# Support Vector Machine: Mathematical Derivation and Program Implementation

Enzhi Li

July 15, 2018

## I Introduction

Support vector machine (SVM) algorithm was initially designed to solve the linearly separable problem. Assume that we have a set of data, $(\boldsymbol{x}_i, y_i), i = 1, 2, ..., N$. Here, $\boldsymbol{x}_i \in \mathbb{R}^N$ is the feature vector, and $y_i$ is the label, whose value can be only -1 or 1. For a linearly separable problem, we can find a line or hyperplane that separates the points with positive label from the points with negative label, as shown in Fig. I.1.

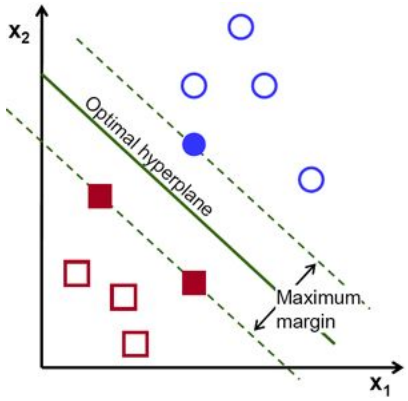


Figure I.1: A two dimensional illustration of SVM

As can be seen from the figure, a single line separates the blue points from the red points. The determination of this single line is called training SVM, and this separating line is called the optimal separating line. This article aims to derive the algorithm for calculating the optimal separating line and also discusses one method of solving the resulting convex optimization problem numerically.

The structure of the article is as follows. In section II, I will give the definition of optimal separating line or hyperplane. In section III, I will show how to find the optimal separating line

using the Lagrangian dual method. The interior point method which is a convex optimization algorithm will be employed in section IV to solve the Lagrangian dual problem. The program implementation of the SVM algorithm and the experimental results are shown in section V. I also include two appendices. Appendix A derives the formula for calculating the distance from a point to a hyperplane. This formula is frequently used during the derivation of SVM algorithm. Appendix B generalizes the SVM to the case where the points cannot be perfectly separated using a single line.

## II Optimal separating line

Assume that we have a set of linearly separable data points $(\boldsymbol{x}_i, y_i), i = 1, 2, ..., N$. For this set of data, we can always find a hyperplane that satisfies the condition that $\beta^T x + \beta_0 = 0$. With this separating hyperplane, the points the lie on one side of this plane satisfy $\beta^T x + \beta_0 > 0$, whereas the points that lie on the other side satisfy $\beta^T x + \beta_0 < 0$. With some proper rescaling, we can make the positively labeled points satisfy the condition that $\beta^T x + \beta_0 \geq 1$, and the negatively labeled points satisfy $\beta^T x + \beta_0 \leq -1$. For any data set $(\boldsymbol{x}_i, y_i)$, there are more than one set of parameters $\beta, \beta_0$ that satisfy these conditions. However, there is one set of parameter for which the distance between the two planes $\beta^T + \beta_0 = \pm 1$ is largest. The hyperplane corresponding to this set of parameters is called the optimal separating line (plane). The training of SVM using given data is just the determination of this set of optimal parameters.

In order to determine this set of optimal parameters, I need to calculate the distance between plane $P_1 : \beta^T x + \beta_0 = 1$ and plane $P_2 : \beta^T x + \beta_0 = -1$. It is known that the distance between point $x_0 \in \mathbb{R}^N$ and a linear set $Ax = b$ is $d = \sqrt{(b - Ax_0)^T (AA^T)^{-1} (b - Ax_0)}$. As a result, the distance between a point $x_0$ that lies on the plane $P_2 : \beta^T x + \beta_0 = -1$ and the plane $P_1 : \beta^T x + \beta_0 = 1$ is $d = \frac{2}{||\beta||_2}$. Thus, the maximization of the separation width can be converted to this equivalent problem:

$$\begin{aligned} &\max_{\beta} \frac{1}{2} ||\beta||^2 \\ &\text{s.t.} - y_i(\beta^T x_i + \beta_0) \leq -1, i = 1, 2, ..., N \end{aligned} \tag{1}$$

