferential of a convex function)**.** Let $f : \mathbb{E} \to \overline{\mathbb{R}}$ be a convex function and $\bar{x} \in \text{dom} f$. A vector $v \in \mathbb{E}^*$ is called a subgradient of f at $\bar{x}$ if

$$f(x) \geq f(\bar{x}) + \langle v, x - \bar{x} \rangle \,, \text{ for all } x \in \mathbb{E}.$$

The subdifferential of f at $\bar{x}$ is defined by

$$\partial f(\bar{x}) := \{v \in \mathbb{E}^* | f(x) - f(\bar{x}) \geq \langle v, x - \bar{x} \rangle\}.$$

Proposition 3.1. Let $f : \mathbb{E} \to \overline{\mathbb{R}}$ be a convex function that is differentiable at $\bar{x}$. Then $\partial f(\bar{x}) = \nabla f(\bar{x})$.

Theorem 3.1 (Sum rule I)**.** Let $f, g : \mathbb{E} \to \overline{\mathbb{R}}$ be proper convex functions. Suppose that f is differentiable at $\bar{x} \in \text{dom } g$. Then we have

$$\partial(f + g)(\bar{x}) = \nabla f(\bar{x}) + \partial g(\bar{x}).$$

Theorem 3.2 (Sum rule II)**.** Let $f, g : \mathbb{E} \to \overline{\mathbb{R}}$ be proper convex functions. Suppose that $\text{dom } f \cap \text{int } (\text{dom } g) \neq \emptyset$. Then we have the sum rule

$$\partial(f + g)(\bar{x}) = \partial f(\bar{x}) + \partial g(\bar{x}), \text{ for any } x \in \text{dom } f \cap \text{dom } g.$$

4 Subdifferential of some special functions

In this section we will find the subdifferential of some functions with special properties and this will help in solving our main problem.

4.1 Max function

Consider $f(x) = \alpha \max\{f_1(x), f_2(x)\}, \alpha \neq 0$, for $f_1, f_2 : \mathbb{R}^n \to \overline{\mathbb{R}}$ are convex, differentiable functions. Take $x \in \mathbb{R}^n$, consider 3 cases:

- $f_1(x) > f_2(x)$. Then $f(x) = \alpha \max\{f_1(x), f_2(x)\} = \alpha f_1(x)$. We have f_1 is convex and differentiable, so
$$\partial f(x) = \partial \alpha f_1(x) = \nabla \alpha f_1(x).$$
- $f_1(x) < f_2(x)$. Then $f(x) = \alpha \max\{f_1(x), f_2(x)\} = \alpha f_2(x)$. Similarly, we have
$$\partial f(x) = \partial \alpha f_2(x) = \nabla \alpha f_2(x).$$
- $f_1(x) = f_2(x)$. Apply the max rule, we have
$$\partial f(x) = [\nabla \alpha f_1(x), \nabla \alpha f_2(x)] \,.$$

Thus

$$\partial f(x) = \begin{cases} \nabla \alpha f_1(x), & \text{if } f_1(x) > f_2(x), \\ [\nabla \alpha f_1(x), \nabla \alpha f_2(x)] \,, & \text{if } f_1(x) = f_2(x), \\ \nabla \alpha f_2(x), & \text{if } f_1(x) < f_2(x). \end{cases} \tag{3}$$

With $[x, y] := \{\lambda x + (1 - \lambda) y | \lambda \in [0, 1]\}$.

4.2 ℓ_1-norm function

Consider $f : \mathbb{R}^n \to \mathbb{R}, f(x) = k \|x\|_1 (k \neq 0)$.
For any $v \in \partial f(x)$, we have

$$f(y) - f(x) \geq \langle v, y - x \rangle, \forall y \in \mathbb{R}^n.$$

Which means

$$\begin{aligned} k\|y\|_1 - k\|x\|_1 \geq \langle v, y - x \rangle, \forall y \in \mathbb{R}^n &\Leftrightarrow k\sum_{i=1}^{n} y_i - k\sum_{i=1}^{n} x_i \geq k\sum_{i=1}^{n} v_i(y_i - x_i), \forall y \in \mathbb{R}^n \\ &\Leftrightarrow k|y_i| - k|x_i| \geq v_i(y_i - x_i), \forall y_i \in \mathbb{R}. \end{aligned}$$

- If $x_i = 0$, then $k|y_i| \geq v_i y_i \Leftrightarrow -k \leq v_i \leq k$.
- If $x_i > 0$, we have $k(|y_i| - x_i) \geq v_i(y_i - x_i), \forall y_i \in \mathbb{R}$. Choose $y_i = 0$ and $y_i = x_i + 1$, we have $v_i \geq k$ and $v_i \leq k$. So $v_i = k$.
- If $x_i < 0$, we have $k(|y_i| + x_i) \geq v_i(y_i - x_i), \forall y_i \in \mathbb{R}$. Choose $y_i = 0$ and $y_i = x_i - 1$, we have $v_i \leq k$ and $v_i \geq k$. So $v_i = -k$.

Thus

$$v_i = \begin{cases} k.\text{sign}(x_i), & \text{if } x_i \neq 0 \\ [-k, k], & \text{if } x_i = 0 \end{cases}. \tag{4}$$

5 Subgradient method

The subgradient method is a very simple algorithm aimed at minimizing the objective function f that is a nondifferentiable convex funtion. This algorithm is very similar to the gradient method for a differentiable function f. Therefore, to avoid confusion between the two algorithms, we give the notes of the subgradient method to distinguish it from the gradient method. As follows:

- As mentioned above, the subgradient method is applied for *nondifferentiable* f.
- Choosing the step size (shown later).
- The subgradient method *is not a descent method*, *i.e.*, the function value can increase and this often happens in many cases. Hence we need a solution (without in gradient method) to overcome this.

