Lecture Notes on Support Vector Machine

Feng Li
fli@sdu.edu.cn
Shandong University, China

1 Hyperplane and Margin

In a n-dimensional space, a hyper plane is defined by

$$\omega^T x + b = 0 \tag{1}$$

where $\omega \in \mathbb{R}^n$ is the outward pointing normal vector, and b is the bias term. The n-dimensional space is separated into two half-spaces $H^+ = \{x \in \mathbb{R}^n \mid \omega^T x + b \geq 0\}$ and $H^- = \{x \in \mathbb{R}^n \mid \omega^T x + b < 0\}$ by the hyperplane, such that we can classify a given point $x_0 \in \mathbb{R}^n$ according to $\text{sign}(\omega^T x + b)$. Specifically, given a point $x_0 \in \mathbb{R}^n$, its label y is defined as $y_0 = \text{sign}(\omega^T x_0 + b)$, i.e.

$$y_0 = \begin{cases} 1, & \omega^T x_0 + b \geq 0 \\ -1, & \text{otherwise} \end{cases} \tag{2}$$

Given any $x_0 \in \mathbb{R}^n$, we can calculate the signed distance from x to the hyperplane as

$$d_0 = \frac{\omega^T x_0 + b}{\|\omega\|} = \left(\frac{\omega}{\|\omega\|}\right)^T x_0 + \frac{b}{\|\omega\|} \tag{3}$$

The sign of the distance, i.e., $\text{sign}(\gamma)$, can be indicated by $y_0 = \text{sign}(\omega^T x_0 + b)$. Therefore, we define the (unsigned) *geometric distance* of x_0 as

$$\gamma_0 = \frac{y_0(\omega^T x_0 + b)}{\|\omega\|} \tag{4}$$

γ_0 is the so-called *margin* of x_0 (with respect to the hyperplane $\omega^T x + b = 0$).

Now, given a set of m training data $\{(x^{(i)}, y^{(i)})\}_{i=1,\cdots,m}$, we first assume that they are *linearly separable*. Specifically, there exists a hyperplane (parameterized by ω and b) such that $\omega^T x^{(i)} + b \geq 0$ for $\forall i$ with $y^{(i)} = 1$, while $\omega^T x^{(i)} + b \leq 0$ for $\forall i$ with $y^{(i)} = -1$. As shown in Fig. 1, for $\forall i = 1, \cdots, m$, we can calculate its margin as

$$\gamma^{(i)} = y^{(i)} \left(\left(\frac{\omega}{\|\omega\|}\right)^T x^{(i)} + \frac{b}{\|\omega\|} \right) \tag{5}$$

With respect to the whole training set, the margin is defined as

$$\gamma = \min_i \gamma^{(i)} \tag{6}$$

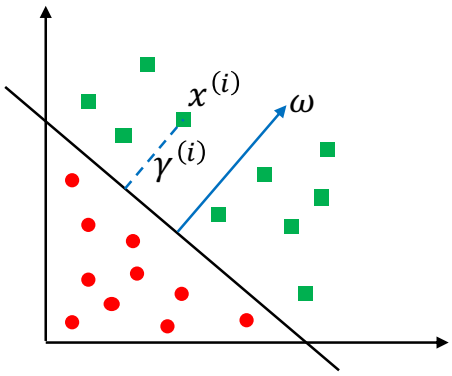


Figure 1: Margin and hyperplane.

2 Support Vector Machine

2.1 Formulation

The hyperplane actually serves as a decision boundary to differentiating positive data samples from negative data samples. Given a test data sample, we will make a more confident decision if its margin (with respect to the decision hyperplane) is larger. By leveraging different values of ω and b, we can construct a infinite number of hyperplanes, but which one is the best? *Supported Vector Machine* (SVM) answers the above question by maximizing γ (see Eq. (6)) as follows

$$\begin{aligned} &\max_{\gamma,\omega,b} \quad \gamma \\ &s.t. \quad y^{(i)}(\omega^T x^{(i)} + b) \geq \gamma \|\omega\|, \quad \forall i \end{aligned}$$

Note that scaling ω and b (e.g., by multiplying both ω and b by the same constant) does **not** change the hyperplane. Hence, we scale (ω, b) such that

$$\min_i \{y^{(i)}(\omega^T x^{(i)} + b)\} = 1,$$

In this case, the representation of the margin becomes $1/\|\omega\|$ according to Eq. (6). Then, the problem formulation can be rewritten as

$$\begin{aligned} &\max_{\omega,b} \quad 1/\|w\| \\ &s.t. \quad y^{(i)}(\omega^T x^{(i)} + b) \geq 1, \quad \forall i \end{aligned}$$

Since maximizing $1/\|\omega\|$ is equivalent to minimizing $\|\omega\|^2 = \omega^T\omega$, we further rewrite the problem formulation as

$$\min_{\omega,b} \quad \omega^T\omega \tag{7}$$

$$s.t. \quad y^{(i)}(\omega^T x^{(i)} + b) \geq 1, \quad \forall i \tag{8}$$

As shown in Fig. 2, the distance from the dashed lines ($\omega^T x + b = 1$ and $\omega^T x + b = -1$) to the hyperplane $\omega^T x + b = 0$ is the margin (see Eq. (6)). The

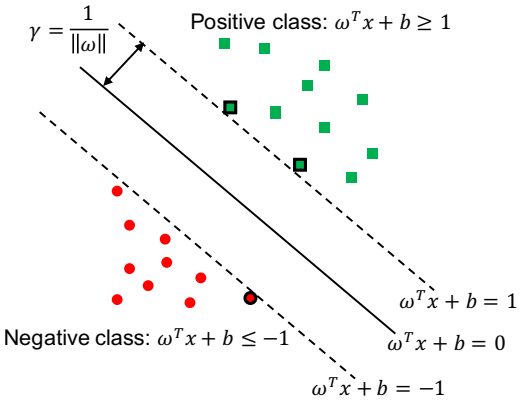


Figure 2: Hard-margin SVM.

aim of the above optimization problem is to find a hyperplane (parameterized by ω and b) with margin $\gamma = 1/\|\omega\|$ maximized, while the resulting dashed lines satisfy the following condition: for each training sample $(x^{(i)}, y^{(i)})$, $\omega^T x^{(i)} + b \geq 1$ if $y^{(i)} = 1$, and $\omega^T x^{(i)} + b \leq 1$ if $y^{(i)} = -1$.