The second line of the above formula comes from the restriction that a positively labeled point for which $y_i = 1$ should satisfy the condition that $\beta^T x_i + \beta_0 \geq 1$, and a negatively labeled point for which $y_i = -1$ should satisfy the condition that $\beta^T x_i + \beta_0 \leq -1$. The problem as defined in Equ. [2] is not easy to solve. In the next section, I am going to show how to convert this problem to its Lagrangian dual which can be solved more easily.

## III Lagrangian duality

In the previous section, I have shown how to convert the derivation of SVM algorithm to a standard convex optimization problem. At the end of the previous section, I have arrived a constrained optimization problem which can be solved using the Lagrangian multiplier method. First define the Lagrangian function

$$L(\beta, \beta_0; \alpha) = \frac{1}{2}\beta^T\beta + \sum_{i=1}^{N} \alpha_i(-y_i(\beta^T x_i + \beta_0) + 1) \tag{2}$$

With the employment of the KKT condition, we know that in order to obtain the optimal solution to the optimization problem, we need to set the gradients of Lagrangian function to zero, that is,

$$\begin{aligned} \nabla_\beta L(\beta, \beta_0, \alpha) &= \beta - \sum_{i=1}^{N} \alpha_i y_i x_i = 0, \\ \frac{\partial L(\beta, \beta_0, \alpha)}{\partial \beta_0} &= -\sum_{i=1}^{N} \alpha_i y_i = 0. \end{aligned} \tag{3}$$

Moreover, we should also have

$$\begin{aligned} \alpha_i &\geq 0, \forall i, \\ \alpha_i\Big( - y_i(\beta^T x_i + \beta_0) + 1\Big) &= 0, \forall i. \end{aligned} \tag{4}$$

In the above formulation, $\beta, \beta_0$ are the original variables, and $\alpha$ are the dual variables. The essence of Lagrangian dual method is the replace the original variables with the dual variables. From the gradient condition, we have $\beta = \sum_{i=1}^{N} \alpha_i y_i x_i$. Insert this into the Lagrangian function, we have

$$L = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j x_i^T x_j - \beta_0 \sum_{i=1}^{N} \alpha_i y_i + \sum_{i=1}^{N} \alpha_i \tag{5}$$

I can eliminate the second term on the right hand side of the equal sign by invoking the KKT condition that $\sum_{i=1}^{N} \alpha_i y_i = 0$. Thus, the minimization of the Lagrangian function yields the dual function, which is

$$g(\alpha) = \inf_{\beta, \beta_0} L(\beta, \beta_0; \alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j x_i^T x_j \tag{6}$$

Taking into consideration the KKT conditions, I can convert the original constrained optimiza-

tion problem to its Lagrangian dual, which is

$$\begin{aligned} & \max_{\alpha} g(\alpha) \\ \text{s.t.} \quad & \alpha_i \geq 0, \forall i, \\ & \sum_{i=1}^{N} \alpha_i y_i = 0, \end{aligned} \tag{7}$$

Next, I am going solve this dual problem using the interior point method.

# IV Interior point method to solve the Lagrangian dual problem

Now, I have converted the original optimization problem to its dual form. Next, I am going to solve this problem using the interior point method. First, I need to rewrite the Lagrangian dual problem into a form that is amenable to the interior point method. The new (equivalent) form is

$$\begin{aligned} & \min_{\alpha} -g(\alpha) \\ \text{s.t.} \quad & -\alpha_i \leq 0, \forall i, \\ & \sum_{i=1}^{N} \alpha_i y_i = 0 \end{aligned} \tag{8}$$

Second, I introduce a new function,

$$h(\alpha, \lambda; t) = -\sum_{i=1}^{N} \alpha_i + \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{N} -\frac{1}{t} \log \alpha_i + \lambda \sum_{i=1}^{N} \alpha_i y_i \tag{9}$$