5.1 Negative subgradient update

Suppose $f : \mathbb{R}^n \to \mathbb{R}$ is convex with domain $\mathbb{R}^n$. The subgradient method for minimizing f described as follows:

Initialization: pick $x^0 \in \mathbb{R}^n$ arbitrarily.
General step: for any $k = 0, 1, 2, \ldots$ execute the following steps:

(a) pick a step size $t_k > 0$ and a subgradient $g^k \in \partial f(x^k)$;

(b) set $x^{k+1} = x^k - t_k g^k$.

However, the subgradient method don't guarantee that the sequence of function values is monotonous. For that reason, we give a rule to choose the best function value after each iteration, which is defined by

$$f_{best}^k = \min\{f_{best}^{k-1}, f^k\}.$$

Obviously, the sequence $\{f_{best}^k\}$ is nonincreasing.
Then we have

$$f_{best}^k = \min\{f(x^0), f(x^1), \ldots, f(x^k)\}.$$

Thus, we can find the best objective value in k iterations. Next, we show how to choose the step size for each iteration of the algorithm.

5.2 Step size rules

There are many ways to choose step size for the subgradient method at each iteration. However, we only introduce two simple and easiest ways to update step size. As follows

- *Constant step size.* $t_k = \beta$ is a positive constant, independent of k.
- *Constant step length.* $t_k = \dfrac{\beta}{\|g^k\|_2}$ with $\beta > 0$. It follows that

$$\left\|x^{k+1} - x^k\right\|_2 = \left\|-\frac{\beta}{\|g^k\|_2} g^k\right\|_2 = \beta.$$

We will of course use these two step size for the subgradient method to solve our main problem.

5.3 Convergence of subgradient method

The subgradient method has many convergent results, but we only focus on presenting two cases: constant step size and constant step length. Because we only focus on using these two ways of choosing the step size for the algorithm. And with that choice, the subgradient method is guaranteed that the objective function value will converge to the optimal value within a sufficiently small range, *i.e.*, we have

$$\lim_{k\to\infty} f_{best}^k - f^* < \varepsilon,$$

where f^* denotes the optimal value of problem and f_{best}^k is defined above. Next, we present some given assumptions to prove this inequality.

We assume that x^* is a minimizer of f, corresponds to the optimal value f^*. We also assume that the norm of the subgradients is bounded means to exist $G > 0$ such that

$$\|g\|_2 \le G, \text{ for all } g \in \partial f(x) \text{ and } y \in \mathbb{R}^n. \tag{5}$$

An example of this is the function f satisfies the Lipschitz condition

$$|f(x) - f(y)| \le G\|x - y\|_2, \forall x, y \in \mathbb{R}^n. \tag{6}$$

The above two conditions are equivalent. Indeed, if the norm of the subgradients is bounded, let $x, y \in \mathbb{R}^n$. Let $g_x \in \partial f(x)$ and $g_y \in \partial f(y)$. Then by the definitions of g_x, g_y and apply Cauchy-Schwarz inequality, we have

$$\begin{aligned} f(x) - f(y) &\le \langle g_x, x - y\rangle \le \|g_x\|_2 \|x - y\|_2 \le G\|x - y\|_2, \\ f(y) - f(x) &\le \langle g_y, y - x\rangle \le \|g_y\|_2 \|y - x\|_2 \le G\|x - y\|_2, \end{aligned}$$

showing the validity of (6).
Now, assume that (6) is satisfied and take $x \in \mathbb{R}^n, g \in \partial f(x)$. If $g = 0$ then $\|g\|_2 = 0 \le G$. If $g \ne 0$, set $g^* = \dfrac{g}{\|g\|_2}$. Then we have

$$\|g^*\|_2 = 1 \text{ and } \langle g, g^*\rangle = \|g\|_2.$$

Take $\varepsilon > 0$ (small enough) such that $x + \varepsilon g^* \in \mathbb{R}^n$. Because g is a subgradient of f, we have

$$f(x + \varepsilon g^*) - f(x) \geq \langle g, \varepsilon g^* \rangle .$$

Hence

$$\varepsilon \|g\|_2 = \langle g, \varepsilon g^* \rangle \leq f(x + \varepsilon g^*) - f(x) \leq G \|x + \varepsilon g^* - x\|_2 = G\varepsilon.$$

Thus $\|g\|_2 \leq G$.

One more condition, we assume that there is a number $R > 0$ that satisfies $\left\|x^0 - x^*\right\|_2 \leq R$.

Proof of convergence of the subgradient method based on the Euclidean distance to the set of optimal points. With x^* and f^* already defined above, we have

$$\begin{aligned}
\left\|x^{k+1} - x^*\right\|_2^2 &= \left\|x^k - t_k g^k - x^*\right\|_2^2 \\
&= \left\|x^k - x^*\right\|_2^2 - 2t_k \left\langle x^k - x^*, g^k \right\rangle + t_k^2 \left\|g^k\right\|_2^2 \\
&\leq \left\|x^k - x^*\right\|_2^2 - 2t_k \left(f(x^k) - f^*\right) + t_k^2 \left\|g^k\right\|_2^2 ,
\end{aligned}$$

where $f^* = f(x^*)$. The last line is obtained because g^k is a subgradient of f at x^k, so that

$$f(x^*) - f(x^k) \geq \left\langle g^k, x^k - x^* \right\rangle .$$

By applying the inequality above recursively, we have

$$\left\|x^{k+1} - x^*\right\|_2^2 \leq \left\|x^0 - x^*\right\|_2^2 - 2\sum_{i=0}^{k} t_i \left(f(x^i) - f^*\right) + \sum_{i=0}^{k} t_i^2 \left\|g^i\right\|_2^2.$$