This is a *quadratic programming* (QP) problem, and can be solved by exiting generic QP solvers, e.g., interior point method, active set method, gradient projection method. Unfortunately, the existing generic QP solvers is of low efficiency, especially in face of a large training set.

2.2 Preliminary Knowledge of Convex Optimization

2.2.1 Optimization Problems and Lagrangian Duality

We now consider the following optimization problem

$$\min_{\omega} \quad f(\omega) \tag{9}$$

$$s.t. \quad g_i(\omega) \leq 0, i = 1, \cdots, k \tag{10}$$

$$h_j(\omega) = 0, j = 1, \cdots, l \tag{11}$$

where $\omega \in \mathcal{D}$ is the variable with $\mathcal{D} = \bigcap_{i=1}^{k} \mathbf{dom} g_i \cap \bigcap_{j=1}^{l} \mathbf{dom} h_j$ representing the feasible domain defined by the constraints. The aim of the above optimization problem is to minimizing the objective function $f(\omega)$ subject to the inequality constraints $g_1(\omega), \cdots, g_k(\omega)$ and the equality constraints $h_1(\omega), \cdots, h_l(\omega)$.

We construct the *Lagrangian* of the above optimization problem as

$$\mathcal{L}(\omega, \alpha, \beta) = f(\omega) + \sum_{i=1}^{k} \alpha_i g_i(\omega) + \sum_{j=1}^{l} \beta_j h_j(\omega) \tag{12}$$

In fact, $\mathcal{L}(\omega, \alpha, \beta)$ can be treated as a weighted sum of the objective and constraint functions. α_i is the so-called *Lagrange multiplier* associated with $g_i(\omega) \leq 0$, while β_i is the one associated with $h_i(\omega) = 0$

We then define its Lagrange dual function $\mathcal{G} : \mathbb{R}^k \times \mathbb{R}^l \to \mathbb{R}$ as an infimum [1] of $\mathcal{L}$ with respect to ω, i.e.,

$$\begin{aligned} \mathcal{G}(\alpha, \beta) &= \inf_{\omega \in \mathcal{D}} \mathcal{L}(\omega, \alpha, \beta) \\ &= \inf_{\omega \in \mathcal{D}} \left(f(\omega) + \sum_{i=1}^{k} \alpha_i g_i(\omega) + \sum_{j=1}^{l} \beta_j h_j(\omega) \right) \end{aligned} \tag{13}$$

We observe that, i) the infimum is unconstrained (as supposed to the original constrained minimization problem); ii) $\mathcal{G}$ is an infimum of a set of affine functions and thus is a *concave* function regardless of the original problem; iii) $\mathcal{G}$ can be $-\infty$ for some α and β

Theorem 1. *Lower Bounds Property: If $\alpha \succeq 0$, then $\mathcal{G}(\alpha, \beta) \leq p^*$ where p^* is the optimal value of the (original) primal problem defined by (9)∼(11).*

Proof. If $\widetilde{\omega}$ is feasible, then we have $g_i(\widetilde{\omega}) \leq 0$ for $\forall i = 1, \cdots, k$ and $h_j(\widetilde{\omega}) = 0$ for $\forall j = 1, \cdots, l$. Since $\alpha \succeq 0$ (i.e., $\alpha_i \geq 0$ for $\forall i$), we have $f(\widetilde{\omega}) \geq \mathcal{L}(\widetilde{\omega}, \alpha, \beta)$ for all feasible $\widetilde{\omega}$'s. Because $\mathcal{L}(\tilde{\omega}, \alpha, \beta) \geq \inf_{\omega \in \mathcal{D}} L(\omega, \alpha, \beta)$,

$$f(\tilde{\omega}) \geq \mathcal{L}(\tilde{\omega}, \alpha, \beta) \geq \inf_{\omega \in \mathcal{D}} L(\omega, \alpha, \beta) = \mathcal{G}(\alpha, \beta)$$

holds for all feasible $\tilde{\omega}$. We now choose the minimizer of $f(\tilde{\omega})$ over all feasible $\tilde{\omega}$'s to get $p^* \geq \mathcal{G}(\alpha, \beta)$. □

It is shown by **Theorem** 1 that, the Lagrange dual function provides a non-trivial lower bound to the primal optimization problem. By optimizing the lower bound, we define the *Lagrange dual problem* with respect to the primal problem (9)∼(11) as follows

$$\max_{\alpha, \beta} \quad \mathcal{G}(\alpha, \beta) \tag{14}$$

$$s.t. \qquad \alpha \succeq 0, \quad \forall i = 1, \cdots, k \tag{15}$$

We denote by d^* the optimal value of the above Lagrange dual problem. The *weak* duality $d^* \leq p^*$ always holds for all optimization problems, and can be used to find non-trivial lower bounds. The duality is said to be *strong* if $d^* = p^*$. In this case, we can optimize the original problem by optimizing its dual problem.

2.2.2 Complementary Slackness

Let ω^* be a primal optimal point and (α^*, β^*) be a dual optimal point.

Theorem 2. *Complementary Slackness: If strong duality holds, then*

$$\alpha_i^* g_i(\omega^*) = 0 \tag{16}$$

for $\forall i = 1, 2, \cdots, k$.

[1] In mathematics, the infimum (abbreviated **inf**; plural **infima**) of a subset S of a partially ordered set T is the greatest element in T that is less than or equal to all elements of S, if such an element exists. More details about infimum and its counterpart **suprema** can be found in `https://en.wikipedia.org/wiki/Infimum_and_supremum`.

Proof.

$$\begin{aligned} f(\omega^*) &= \mathcal{G}(\alpha^*, \beta^*) \\ &= \inf_{\omega} \left(f(\omega) + \sum_{i=1}^{k} \alpha_i^* g_i(\omega) + \sum_{j=1}^{l} \beta_j^* h_j(\omega) \right) \\ &\leq f(\omega^*) + \sum_{i=1}^{k} \alpha_i^* g_i(\omega^*) + \sum_{j=1}^{l} \beta_j^* h_j(\omega^*) \\ &\leq f(\omega^*) \end{aligned}$$

The first equality is due to the strong duality, and we have the second one according to the definition of the dual function. The third inequality follows because the infimum of the Lagrangian over ω is less than or equal to its value at $\omega = \omega^*$. We have the last inequality since $\alpha_i^* \geq 0$ and $g_i(\omega^*) \leq 0$ for $\forall i = 1, \cdots, k$, and $h_j(\omega^*) = 0$ for $j = 1, \cdots, l$. In fact, the last two inequalities should hold with equality, such that $\sum_{i=1}^{k} \alpha_i^* g_i(\omega^*) = 0$. Since each term, i.e., $\alpha_i^* g_i(\omega^*)$, is nonpositive, we thus conclude $\alpha_i^* g_i(\omega^*) = 0$ for $\forall i = 1, 2, \cdots, k$. □

Another observation is that, since the inequality in the third line holds with equality, ω^* actually minimizes $\mathcal{L}(\omega, \alpha^*, \beta^*)$ over ω.