For notational convenience, I define this symmetric matrix $A_{ij} = y_i y_j x_i^T x_j$. This matrix can be equivalently rewritten as $A = \xi^T \xi$, where $\xi = (y_1 x_1, y_2 x_2, ..., y_N x_N)$. Generally, the dimension of the feature vector is much smaller than the number of data points, and thus the matrix $A$ is singular. With this notation, the function $h(\alpha, \lambda; t)$ can be simplified to the form

$$h(\alpha, \lambda; t) = -\sum_{i=1}^{N} \alpha_i + \frac{1}{2} \alpha^T A \alpha + \sum_{i=1}^{N} -\frac{1}{t} \log \alpha_i + \lambda \alpha^T y \tag{10}$$

Here, $\alpha, \lambda$ are the optimization variables over which I need to minimize the target function $h(\alpha, \lambda; t)$, and $t$ is a parameter over which I am going to scan my optimized solutions. Once I know the value of $\alpha$, I can find the value of $\beta = \sum_{i=1}^{N} \alpha_i y_i x_i$. The intercept $\beta_0$ can be easily determined from the support vectors. Next, I am going calculate the values of $\alpha, \lambda$ for which the target function $h$ is minimized. To do this, I am going to set the gradient of $h$ to zero. The gradients of $h$ with respect to the optimization variables are

$$\begin{aligned} \frac{\partial h}{\partial \alpha_k} &= -1 + A_{kj} \alpha_j - \frac{1}{t} \frac{1}{\alpha_k} + \lambda y_k \\ \frac{\partial h}{\partial \lambda} &= \alpha^T y \end{aligned} \tag{11}$$

Minimizing the function $h$ is equivalent to the solution of this equation:

$$\begin{pmatrix} \frac{\partial h}{\partial \alpha} \\ \frac{\partial h}{\partial \lambda} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \tag{12}$$

This equation can be solved numerically using the Newton's method, which is

$$\begin{pmatrix} \alpha^{(n+1)} \\ \lambda^{(n+1)} \end{pmatrix} = \begin{pmatrix} \alpha^{(n)} \\ \lambda^{(n)} \end{pmatrix} - H^{-1} \begin{pmatrix} \frac{\partial h}{\partial \alpha} \\ \frac{\partial h}{\partial \lambda} \end{pmatrix} \tag{13}$$

Here, $H$ is Hessian matrix, the explicit form of which is

$$H = \begin{pmatrix} A_{ij} + \frac{1}{t}\frac{\delta_{ij}}{\alpha_i^2} & y \\ y^T & 0 \end{pmatrix}, \tag{14}$$

where $A_{ij} = y_i y_j x_i^T x_j$. With this method, I can calculate the optimized values of $(\alpha_t^\star, \lambda_t^\star)$ for fixed values of $t$. The interior point method is essentially an approximation to the original problem. The value of parameter $t$ dictates the precision of this approximation. The larger the value of $t$, the better the approximation. The limiting case $\lim_{t\to\infty}(\alpha_t^\star, \lambda_t^\star) = (\alpha^\star, \lambda^\star)$ yields the genuine values of the optimization variables.

# V Program implementation and experimental results

I have written a program to implement this SVM algorithm. The experimental results are shown in Fig. V.1.