Using two condition $\left\|x^0 - x^*\right\|_2 \leq R$ and $\left\|x^{k+1} - x^*\right\|_2^2$, we have

$$\begin{aligned}
0 \leq \left\|x^{k+1} - x^*\right\|_2^2 &\leq \left\|x^0 - x^*\right\|_2^2 - 2\sum_{i=0}^{k} t_i \left(f(x^i) - f^*\right) + \sum_{i=0}^{k} t_i^2 \left\|g^i\right\|_2^2 \\
&\leq R^2 - 2\sum_{i=0}^{k} t_i \left(f(x^i) - f^*\right) + \sum_{i=0}^{k} t_i^2 \left\|g^i\right\|_2^2.
\end{aligned}$$

Thus

$$2\sum_{i=0}^{k} t_i \left(f(x^i) - f^*\right) \leq R^2 + \sum_{i=0}^{k} t_i^2 \left\|g^i\right\|_2^2.$$

Since f^* is the optimal (minimum) value, so $f(x^i) - f^* \geq 0, i = \overline{1,k}$. Then we have

$$\sum_{i=0}^{k} t_i \left(f(x^i) - f^*\right) \geq \left(\sum_{i=0}^{k} t_i\right) \min_{i=0,\ldots,k} \left(f(x^i) - f^*\right) \geq \left(\sum_{i=0}^{k} t_i\right) \left(f_{best}^k - f^*\right).$$

Applying this inequality to the above inequality, we get

$$2\left(\sum_{i=0}^{k} t_i\right) \left(f_{best}^k - f^*\right) \leq R^2 + \sum_{i=0}^{k} t_i^2 \left\|g^i\right\|_2^2.$$

Then

$$f_{best}^k - f^* \leq \frac{R^2 + \sum_{i=0}^{k} t_i^2 \left\|g^i\right\|_2^2}{2\sum_{i=0}^{k} t_i}. \tag{7}$$

Finally, using the assumption $\|g\|_2 \leq G,$ for all $g \in \partial f(x)$ and $x \in \mathbb{R}^n$, we have

$$f_{best}^k - f^* \leq \frac{R^2 + G^2 \sum_{i=0}^{k} t_i^2}{2\sum_{i=0}^{k} t_i}. \tag{8}$$

This is an important inequality that helps us infer convergent results in many different cases.

Constant step size. When $t_k = \beta$ is a positive constant, independent of k, from inequality (8), we have

$$f_{best}^k - f^* \leq \frac{R^2 + G^2(k+1)\beta^2}{2(k+1)\beta}.$$

Let $k \to \infty$, we have

$$0 \leq \lim_{k\to\infty} f_{best}^k - f^* \leq \frac{G^2\beta}{2}.$$

Thus, f_{best}^k converges to the optimal value in the interval $\left[0, \frac{G^2\beta}{2}\right]$. Furthermore, for $k+1 \leq \frac{2R^2}{G^2\beta^2}$ we also have

$$f_{best}^k - f^* \leq \frac{R^2 + G^2(k+1)\beta^2}{2(k+1)\beta} \leq \frac{G^2\beta}{2}.$$

Which means we can find that $f_{best}^k - f^* \leq \frac{G^2\beta}{2}$ within at most $\frac{2R^2}{G^2\beta^2} - 1$ steps.

Constant step length. When $t_k = \frac{\beta}{\|g^k\|_2}$ with $\beta > 0$, the inequality (7) becomes

$$f_{best}^k - f^* \leq \frac{R^2 + (k+1)\beta^2}{2\sum_{i=0}^{k} \frac{\beta}{\|g^i\|_2}} \leq \frac{R^2 + (k+1)\beta^2}{2\sum_{i=0}^{k} \frac{\beta}{G}} = \frac{GR^2 + (k+1)G\beta^2}{2(k+1)\beta},$$

by using $\|g\|_2 \leq G, \forall g \in \partial f(x)$ and $x \in \mathbb{R}^n$.
Let $k \to \infty$, we have

$$0 \leq \lim_{k\to\infty} f_{best}^k - f^* \leq \frac{G\beta}{2}.$$

Therefore, in this case the subgradient method converges to an optimal value in the interval $\left[0, \frac{G\beta}{2}\right]$.

6 Solve main problem

Now we begin solving the main problem of the project. We will use two data sets: leukemia and colon-cancer with some properties:

- Colon-Cancer:
  Number of classes: 2
  Number of data: 62
  Number of features: 2000
  Preprocessing: shuffle, normalize and split data into 80. So the number of training data is 49 and the number of test data is 13.

- Leukemia:
  Number of classes: 2
  Number of data: 38 (training) / 34 (testing)
  Number of features: 7129
  Preprocessing: normalize

In all trainings, we set $C = 100$ and $\alpha = 0.01$.

6.1 Use a solver to train (2) with $R = 0$ and $R = \alpha\|\cdot\|_1$, and test on two data sets: leukemia and colon-cancer

Recall problem (2)

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m} \max(0, 1 - y_i(w^T x^i + b)) + R(w).$$

Set

$$\eta_i = \max(0, 1 - y_i(w^T x^i + b)) \Rightarrow \begin{cases} \eta_i \geq 0, & i = \overline{1,m} \\ \eta_i \geq 1 - y_i(w^T x^i + b), & i = \overline{1,m} \end{cases}$$
$$\Rightarrow \begin{cases} -\eta_i \leq 0, & i = \overline{1,m} \\ -\eta_i - y_i(w^T x^i + b) \leq -1, & i = \overline{1,m} \end{cases}.$$

Then the constraints will change slightly. For each data pair $(\mathbf{x^n}, y_n)$, instead of hard constraints $y_n(\mathbf{w^T x^n} + b) \geq 1$, we will have some soft constraints:

$$-\eta_i - y_i(w^T x^i + b) \leq -1, i = \overline{1,m},$$

with $-\eta_i \leq 0, i = \overline{1,m}$.
Now, we have equivalence problem

$$\begin{array}{ll} \min\limits_{w,b,\eta_i} & \frac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{m} \eta_i + R(w) \\ \text{s.t} & -\eta_i \leq 0 \qquad\qquad\qquad i = \overline{1,m} \\ & -\eta_i - y_i(w^T x^i + b) \leq -1 \quad i = \overline{1,m} \end{array} \tag{9}$$

with $w \in \mathbb{R}^n, \eta_i, b \in \mathbb{R}, i = \overline{1,m}$.
To solve the above problem, we have to change the above problem into the "Quadratic Programming". To do that, We will consider two cases in turn: $R(w) = 0$ and $R(w) = \alpha\|w\|_1$.