2.2.3 Karush-Kuhn-Tucker (KKT) Conditions

We assume that the objective function and the inequality constraint functions are differentiable. Again, let ω^* and (α^*, β^*) be any primal and dual optimal points, respectively, and suppose strong duality holds. Since ω^* is the minimizer of $\mathcal{L}(\omega, \alpha^*, \beta^*)$ over ω, it follows that the gradient of $\mathcal{L}$ vanishes at ω^*, i.e.,

$$\nabla f(\omega^*) + \sum_{i=1}^{k} \alpha_i^* \nabla g_i(\omega^*) + \sum_{j=1}^{l} \beta_j^* \nabla h_j(\omega^*) = 0 \tag{17}$$

which is the so-called *stationarity* condition. Since ω^* and α^* should be in the feasible domains of the primal problem and the dual problem, respectively, we have the *primal feasibility* conditions (18)∼(18) and the *dual feasibility* condition (20) holds

$$g_i(\omega^*) \leq 0, \ \forall i = 1, \cdots, k \tag{18}$$
$$h_j(\omega^*) = 0, \ \forall j = 1, \cdots, l \tag{19}$$
$$\alpha_i^* \geq 0, \ \forall i = 1, \cdots, k \tag{20}$$
$$\tag{21}$$

All these conditions (16)∼(20) are so-called *Karush-Kuhn-Tucker* (KKT) conditions. For any optimization problem with differentiable objective and constraint functions for which strong duality obtains, any pair of primal and dual optimal points must satisfy the KKT conditions.

2.2.4 Convex Optimization Problems

An optimization problem is *convex*, if both objective function $f(\omega)$ and inequality constraints $g_i(\omega)$ $(i = 1, \cdots, k)$ are convex and the equality constraints

$h_j(\omega)$ are affine functions. Therefore, a convex optimization problem can be represented by

$$\min_{\omega} \quad f(w) \tag{22}$$
$$s.t. \quad g_i(w) \leq 0, i = 1, \cdots, k \tag{23}$$
$$Aw - b = 0 \tag{24}$$

where $A \in \mathbb{R}^{l \times n}$ and $b \in \mathbb{R}^l$.

Although strong duality does not hold (in general), but we usually (but not always) have strong duality for convex optimization problems. There are many results that establish conditions on the problem, beyond convexity, under which strong duality holds. These conditions are called *constraint qualifications*. One simple constraint qualification is *Slater's condition*.

Theorem 3. *Slater's condition: Strong duality holds for a convex problem*

$$\begin{aligned} \min_{\omega} \quad & f(w) \\ s.t. \quad & g_i(w) \leq 0, i = 1, \cdots, k \\ & Aw - b = 0 \end{aligned}$$

if it is strictly feasible, i.e.,

$$\exists\ \omega \in \mathbf{relint}\mathcal{D} : g_i(\omega) < 0, i = 1, \cdots, m, Aw = b$$

Detailed proof of the above theorem can be found in Prof. Boyd and Prof. Vandenberghe's *Convex Optimization* book (`https://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf`, see Sec. 5.3.2, pp. 234-236).

For convex optimization problem, the KKT conditions are also sufficient for the points to be primal and dual optimal. In particular, suppose $\widetilde{\omega}$, $\widetilde{\alpha}$, and $\widetilde{\beta}$ are any points satisfying the following KKT conditions

$$g_i(\widetilde{\omega}) \leq 0,\ \forall i = 1, \cdots, k \tag{25}$$
$$h_j(\widetilde{\omega}) = 0,\ \forall j = 1, \cdots, l \tag{26}$$
$$\widetilde{\alpha}_i \geq 0,\ \forall i = 1, \cdots, k \tag{27}$$
$$\widetilde{\alpha}_i g_i(\widetilde{\omega}) = 0,\ \forall i = 1, \cdots, k \tag{28}$$
$$\nabla f(\widetilde{\omega}) + \sum_{i=1}^{k} \widetilde{\alpha}_i \nabla g_i(\widetilde{\omega}) + \sum_{j=1}^{l} \widetilde{\beta}_j \nabla h_j(\widetilde{\omega}) = 0 \tag{29}$$

then they are primal and dual optimal with strong duality holding.

3 Duality of SVM

We now re-visit our problem formulation of SVM. The (primal) SVM problem is given

$$\min_{\omega, b} \quad \frac{1}{2}\|\omega\|^2 \tag{30}$$
$$\text{s.t.} \quad y^{(i)}(\omega^T x^{(i)} + b) \geq 1, \quad \forall i \tag{31}$$

where we introduce a constant $1/2$ so as to simplify our later derivations.

Theorem 4. *The dual optimization problem of the primal SVM problem (30)∼(31) can be formulated as*

$$\max_{\alpha} \quad \mathcal{G}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j (x^{(i)})^T x^{(j)} \tag{32}$$

$$\text{s.t.} \quad \sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \tag{33}$$

$$\alpha_i \geq 0 \quad \forall i \tag{34}$$

Proof. We first define the Lagrangian of the primal SVM problem as

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i (y^{(i)}(w^T x^{(i)} + b) - 1) \tag{35}$$

where $\alpha_i \geq 0$ is the Lagrangian multiplier for the i-th inequality constraint. We then calculate the Lagrange dual function $\mathcal{G}(\alpha)$ by taking the infimum of $\mathcal{L}(\omega, b, \alpha)$ over ω and b. In particular, we calculate the gradient of $\mathcal{L}(\omega, b, \alpha)$ with respect to ω, and let the gradient be zero,

$$\bigtriangledown_{\omega} \mathcal{L}(\omega, b, \alpha) = w - \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} = 0$$

and we thus have

$$\omega = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} \tag{36}$$

Similarly,

$$\frac{\partial}{\partial b} \mathcal{L}(\omega, b, \alpha) = \sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \tag{37}$$