# A Derivation of the distance formula

In this appendix, I am going derive the formula for the distanced between a point $x_0$ and a linear set. A linear set is defined as the set of points that satisfy this condition: $\Sigma : \{x : Ax = b\}$. A hyperplane is a special case of linear set. The distance between a point $x_0$ and the set $\Sigma$ is defined as

$$d = \min_{x\in\Sigma} ||x - x_0||_2 \tag{15}$$

I am going derive a formula for this distance using the convex optimization method. I am going to reformulate the distance problem as follows:

$$\begin{aligned} &\min_x \frac{1}{2}||x - x_0||^2 \\ \text{s.t.} \quad &Ax = b \end{aligned} \tag{16}$$

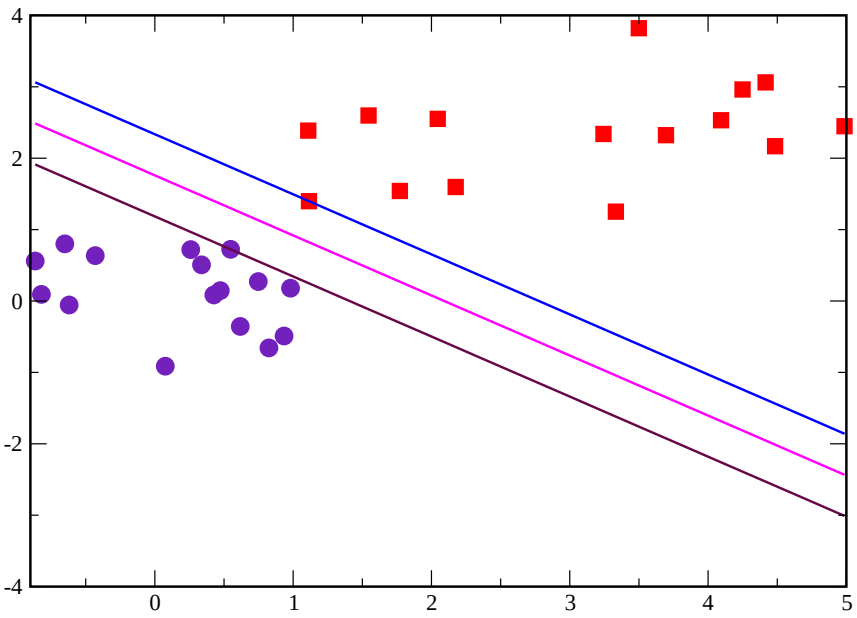


Figure V.1: Experimental results of my SVM program. The points are linearly separated.

Define the Lagrangian function as

$$L(x;\lambda) = \frac{1}{2}(x - x_0)^T(x - x_0) + \lambda^T(Ax - b) \tag{17}$$

The gradient of this function is

$$\nabla_x L = x - x_0 + A^T\lambda \tag{18}$$

Setting the gradient to zero yields $x = x_0 - A^T\lambda$. The Lagrangian dual of this problem is

$$g(\lambda) = \inf_x L(x;\lambda) = \frac{1}{2}\lambda^T AA^T\lambda + \lambda^T(Ax_0 - AA^T\lambda - b) = -\frac{1}{2}\lambda^T AA^T\lambda + \lambda^T(Ax_0 - b) \tag{19}$$

The gradient of the dual function is $\nabla_\lambda g(\lambda) = -AA^T\lambda + Ax_0 - b$. Setting the gradient to zero yields $\lambda^\star = (AA^T)^{-1}(Ax_0 - b)$. Since in a linear set, the number of constraints cannot be larger than the dimension of the space, thus the matrix $AA^T$ is invertible and can thus be safely used in the previous formula. We thus have $g(\lambda^\star) = \max_\lambda g(\lambda) = \frac{1}{2}(Ax_0 - b)^T(AA^T)^{-1}(Ax_0 - b) = \frac{1}{2}d^2$. And the distance is $d = \sqrt{(Ax_0 - b)^T(AA^T)^{-1}(Ax_0 - b)}$. The distance between a point $x_0$ and a hyperplane $P : \beta^T x + \beta_0 = 0$ is $d = \frac{||\beta^T x_0 + \beta_0||}{||\beta||}$. It is easy to see that the distance between the two planes $P_1 : \beta^T x + \beta_0 = 1; P_2 : \beta^T x + \beta_0 = -1$ is $d = \frac{2}{||\beta||}$.