- In the case $R(w) = 0$, by setting

$$z = \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ b \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix} \in \mathbb{R}^{n+m+1}, P = \begin{bmatrix} 1 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \in \mathbb{R}^{n+m+1}, q = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^{n+m+1}$$

So, the objective function will be changed into $\frac{1}{2}z^T Pz + q^T z$.
We continue to set:

$$G = \begin{bmatrix} 0 & \dots & 0 & 0 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & \dots & -1 \\ -y_1 x_1^1 & \dots & -y_m x_n^1 & -y_1 & -1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -y_m x_1^m & \dots & -y_m x_n^m & -y_m & 0 & \dots & -1 \end{bmatrix} \in \mathbb{R}^{2m \times (n+m+1)}, h = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ \vdots \\ -1 \end{bmatrix} \in \mathbb{R}^{2m},$$

Then we can write the constraints as $Gz \preceq h$.
So we can rewrite problem (9) as

$$\begin{array}{ll} \min\limits_{z} & \frac{1}{2}z^T Pz + q^T z \\ \text{s.t} & Gz \preceq h \end{array} . \tag{10}$$

Note that the "new" problem is *Quadratic programming*. We have changed the first problem into the Quadratic programming problem. We can solve this problem by using CVXOPT package in Python.
The results of the problem with the Leukemia data set are given in the following table:

w	$[[-1.26669610e-03]$ $[-3.15946126e-04]$ $[4.41733953e-04]$ $\dots$ $[8.97540222e-05]$ $[-2.47747354e-03]$ $[-1.30346180e-04]]$
b	$[0.42760965]$
Running time	56.21150183677673
Accuracy	82.35%

The results of the problem with the Colon Cancer data set are given in the following table:

w	$[[0.00313482]$ $[-0.00064282]$ $[-0.00246087]$ $\dots$ $[-0.00445039]$ $[-0.00216628]$ $[-0.00490985]]$
b	$[-0.49607937]$
Running time	1.84940886497749756
Accuracy	84.62%

- In the case $R(w) = \alpha \|w\|_1$, the problem is

$$\begin{array}{lll} \min\limits_{w,b,k} & \frac{1}{2}\|w\|^2 + C\sum\limits_{i=1}^{m}\eta_i + \alpha\sum\limits_{i=1}^{m}|w_i| & \\ \text{s.t} & -\eta_i \leq 0 & i = \overline{1,m} \\ & -\eta_i - y_i(w^T x^i + b) \leq -1 & i = \overline{1,m} \end{array}$$

with $w \in \mathbb{R}^n, \eta_i, b \in \mathbb{R}, i = \overline{1,m}$.
We try to convert the problem to the equivalence problem again. For any $x \in \mathbb{R}$, we have

$$x = x^+ - x^-, \text{ where } \begin{cases} x^+ = \max\{0, x\} \\ x^- = \max\{0, -x\} \end{cases}$$

and

$$|x| = x^+ + x^- = u + v, \text{ where } \begin{cases} u = x^+ = \max\{0, x\} \geq 0 \\ v = x^- = \max\{0, -x\} \geq 0 \end{cases}.$$

By this way, for any $w \in \mathbb{R}^n$, we have

$$w_i = u_i - v_i \text{ and } |w_i| = u_i + v_i, \text{ with } u_i, v_i \geq 0, i = \overline{1,n}.$$

Set $u = (u_1, \ldots, u_n)^T, v = (v_1, \ldots, v_n)^T$, then

$$w = u - v \text{ and } \|w\|_1 = \sum_{i=1}^{n} |w_i| = \sum_{i=1}^{n} u_i + v_i.$$

Now, apply it to the problem we get

$$\begin{array}{rll} \min\limits_{u,v,b,k} & \frac{1}{2}\|u - v\|^2 + C\sum\limits_{i=1}^{m} \eta_i + \alpha \sum\limits_{i=1}^{m} u_i + v_i & \\ \text{s.t} & -\eta_i \leq 0 & i = \overline{1,m} \\ & -\eta_i - y_i((u-v)^T x^i + b) \leq -1 & i = \overline{1,m} \\ & -u_i \leq 0 & i = \overline{1,n} \\ & -v_i \leq 0 & i = \overline{1,n} \end{array}$$

which is equivalent to

$$\begin{array}{rll} \min\limits_{u,v,b,k} & \frac{1}{2}\|u\|^2 + \frac{1}{2}\|v\|^2 - u^T v + C\sum\limits_{i=1}^{m} \eta_i + \alpha \sum\limits_{i=1}^{m} u_i + v_i & \\ \text{s.t} & -\eta_i \leq 0 & i = \overline{1,m} \\ & -\eta_i - y_i u^T x^i + y_i v^T x^i - y_i b \leq -1 & i = \overline{1,m} \\ & -u_i \leq 0 & i = \overline{1,n} \\ & -v_i \leq 0 & i = \overline{1,n} \end{array}$$