In another word, the above two equations are necessary to calculating $\inf_{\omega,b} \mathcal{L}(\omega, b, \alpha)$ over ω and b. Substituting (36) and (37) into (35) gives us

$$\begin{aligned}
& \frac{1}{2}\|\omega\|^2 - \sum_{i=1}^{m} \alpha_i [y^{(i)}(\omega^T x^{(i)} + b) - 1] \\
= \; & \frac{1}{2}\omega^T \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} - \omega^T \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} - \sum_{i=1}^{m} \alpha_i y^{(i)} b + \sum_{i=1}^{m} \alpha_i \\
= \; & -\frac{1}{2}\omega^T \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} - \sum_{i=1}^{m} \alpha_i y^{(i)} b + \sum_{i=1}^{m} \alpha_i \\
= \; & -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} (x^{(i)})^T x^{(j)} - b \sum_{i=1}^{m} \alpha_i y^{(i)} + \sum_{i=1}^{m} \alpha_i \\
= \; & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} (x^{(i)})^T x^{(j)} - b \sum_{i=1}^{m} \alpha_i y^{(i)} \\
= \; & \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} (x^{(i)})^T x^{(j)}
\end{aligned}$$

which completes our proof. □

It is a convex optimization problem respecting Slater's condition; therefore, the strong duality ($p^* = d^*$) holds and optimal solutions of ω, α and β satisfy the KKT conditions. We can use several off-the-shelf solvers (e.g., quadprog (MATLAB), CVXOPT, CPLEX, IPOPT, etc.) to solve such a QP problem.

Let α^* be the optimal value of α for the dual SVM problem. We can use Eq. (36) to calculate the optimal value of ω, i.e., ω^*. The question is, being aware of ω^*, how to calculate the optimal value of b, i.e., b^*? Due to the complementary slackness,

$$\alpha_i^*(y^{(i)}({\omega^*}^T x^{(i)} + b^*) - 1) = 0$$

for $\forall i = 1, \cdots, k$, we have

$$y^{(i)}({\omega^*}^T x^{(i)} + b^*) = 1$$

for $\forall i$ such that $\alpha_i^* > 0$. As $y^{(i)} \in \{-1, 1\}$, we have

$$b^* = y^{(i)} - {\omega^*}^T x^{(i)}$$

for $\forall i$ such that $\alpha_i^* > 0$. For robustness, we calculated the optimal value for b by taking the averages across all b^*'s

$$b^* = \frac{\sum_{i:\alpha_i^*>0}(y^{(i)} - {\omega^*}^T x^{(i)})}{\sum_{i=1}^m \mathbf{1}(\alpha_i^* > 0)}$$

In fact, most α_i's in the solution are zeros. According to complementary slackness (see **Theorem** 2),

$$\alpha_i^*[1 - y^{(i)}({\omega^*}^T x^{(i)} + b^*)] = 0$$

α_i^* is non-zero only if $x^{(i)}$ lies on the one of the two margin boundaries (i.e., the dash lines shown in Fig. 2) such that $y^{(i)}({\omega^*}^T x^{(i)} + b) = 1$. These data samples are so-called *support vector*, i.e., the vectors "supporting" the margin boundaries. We can redefine ω by

$$w = \sum_{s \in \mathcal{S}} \alpha_s y^{(s)} x^{(s)}$$

where $\mathcal{S}$ denotes the set of the indices of the support vectors

4 Kernel based SVM

By far, one of our assumption is that the training data can be separated linearly. Nevertheless, Linear models (e.g., linear regression, linear SVM etc.) cannot reflect the nonlinear pattern in the data, as demonstrated in Fig. 4.

The basic idea of kernel method is to make linear model work in nonlinear settings by introducing kernel functions. In particular, by mapping the data into a higher-dimensional feature space where it exhibits linear patterns, we can employ the linear classification model in the new feature space.

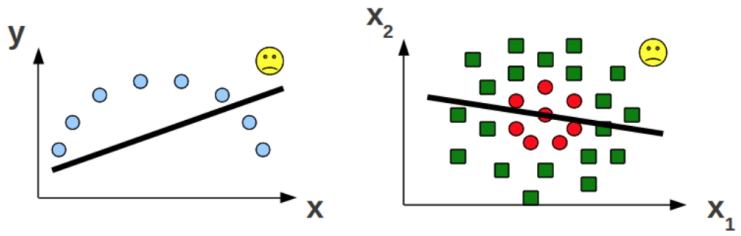


Figure 3: Non-linear data v.s. linear classifier

We take the following binary classification problem for example. As shown in Fig. 4 (a), Each sample is represented by a single feature x (i.e., the data samples lie in a 1-dimensional space), and no linear separator exists for this data. We map each data sample into a 2-dimensional space by $x \to \{x, x^2\}$, such that each sample now has two features ("derived" from the old representation). As shown in Fig. 4 (b), data become linearly separable in the new higher-dimensional feature space

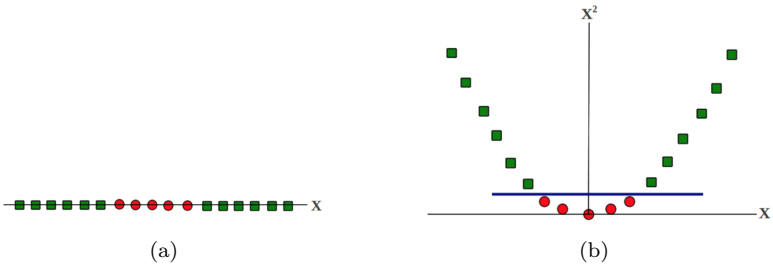


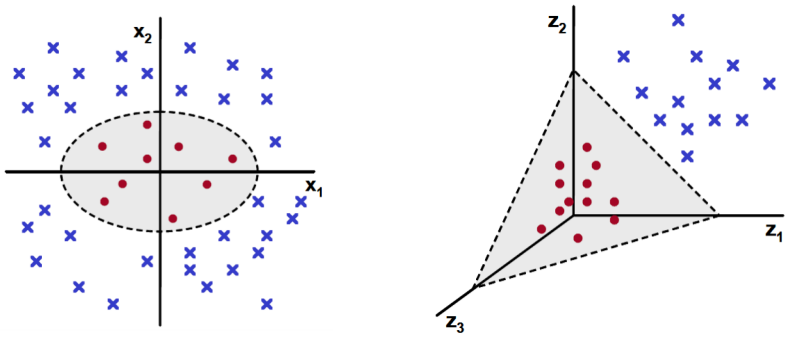
Figure 4: Feature mapping for 1-dimensional feature space.