## B Soft margin SVM

Up until now, I have assumed that the data points are perfectly separable, which is in general not the case in real world. We need to resort to the soft margin SVM when the data points are not perfectly separable. The only difference between the hard margin SVM and the soft margin SVM is that the dual problem of the soft margin SVM is

$$\begin{aligned} & \max_{\alpha} g(\alpha) \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, i = 1, 2, ..., N \\ & \sum_{i=1}^{N} \alpha_i y_i = 0 \end{aligned} \tag{20}$$

When $\alpha_i = 0$, we have $y_i(\beta^T x_i + \beta_0) > 1$, and the point lies outside the two boundary planes; when $\alpha_i = C$, we have $y_i(\beta^T x_i + \beta_0) < 1$, and the point lies within the two boundary planes; when $0 < \alpha_i < C$, we have $y_i(\beta^T x_i + \beta_0) = 1$, and the point lies on one of the two boundaries. I can still solve the dual problem using the interior point method. Define a target function

$$f(\alpha, \lambda; t_1, t_2) = -\sum_i \alpha_i + \frac{1}{2}\alpha^T A \alpha + \lambda \sum_i \alpha_i y_i - \frac{1}{t_1}\sum_i \log \alpha_i - \frac{1}{t_2}\sum_i \log(C - \alpha_i) \tag{21}$$

For sake of simplicity, I can set $t_1 = t_2$. Thus, the target function is simplified to the form

$$f(\alpha, \lambda; t) = -\sum_i \alpha_i + \frac{1}{2}\alpha^T A \alpha + \lambda \sum_i \alpha_i y_i - \frac{1}{t}\sum_i \log \alpha_i (C - \alpha_i) \tag{22}$$

The gradient of this function is

$$\begin{aligned} \frac{\partial f}{\partial \alpha_k} &= -1 + A_{kj}\alpha_j + \lambda y_k - \frac{1}{t}\Big(\frac{1}{\alpha_k} - \frac{1}{C - \alpha_k}\Big) \\ \frac{\partial f}{\partial \lambda} &= \alpha^T y \end{aligned} \tag{23}$$

Hessian matrix is

$$H = \begin{pmatrix} A_{kl} + \frac{1}{t}\delta_{kl}\Big(\frac{1}{\alpha_k^2} + \frac{1}{(C-\alpha_k)^2}\Big) & y \\ y^T & 0 \end{pmatrix} \tag{24}$$

When $C \to \infty$, we recover the hard margin SVM. The results are shown Fig. B.1.

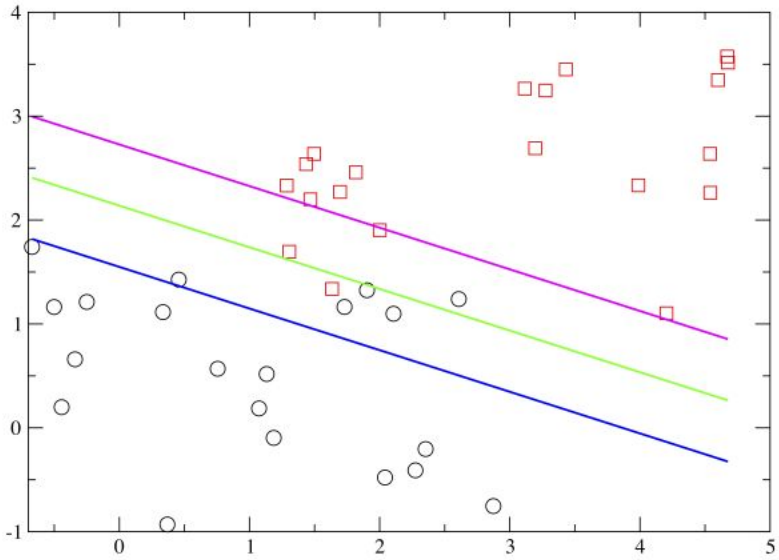


Figure B.1: Soft margin SVM results.