Set

$$z = \begin{bmatrix} u_1 \\ \vdots \\ u_n \\ v_1 \\ \vdots \\ v_n \\ b \\ \eta_1 \\ \vdots \\ \eta_m \end{bmatrix}, P_1 = \begin{bmatrix} 1 & \ldots & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 1 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 1 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & 0 & \ldots & 1 & 0 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \end{bmatrix},$$

$$P_2 = \begin{bmatrix} 0 & \dots & 0 & -1 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & -1 & 0 & 0 & \dots & 0 \\ -1 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & -1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{bmatrix}, q = \begin{bmatrix} \alpha \\ \vdots \\ \alpha \\ \alpha \\ \vdots \\ \alpha \\ 0 \\ C \\ \vdots \\ C \end{bmatrix},$$

Then we have the objective function:

$$z^T P z + q^T z$$

with $P = \frac{1}{2}P_1 + P_2$.

We continue to set:

$$G = \begin{bmatrix} 0 & \dots & 0 & 0 & \dots & 0 & 0 & -1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & -1 \\ -y_1 x_1^1 & \dots & -y_1 x_n^1 & y_1 x_1^1 & \dots & y_1 x_n^1 & -y_1 & -1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -y_m x_1^m & \dots & -y_m x_n^m & y_m x_1^m & \dots & y_m x_n^m & -y_m & 0 & \dots & -1 \\ -1 & \dots & 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & -1 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 0 & \dots & 0 & -1 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & \dots & -1 & 0 & 0 & \dots & 0 \end{bmatrix}, h = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 \\ \vdots \\ -1 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

$z, q \in \mathbb{R}^{2n+m+1}; P_1, P_2 \in \mathbb{R}^{(2n+m+1)\times(2n+m+1)}, G \in \mathbb{R}^{(2n+2m)\times(2n+m+1)}, h \in \mathbb{R}^{2n+2m}$.

Then we can write the constraints as $Gz \preceq h$.

With the above setting, we get the problem

$$\begin{array}{ll} \min\limits_{z} & z^T P z + q^T z \\ \text{s.t} & Gz \preceq h \end{array}. \tag{11}$$

Where $P = \frac{1}{2}P_1 + P_2$ and note that the "new" problem is also *Quadratic programming*. We have changed the first problem into the Quadratic programming problem. We can solve this problem by using CVXOPT package in Python.

The results of the problem with the Colon Cancer data set are given in the following table:

w	$[[2.87447297e-03]$ $[-2.84842574e-13]$ $[-2.04800234e-03]$ ... $[-4.40001929e-03]$ $[-1.70090032e-03]$ $[-4.91259971e-03]]$
b	$[-0.49236312]$
Running time	130.57432007789612
Accuracy	84.62%

6.2 Use the subgradient method to solve this problem

Set

$$F(w,b) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\max(0, 1 - y_i(w^T x^i + b)) + R(w) = f(w,b) + \sum_{i=1}^{m} g_i(w,b) + R(w,b),$$

with $f(w,b) = \frac{1}{2}\|w\|^2, g_i(w,b) = C.\max(0, 1 - y_i(w^T x^i + b)), i = \overline{1,m}$ and $R(w,b) = R(w)$.
To use the subgradient method, we have to calculate the subdifferential of the function F. Suppose $F : \mathbb{R}^n \to \mathbb{R}$. Apply Sum rule 1 and Sum rule 2, we have:

$$\begin{aligned}\partial F(w,b) = \partial\left(f + \sum_{i=1}^{m} g_i(w,b) + R\right)(w,b) &= \partial f(w,b) + \sum_{i=1}^{m}\partial g_i(w,b) + \partial R(w,b)\\ &= \nabla f(w,b) + \sum_{i=1}^{m}\partial g_i(w,b) + \partial R(w,b).\end{aligned}$$

Now we have to calculate the subdifferential of $g_i(w,b)$ and $R(w,b)$. We have

$$g_i(w,b) = C.\max(0, 1 - y_i(w^T x^i + b)), i = \overline{1,m},$$

and $0, 1 - y_i(x)$ are convex, differentiable.
Then applying (3) gives us

$$\partial g_i(x) = \begin{cases} 0, & \text{if } y_i(w^T x^i + b)) > 1,\\ \left[0, \nabla C(1 - y_i(w^T x^i + b))\right], & \text{if } y_i(w^T x^i + b)) = 1,\\ \nabla C(1 - y_i(w^T x^i + b)), & \text{if } y_i(w^T x^i + b)) < 1.\end{cases} \tag{12}$$

On the other hand

- If $R(w) = 0$, then $\partial R(w,b) = \nabla R(w,b) = 0$.
- If $R(w) = \alpha\|w\|_1$, then

$$\begin{aligned}\partial R(w,b) &= \left\{v \in \mathbb{R}^n \middle| \sum_{i=1}^{n-1}|y_i| - \sum_{i=1}^{n-1}|w_i| \geq \sum_{i=1}^{n-1} v_i(y_i - w_i) + v_n(y_n - b), \forall y \in \mathbb{R}^n\right\}\\ &= \left\{v \in \mathbb{R}^{n-1} \middle| \sum_{i=1}^{n-1}|y_i| - \sum_{i=1}^{n-1}|w_i| \geq \sum_{i=1}^{n-1} v_i(y_i - w_i), \forall y \in \mathbb{R}^n\right\}\\ &\quad \times \{v_n \in \mathbb{R} | 0 \geq v_n(y_n - b), \forall y_n \in \mathbb{R}\}\\ &= I_1 \times I_2 \times \ldots \times I_{n-1} \times \{0\}\end{aligned}$$