Another example is given in Fig. 5. The data sample can be defined by $x = \{x_1, x_2\}$, and there is no linear separator exists for this data. We apply the mapping $x = \{x_1, x_2\} \to z = \{x_1^2, \sqrt{2}x_1x_2, x_2^2\}$, such that the data become linearly separable in the resulting 3-dimensional feature space.

We now consider a general quadratic feature mapping ϕ

$$\phi : x \to \{x_1^2, x_2^2, \cdots, x_n^2, x_1x_2, x_1x_2, \cdots, x_1x_n, \cdots, x_{n-1}x_n\}$$

where each new feature uses a pair of the original features. It can be observed that, the feature mapping leads to a huge number number of new features, such that i) computing the mapping itself can be inefficient, especially when the new feature space is of much higher dimension; ii) storing and utilizing the data samples in the new feature space can be expensive (e.g., we have to store all the high-dimensional images of the data samples and computing inner products in the high-dimensional feature space is of considerable overhead). Fortunately, the concept of kernels helps us avoid all these issues! With the help of kernels, the mapping does not have to be explicitly computed, and computations in the new high-dimensional feature space remains efficient.



(a) (b)

Figure 5: Feature mapping for 2-dimensional feature space.

Take the quadratic mapping as example again. Consider a 2-dimensional input space (i.e., the original feature space), we define a kernel function K that takes $x = (x_1, x_2)$ and $z = (z_1, z_2)$ as inputs

$$\begin{aligned} K(x,z) &= (x^T z)^2 \\ &= (x_1 z_1 + x_2 z_2)^2 \\ &= x_1^2 z_1^2 + x_2^2 z_2^2 + 2x_1 x_2 z_1 z_2 \\ &= (x_1^2, \sqrt{2}x_1 x_2, x_2^2)^T (z_1^2, \sqrt{2}z_1 z_2, z_2^2) \end{aligned} \tag{38}$$

It is demonstrated that the kernel function K implicitly defines a mapping

$$\phi(x) = \{x_1^2, \sqrt{2}x_1 x_2, x_2^2\}$$

Through the kernel function, when computing the inner product $< \phi(x), \phi(z) >$, we do not have to map x and z into the new higher-dimensional feature space first. Instead, $< \phi(x), \phi(z) >$ can be calculated in the original lower-dimensional input space. Formally speaking, each kernel K is associated with a feature mapping ϕ, which takes input $x \in \mathcal{X}$ (input space) and maps it to $\mathcal{F}$ (feature space). $\mathcal{F}$ needs to be a vector space with dot product defined, and is thus the so-called a *Hilbert space*. In another word, kernel $K(x,z) = \phi(x)^T \phi(z)$ takes two inputs and gives their similarity in $\mathcal{F}$ space

$$K : \mathcal{X} \times \mathcal{X} \to \mathbb{R} \tag{39}$$

The problem is, can any function be used as a kernel function? The answer is no. Kernel functions must satisfy *Mercer's Condition*. To introduce Mercer's condition, we need to define the quadratically integrable (or square integrable) function concept. A function $q : \mathbb{R}^n \to \mathbb{R}$ is square integrable if

$$\int_{-\infty}^{\infty} q^2(x)dx < \infty$$

A function $K(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ satisfies Mercer's condition if for any square integrable function $q(x)$, the following inequality holds for $\forall x, z \in \mathbb{R}^n$.

$$\int \int q(x)K(x,z)q(z)dxdz \geq 0$$

Let K_1 and K_2 be two kernel functions, then the following rules hold:

- Direct sum: $K(x,z) = K_1(x,z) + K_2(x,z)$
- Scalar product: $K(x,z) = \alpha K_1(x,z)$
- Direct product: $K(x,z) = K_1(x,z)K_2(x,z)$

Kernels can be constructed by composing these rules.

In SVM, Mercer's condition can be translated to another way to check whether K is a valid kernel. The kernel function K also defines the so-called *kernel matrix* over the data set (also denoted by K). Given m samples $\{x^{(1)}, x^{(2)}, \cdots, x^{(m)}\}$, the (i,j)-th entry of K is

$$K_{i,j} = K(x^{(i)}, x^{(j)}) = \phi(x^{(i)})^T \phi(x^{(j)})$$

If the matrix K is positive semi-definite, $K(\cdot,\cdot)$ is a valid kernel function.

Follows are some commonly used kernels:

- Linear (trivial) Kernal:
$$K(x,z) = x^T z$$
- Quadratic Kernel
$$K(x,z) = (x^T z)^2 \quad or \quad (1 + x^T z)^2$$
- Polynomial Kernel (of degree d)
$$K(x,z) = (x^T z)^d \quad or \quad (1 + x^T z)^d$$
- Gaussian Kernel
$$K(x,z) = \exp\left(-\frac{\|x-z\|^2}{2\sigma^2}\right)$$
- Sigmoid Kernel
$$K(x,z) = \tanh(\alpha x^T + c)$$

Overall, kernel $K(x,z)$ represents a dot product in some high-dimensional feature space $\mathcal{F}$

$$K(x,z) = (x^T z)^2 \quad \text{or} \quad (1 + x^T z)^2$$

Any learning algorithm in which data samples only appear as dot products ${x^{(i)}}^T x^{(j)}$ can be kernelized, by replacing ${x^{(i)}}^T x^{(j)}$ with $K(x^{(i)}, x^{(j)})$. Actually, most learning algorithms are like that, such as SVM, linear regression, etc. Many of the unsupervised learning algorithms (e.g., K-means clustering, Principal Component Analysis, etc.) can be kernelized too.

Recall that, the dual problem of SVM can be formulated as

$$\begin{aligned} \max_\alpha \quad & \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i,j=1}^m y^{(i)} y^{(j)} \alpha_i \alpha_j < x^{(i)}, x^{(j)} > \\ \text{s.t.} \quad & \sum_{i=1}^m y^{(i)} \alpha_i = 0 \\ & \alpha_i \geq 0, \quad \forall i = 1, 2, \cdots, m \end{aligned}$$

Replacing $< x^{(i)}, x^{(j)} >$ by $\phi(x^{(i)})^T \phi(x^{(j)}) = K(x^{(i)}, x^{(j)}) = K_{ij}$ gives us

$$\max_\alpha \quad \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j K_{i,j} \tag{40}$$

$$\text{s.t.} \quad \sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \tag{41}$$

$$\alpha_i \geq 0, \quad \forall i = 1, 2, \cdots, m \tag{42}$$

SVM now learns a linear separator in the kernel defined feature space $\mathcal{F}$, and this corresponds to a non-linear separator in the original space $\mathcal{X}$.