with

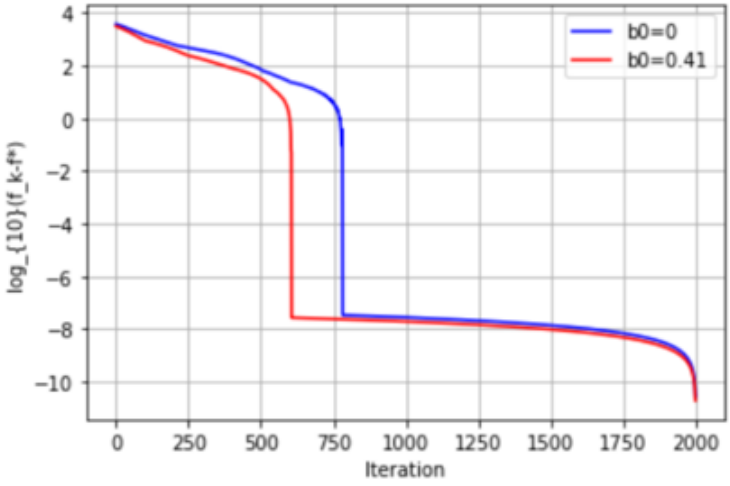
$$I_i = \begin{cases} \alpha.\text{sign}(w_i), & \text{if } w_i \neq 0 \\ [-\alpha, \alpha], & \text{if } w_i = 0 \end{cases}.$$

The last equality is obtained by applying (4).
After finishing calculating the subdifferential, we apply the subgradient method as described in the previous section to solve the problem.

- Now we will present the result of the problem with $R(w) = 0$.

  With the Leukemia data set, we use step size $= 0.000000005$ and $step\_size = \frac{step\_size}{\ln(\sqrt{step\_size})}$.
  The results of the problem with the leukemia data set are given in the following table:

w	$[[-1.11526139e-03]$ $[4.64555842e-05]$ $[1.85185450e-04]$ ... $[5.04043448e-04]$ $[-2.13409519e-03]$ $[-8.66585502e-05]]$
b	$[0.003265]$
Running time	22.971091747283936
Accuracy	82.35%



-

  With the Colon Cancer data set, we use step size $= 0.000015$ and $step\_size = \frac{step\_size}{\ln(\sqrt{step\_size})}$.
  The results of the problem with the Colon Cancer data set are given in the following table:

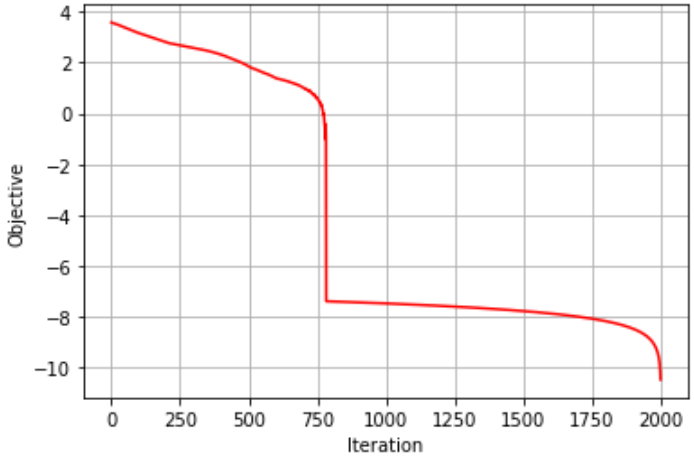
w	$[[0.0205644]$ $[0.01541731]$ $[0.00023885]$ ... $[-0.00568798]$ $[-0.00505122]$ $[-0.00354883]]$
b	$[-0.07102215]$
Running time	327.9976124763489
Accuracy	61.54%

- Now we will present the result of the problem with $R(w) = \alpha||.||_1$.

  With the Leukemia data set, we use step size = 0.000000005 and $step\_size = \frac{step\_size}{\ln(\sqrt{step\_size})}$.

  The results of the problem with the leukemia data set are given in the following table:

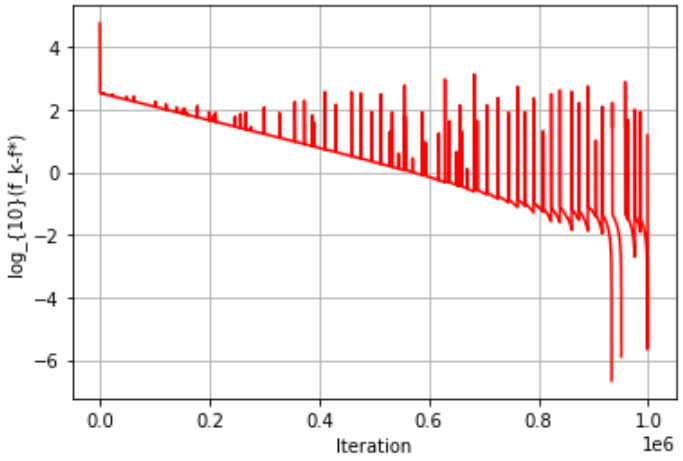
w	$[[-1.11526139e-03]$ $[4.64555842e-05]$ $[1.85185450e-04]$ ... $[5.04043448e-04]$ $[-2.13409519e-03]$ $[-8.66585502e-05]]$
b	[0.003265]
Running time	22.97109174728393
Accuracy	82.35%



With the Colon Cancer data set, we use step size = 0.000005 and $step\_size = \frac{step\_size}{\ln(\sqrt{step\_size})}$.