Supposing α_i^* $(i = 1, \cdots, m)$ is the optimal solution to the above optimization problem, we can define the linear decision boundary $\omega^{*T} \phi(x) + b^* = 0$ in the high-dimensional feature space. As what we have shown in Sec. 3, in the feature space, ω^* can be calculated by

$$\omega^* = \sum_{i:\alpha_i^* > 0} \alpha_i^* y^{(i)} \phi(x^{(i)}) \tag{43}$$

and b^* can be calculated according to any data sample i such that $\alpha_i^* > 0$ as follows

$$\begin{aligned} b^* &= y^{(i)} - \omega^{*T} \phi(x^{(i)}) \\ &= y^{(i)} - \sum_{j:\alpha_j^* > 0} \alpha_j^* y^{(j)} \phi^T(x^{(j)}) \phi(x^{(i)}) \\ &= y^{(i)} - \sum_{j:\alpha_j^* > 0} \alpha_j^* y^{(j)} K_{ij} \end{aligned} \tag{44}$$

Given a test data sample x, the prediction can be made by

$$\begin{aligned} y &= \text{sign}\left(\omega^{*T} \phi(x) + b^*\right) \\ &= \text{sign}\left(\sum_{i:\alpha_i^* > 0} \alpha_i^* y^{(i)} \phi^T(x^{(i)}) \phi(x) + b^*\right) \\ &= \text{sign}\left(\sum_{i:\alpha_i^* > 0} \alpha_i^* y^{(i)} K(x^{(i)}, x) + b^*\right) \end{aligned} \tag{45}$$

Kernelized SVM needs the support vectors in the test phase (except when you can write $\phi(x)$ as an explicit, reasonably-sized vector). In the unkernelized version, $\omega^* = \sum_{s \in \mathcal{S}} \alpha_s^* y^{(s)} x^{(s)}$ can be computed and stored as a n-dimensional vector, so the support vectors need not be stored.

5 Regularized SVM

We now introduce regularization to SVM. The regularized SVM is also called *Soft-Margin SVM*. In the regularized SVM, we allow some training examples to be misclassified, such that some training examples may fall within the margin region, as shown in Fig. 5. For the linearly separable case, the constraints are

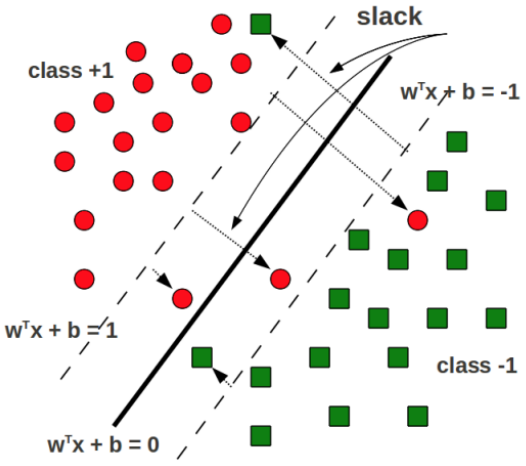


Figure 6: Regularized (Soft-Margin) SVM

$$y^{(i)}(\omega^T x^{(i)} + b) \geq 1$$

for $\forall i = 1, \cdots, m$, while in the non-separable case, we relax the above constraints as:

$$y^{(i)}(\omega^T x^{(i)} + b) \geq 1 - \xi_i$$

for $\forall i = 1, \cdots, m$, where ξ_i is called *slack variable*.

In the non-separable case, we allow misclassified training examples, but we would like the number of such training examples to be as small as possible, by minimizing the sum of the slack variables $\sum_i \xi_i$. We reformulating the SVM problem by introducing slack variables ξ_i

$$\min_{w,b,\xi} \quad \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \tag{46}$$

$$s.t. \quad y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \quad \forall i = 1, \cdots, m \tag{47}$$

$$\xi_i \geq 0, \quad \forall i = 1, \cdots, m \tag{48}$$

The parameter C is used to tune the trade-off between the following two goals: i) although small C implies that $\|\omega\|^2/2$ dominates such that large margins are preferred, this allows a potential large number of misclassified training examples; ii) large C means $C\sum_{i=1}^{m}\xi_i$ dominates such that the number of the misclassified examples is decreased at the expense of having a small margin.

The Lagrangian of the optimization problem (46)$\sim$(48) can be defined by

$$\mathcal{L}(\omega, b, \xi, \alpha, r) = \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i[y^{(i)}(\omega^T x^{(i)} + b) - 1 + \xi_i] - \sum_{i=1}^{m} r_i\xi_i$$

and according to the KKT conditions, we have

$$\bigtriangledown_{\omega}\mathcal{L}(\omega^*, b^*, \xi^*, \alpha^*, r^*) = 0 \quad \Rightarrow \quad \omega^* = \sum_{i=1}^{m} \alpha_i^* y^{(i)} x^{(i)} \tag{49}$$

$$\bigtriangledown_{b}\mathcal{L}(\omega^*, b^*, \xi^*, \alpha^*, r^*) = 0 \quad \Rightarrow \quad \sum_{i=1}^{m} \alpha_i^* y^{(i)} = 0 \tag{50}$$

$$\bigtriangledown_{\xi_i}\mathcal{L}(\omega^*, b^*, \xi^*, \alpha^*, r^*) = 0 \quad \Rightarrow \quad \alpha_i^* + r_i^* = C, \text{ for } \forall i \tag{51}$$

$$\alpha_i^*, r_i^*, \xi_i^* \geq 0, \text{ for } \forall i \tag{52}$$

$$y^{(i)}(\omega^{*T} x^{(i)} + b^*) + \xi_i^* - 1 \geq 0, \text{ for } \forall i \tag{53}$$

$$\alpha_i^*(y^{(i)}(\omega^{*T} x^{(i)} + b^*) + \xi_i^* - 1) = 0, \text{ for } \forall i \tag{54}$$

$$r_i^* \xi_i^* = 0, \text{ for } \forall i \tag{55}$$

We then formulate the corresponding dual problem as

$$\max_{\alpha} \quad \mathcal{J}(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j < x^{(i)}, x^{(j)} > \tag{56}$$

$$s.t. \quad 0 \leq \alpha_i \leq C, \quad \forall i = 1, \cdots, m \tag{57}$$

$$\sum_{i=1}^{m} \alpha_i y^{(i)} = 0 \tag{58}$$

We can use existing QP solvers to address the above optimization problem and calculate the optimal value of α.