The results of the problem with the Colon Cancer data set are given in the following table:

w	$[[0.0098776]$ $[0.00340655]$ $[-0.00020928]$ ... $[-0.01041994]$ $[0.00069639]$ $[-0.00509361]]$
b	[0.22630316]
Running time	1148.26150393486
Accuracy	61.54%



6.3 Write and solve the dual problem of (2) with $R = 0$, and evaluate on the data sets

Consider primal problem:

$$\begin{aligned} \min_{\mathbf{w},b,\eta} \quad & \frac{1}{2}\|w\|^2 + C\sum_{n=1}^{m}\eta_i \\ \text{s.t} \quad & -\eta_i \leq 0 \qquad\qquad\qquad\qquad i = \overline{1,m} \\ & -\eta_i - y_i(\mathbf{w}^T\mathbf{x}^i + b) \leq -1 \quad i = \overline{1,m} \end{aligned}$$

with $\mathbf{w} \in \mathbb{R}^n, \eta \in \mathbb{R}^m, b \in \mathbb{R}$.
Clearly, this is the convex problem (norm and linear functions).
First, we need to we need to check the Slater condition for the convex optimization problem:
Consider constraints, for all $i = \overline{1,m}$, we choose $\mathbf{w} = \mathbf{0}, b = 0$ and $\eta_i = 2$, we have

$$y_i(\mathbf{w}^T\mathbf{x}^i + b) + \eta_i = 2 > 1.$$

Therefore, there exist $(\mathbf{w}, b, \eta) = (\mathbf{0}, 0, \eta)$ is a strictly feasible point where $\eta = \begin{bmatrix} 2 \\ 2 \\ \vdots \\ 2 \end{bmatrix} \in \mathbb{R}^m$.

Thus, Slater criterion is satisfied, then we have strong duality, the optimal values of the primal and dual problems are the same and KKT condition is the necessary and sufficient condition of primal and dual problem.
The Lagrangian of the problem is:

$$L(\mathbf{w}, b, \eta, \lambda, \mu) = \frac{1}{2}\ \|\mathbf{w}\|_2^2 + C\sum_{i=1}^{m}\eta_i + \sum_{i=1}^{m}\lambda_i(1 - \eta_i - y_i(\mathbf{w}^T\mathbf{x}^i + b)) - \sum_{i=1}^{m}\mu_i\eta_i,$$

where $\lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_m \end{bmatrix} \in \mathbb{R}_+^m, \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_m \end{bmatrix} \in \mathbb{R}_+^m.$
The dual objective function is:

$$g(\lambda, \mu) = \min_{\mathbf{w},b,\eta} L(\mathbf{w}, b, \xi, \lambda, \mu)$$

To find the dual objective function we need to minimize the Lagrangian with respect to $\mathbf{w}, b, \eta$. The minimizer of the Lagrangian is attained at the stationary point of the Lagrangian which is the solution to

$$\begin{aligned}
&\nabla_{w,b,\eta} L(\mathbf{w}, b, \xi, \lambda, \mu) = 0 \\
\Leftrightarrow &\begin{cases} \dfrac{\partial L}{\partial \mathbf{w}} = 0 \\ \dfrac{\partial L}{\partial b} = 0 \\ \dfrac{\partial L}{\partial \eta} = 0 \end{cases} \Leftrightarrow \begin{cases} \mathbf{w} - \displaystyle\sum_{i=1}^{m} \lambda_i y_i \mathbf{x}^i = 0 \\ \displaystyle\sum_{i=1}^{m} \lambda_i y_i = 0 \\ \mathbf{C} - \lambda - \mu = 0 \end{cases} \Leftrightarrow \begin{cases} \mathbf{w} = \displaystyle\sum_{i=1}^{m} \lambda_i y_i \mathbf{x}^i \\ \displaystyle\sum_{i=1}^{m} \lambda_i y_i = 0 \\ \lambda = \mathbf{C} - \mu \end{cases}
\end{aligned}$$

where $\mathbf{C} = \begin{bmatrix} C \\ C \\ \vdots \\ C \end{bmatrix} \in \mathbb{R}^m_+$.

Substituting this value back into the Lagrangian we obtain that

$$\begin{aligned}
g(\lambda, \mu) &= \frac{1}{2} w^T w - w^T (\sum_{i=1}^{m} \lambda_i y_i x_i) - \sum_{i=1}^{m} \lambda_i y_i b + \sum_{i=1}^{m} \lambda_i \\
&= -\frac{1}{2} w^T w + \sum_{i=1}^{m} \lambda_i \\
&= \sum_{i=1}^{m} \lambda_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j.
\end{aligned}$$

We can rewrite the dual problem as:

$$\begin{aligned}
\min_{\lambda} \quad & \frac{1}{2} \lambda^T \mathbf{P} \lambda - \mathbf{q}^T \lambda \\
\text{s.t} \quad & \lambda_i \leq C && i = \overline{1, m} \\
& -\lambda_i \leq 0 && i = \overline{1, m} \\
& y^T \lambda = 0
\end{aligned}$$

where $\mathbf{P} = \begin{bmatrix} y_1 \mathbf{x}_1 \\ y_2 \mathbf{x}_2 \\ \vdots \\ y_m \mathbf{x}_m \end{bmatrix} \times \begin{bmatrix} y_1 \mathbf{x}_1 \\ y_2 \mathbf{x}_2 \\ \vdots \\ y_m \mathbf{x}_m \end{bmatrix}^T \in \mathbb{R}^{m \times m}, \mathbf{q} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \in \mathbb{R}^m$.

Now we will present the result of the dual problem with $R(w) = 0$.

The results of the dual problem with the Leukemia data set are given in the following table:

w	$[[-1.26669610e-03]$ $[-3.15946126e-04]$ $[4.41733953e-04]$ $\dots$ $[8.97540222e-05]$ $[-2.47747354e-03]$ $[-1.30346180e-04]]$
b	$[0.42760965]$
Running time	0.02834463119506836
Accuracy	82.35%

The results of the dual problem with the Colon Cancer data set are given in the following table:

w	$[[0.00313482]$ $[-0.00064282]$ $[-0.00246087]$ $\dots$ $[-0.00445039]$ $[-0.00216628]$ $[-0.00490985]]$
b	$[-0.49607937]$
Running time	0.020478487014770508
Accuracy	84.62%

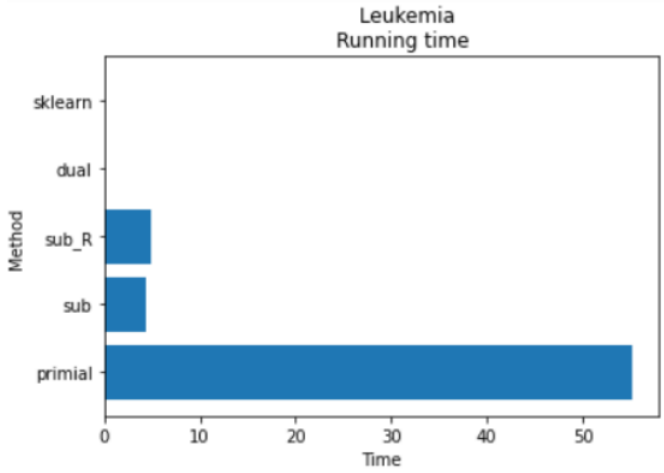
Now we present the solution of the problem by package sklearn.svm.SVC with R=0. With the Leukemia data set, we have the table below:

w	$[[-1.26629026e-03]$ $[-3.15977155e-04]$ $[4.41701996e-04]$ $\dots$ $[8.94080208e-05]$ $[-2.47747125e-03]$ $[-1.30397536e-04]]$
b	$[0.4275808701001927]$
Running time	0.026788711547851562
Accuracy	82.35%

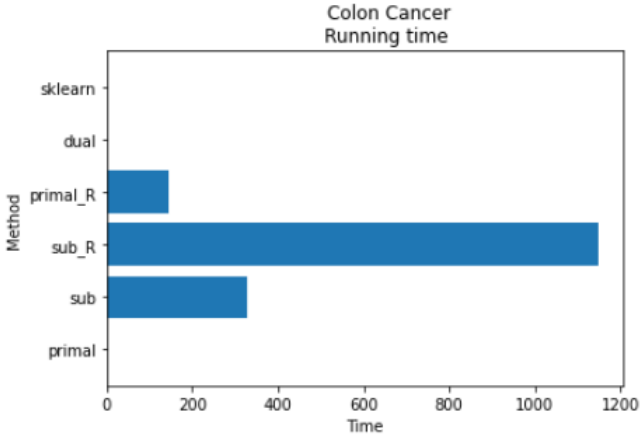
With the Colon Cancer data set, we have the table below

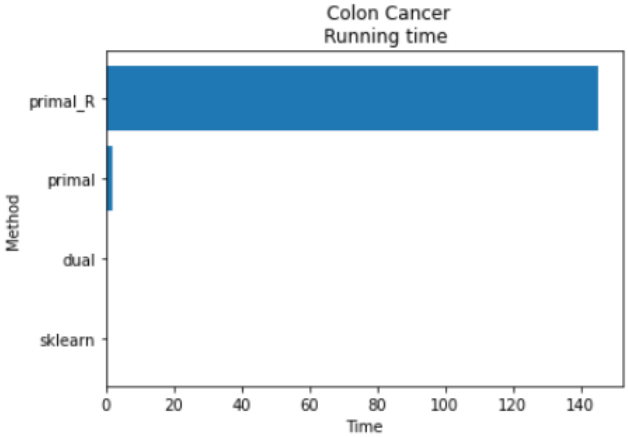
w	$[[0.00313429]$ $[-0.00064252]$ $[-0.00246048]$ $\dots$ $[-0.00444992]$ $[-0.00216481]$ $[-0.00490751]]$
b	$[-0.49607343317600394]$
Running time	0.011198282241821289
Accuracy	84.62%

In this report, we have shown some method to solve the Support Vector Machine problem. The running time of those method is given by the following pictures. The following picture is the running time of the Leukemia data set:



The picture below is the running time of the Colon Cancer data set:





7 Conclusions and discussions

1. Converting the problem to a constrained problem and using the CVXOPT.Solvers.qp package to help to solve the Quadratic Programming problem makes it possible to calculate the solution accurately.

But the computational volume is very large. In particular:

- In the Leukemia dataset when we convert the problem with $R = \alpha||\cdot||_1$ to a constrained problem, we need to find a solution $z \in \mathbb{R}^{14287}$ and matrix $G \in \mathbb{R}^{14316\times14288}$. Then Colab ran out of RAM and could not solve it.
- In the Colon Cancer dataset, fortunately Colab can solve it but has the longest solving time among the methods.

2. Using the Subgradient method to solve the unconstrained problem helps us avoid having to face complicated computational problems. But the method will find incorrect solution. We can be stuck in the neighborhood of the solution. Besides, choosing the right step size and updating step is very important. They affect convergence and convergence time.

3. For the two datasets Leukemia and Colon Cancer, both have the characteristic that the number **N-feature** is very much larger than the number **m-sample**. Therefore, solving the dual problem with the solution $\lambda \in \mathbb{R}^m$ helps to reduce the amount of computation a lot, compared to the original constrained problem. The running time is approximately the same as the running time when we use the sklearn.svm.SVC library.

4. Source code: `https://colab.research.google.com/drive/14t2drgV2X25BaSYzexByv5_P-ySxdPWi?usp=sharing`

References

[1] Beck, A. (2017). *First-order methods in optimization*. Society for Industrial and Applied Mathematics.

[2] Boyd, S., Xiao, L., & Mutapcic, A. (2003). Subgradient methods. *lecture notes of EE392o, Stanford University, Autumn Quarter, 2004*, 2004-2005.

[3] `https://machinelearningcoban.com/`.

[4] `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`