Let α_i^* be the optimal values of α. According to KKT conditions, ω^* can be calculated by (49). We then show how to calculate the optimal value of b (denoted by b^*). Since $\alpha_i^* + r_i^* = C$ and $r_i^* \xi_i^* = 0$ holds for $\forall i$, we have

$$(C - \alpha_i^*)\xi_i^* = 0, \ \forall i$$

Hence, for $\forall i$ such that $\alpha_i^* \neq C$, we have $\xi_i^* = 0$ and thus

$$\alpha_i^*(y^{(i)}(\omega^{*T} x^{(i)} + b^*) - 1) = 0$$

according to (54). Moreover, since $\forall i$ such that $0 < \alpha_i^* < C$, we have

$$y^{(i)}(\omega^{*T} x^{(i)} + b^*) = 1$$

and thus

$$\omega^{*T} x^{(i)} + b^* = y^{(i)}$$

We finally calculate b^* according to any data sample i such that $0 < \alpha_i^* < C$ [2] as follows

$$b^* = y^{(i)} - \omega^{*T} x^{(i)} \tag{59}$$

To improve the precision of the numerical computations, we can calculate b^* by taking into account all data samples with $0 < \alpha_i^* < C$

$$b^* = \frac{\sum_{i:0<\alpha_i^*<C}(y^{(i)} - \omega^{*T} x^{(i)})}{\sum_{i=1}^{m} \mathbf{1}(0 < \alpha_i^* < C)} \tag{60}$$

[2] Such data samples are the support vectors in the soft-margin SVM.

6 Sequential Minimal Optimization Algorithm

For the optimization problem defined in (56)∼(58), we cannot directly apply coordinate descent algorithm, due to the equality constraint (58) where it is demonstrate that any variable α_i of α is determined by the others, i.e.

$$\alpha_i y^{(i)} = -\alpha_1 y^{(1)} - \alpha_2 y^{(2)} - \cdots - \alpha_{i-1} y^{(i-1)} - \alpha_{i+1} y^{(i+1)} - \cdots - \alpha_m y^{(m)}$$

Therefore, in the *Sequential Minimal Optimization* (SMO) algorithm, we optimize two of the variables at one time. We first summarize the general form of the SMO algorithm in **Algorithm** 1. The algorithm achieves the convergence

Algorithm 1: SMO algorithm

1: **Given** a starting point $\alpha \in$ **dom** $\mathcal{J}$
2: **repeat**
3: Select some pair of α_i and α_j to update next (using a heuristic that tries to pick the two α's);
4: Re-optimize $\mathcal{J}(\alpha)$ with respect to α_i and α_j, while holding all the other α_k's $(k \neq i, j)$ fixed
5: **until** convergence criterion is satisfied

if the outputs of the algorithm, i.e., α (and thus ω and b which are calculated according to α), satisfy all the KKT conditions. To verify if the KKT conditions holds for these parameters, we introduce some corollaries according to the KKT conditions.

Corollary 1. *For* $\forall i = 1, 2, \cdots, m$, *when* $\alpha_i^* = 0$, $y^{(i)}({\omega^*}^T x^{(i)} + b^*) \geq 1$.

Proof.

$$\begin{aligned}
&\because \alpha_i^* = 0, \alpha_i^* + r_i^* = C \quad (51)\\
&\therefore r_i^* = C\\
&\because r_i^* \xi_i^* = 0 \quad (55)\\
&\therefore \xi_i^* = 0\\
&\because y^{(i)}({\omega^*}^T x^{(i)} + b^*) + \xi_i^* - 1 \geq 0 \quad (53)\\
&\therefore y^{(i)}({\omega^*}^T x^{(i)} + b^*) \geq 1
\end{aligned}$$

□

Corollary 2. *For* $\forall i = 1, 2, \cdots, m$, *when* $\alpha_i^* = C$, $y^{(i)}({\omega^*}^T x^{(i)} + b^*) \leq 1$

Proof.

$$\begin{aligned}
&\because \alpha_i^* = C,\ \alpha_i^*(y^{(i)}({\omega^*}^T x^{(i)} + b^*) + \xi_i^* - 1) = 0 \quad (54)\\
&\therefore y^{(i)}({\omega^*}^T x^{(i)} + b^*) + \xi_i^* - 1 = 0\\
&\because \xi_i^* \geq 0 \quad (52)\\
&\therefore y^{(i)}({\omega^*}^T x^{(i)} + b^*) = 1 - \xi^* \leq 1
\end{aligned}$$

□

Corollary 3. *For $\forall i = 1, 2, \cdots, m$, when $0 < \alpha_i^* < C$, $y^{(i)}(\omega^{*T} x^{(i)} + b^*) = 1$.*

Proof.

$$
\begin{aligned}
&\because 0 < \alpha_i^* < C, \alpha_i^* + r_i^* = C \quad (51)\\
&\therefore 0 < r_i^* < C\\
&\because r_i^* \xi_i^* = 0 \ (55)\\
&\therefore \xi_i^* = 0\\
&\because 0 < \alpha_i^* < C, \alpha_i^*(y^{(i)}(\omega^{*T} x^{(i)} + b) + \xi_i^* - 1) = 0 \quad (54)\\
&\therefore y^{(i)}(\omega^{*T} x^{(i)} + b^*) + \xi_i^* - 1 = 0\\
&\therefore y^{(i)}(\omega^{*T} x^{(i)} + b^*) = 1
\end{aligned}
$$

□

Remarks: According to these corollaries, for $\forall i = 1, \cdots, m$, $x^{(i)}$ is i) correctly classified if $\alpha_i^* = 0$; ii) misclassified if $\alpha_i^* = C$; and iii) a support vector if $0 \leq \alpha_i^* \leq C$.

According to the KKT conditions and the corollaries, the SMO algorithm terminates when the following conditions hold (with some precision ϵ).

$$
\begin{aligned}
&\sum_{i=1}^{m} \alpha_i y^{(i)} = 0, \quad 0 \leq \alpha_i \leq C, \quad \forall i = 1, \cdots, m\\
&y^{(i)} \left(\sum_{j=1}^{m} \alpha_j y^{(j)} < x^{(i)}, x^{(j)} > -b \right) = \begin{cases} \geq 1, & \forall i : \alpha_i = 0 \\ = 1, & \forall i : 0 < \alpha_i < C \\ \leq 1, & \forall i : \alpha_i = C \end{cases}
\end{aligned}
$$

As shown in **Algorithm** 1, we have to tune two of the α_i's which do not respect the above conditions (and thus the KTT conditions). In the following, we take α_1 and α_2 for example to explain the optimization process of the SMO algorithm (i.e., Line 4 in **Algorithm** 1). By treating α_1 and α_2 as variables while the others as known quantities, the objective function (56) can be re-written as

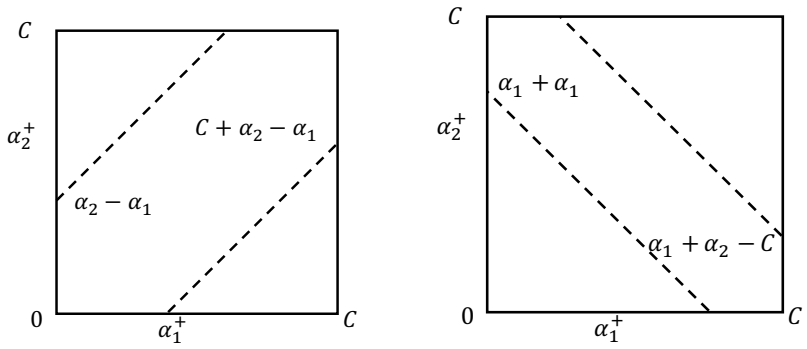
$$
\begin{aligned}
\mathcal{J}(\alpha_1^+, \alpha_2^+) \quad = \quad & \alpha_1^+ + \alpha_2^+ - \frac{1}{2} K_{11} {\alpha_1^+}^2 - \frac{1}{2} K_{22} {\alpha_2^+}^2 - S K_{12} \alpha_1^+ \alpha_2^+ \\
& - y^{(1)} V_1 \alpha_1^+ - y^{(2)} V_2 \alpha_2^+ + \Psi \qquad (61)
\end{aligned}
$$

where we denote by α_1^+ and α_2^+ the variables we try optimize in the current iteration (while treating α_1 and α_2 as the results in the last iteration) and

$$
\begin{cases}
K_{ij} = < x^{(i)}, x^{(j)} > \\
S = y^{(1)} y^{(2)} \\
\Psi = \sum_{i=3}^{m} \alpha_i - \frac{1}{2} \sum_{i=3}^{m} \sum_{j=3}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j K_{ij} \\
V_i = \sum_{j=3}^{m} y^{(j)} \alpha_j K_{ij}
\end{cases}
$$

According to the constraint (58), we can define

$$
\zeta = \alpha_1^+ y^{(1)} + \alpha_2^+ y^{(2)} = -\sum_{i=3}^{m} \alpha_i y^{(i)} = \alpha_1 y^{(1)} + \alpha_2 y^{(2)} \qquad (62)
$$



(a) $y^{(1)}y^{(2)} = -1$ (b) $y^{(1)}y^{(2)} = 1$

Figure 7: α_1^+ and α_2^+.

which confines the optimization to be on a line. Since $0 \leq \alpha_1, \alpha_2 \leq C$, we can derive a lower bound L and an upper bound H for them. As shown in Fig. 7(a), when $y^{(1)}y^{(2)} = -1$, we have

$$H = \min\{C, C + \alpha_2 - \alpha_1\} \text{ and } L = \max\{0, \alpha_2 - \alpha_1\} \tag{63}$$

When $y^{(1)}y^{(2)} = 1$ (as shown in Fig. 7(b)), H and L can be calculated as

$$H = \min\{C, \alpha_2 + \alpha_1\} \text{ and } L = \max\{0, \alpha_1 + \alpha_2 - C\} \tag{64}$$

Then, our problem can be formulated as

$$\begin{array}{ll} \max_{\alpha_2} & \mathcal{J}((\zeta - \alpha_2^+ y^{(2)})y^{(1)}, \alpha_2^+) \quad (65) \\ \text{s.t.} & L \leq \alpha_2^+ \leq H \quad (66) \end{array}$$

Therefore, we can find the extremum by letting the first derivative (with respect to α_2^+) to be zero as follows

$$\begin{aligned} & \frac{\partial}{\partial \alpha_2^+} f((\zeta - \alpha_2^+ y^{(2)})y^{(1)}, \alpha_2^+) \\ = \; & -S + 1 + SK_{11}(\zeta y^{(1)} - S\alpha_2^+) - K_{22}\alpha_2^+ - SK_{12}(\zeta y^{(1)} - S\alpha_2^+) \\ & + K_{12}\alpha_2^+ + y^{(2)}V_1 - y^{(2)}V_2 = 0 \end{aligned} \tag{67}$$

By assuming

$$E_i = \sum_{j=1}^{m} y^{(j)}\alpha_j K_{ij} + b - y^{(i)} \tag{68}$$

we then have

$$\begin{aligned}
& (K_{11} - 2K_{12} + K_{22})\alpha_2^+ \\
= \;& \zeta y^{(2)}(K_{11} - K_{12}) + y^{(2)}(V_1 - V_2) - S + 1 \\
= \;& y^{(2)}(y^{(1)}\alpha_1 + y^{(2)}\alpha_2)(K_{11} - K_{12}) \\
& + y^{(2)}\left(\sum_{i=1}^{m} y^{(i)}\alpha_i(K_{1i} - K_{2i}) - y^{(1)}\alpha_1(K_{11} - K_{12}) - y^{(2)}\alpha_2(K_{12} - K_{22})\right) \\
& - S + 1 \\
= \;& (S\alpha_1 + \alpha_2)(K_{11} - K_{12}) - S\alpha_1(K_{11} - K_{12}) \\
& - \alpha_2(K_{12} - K_{22}) + y^{(2)}(E_1 - E_2) \\
= \;& \alpha_2(K_{11} - 2K_{12} + K_{22}) + y^{(2)}(E_1 - E_2)
\end{aligned}$$

and thus

$$\alpha_2^+ = \alpha_2 + \frac{y^{(2)}(E_1 - E_2)}{K_{11} - 2K_{12} + K_{22}}$$

Since α_2^+ should be in the range of $[L, H]$,

$$\alpha_2^+ = \begin{cases} H, & \alpha_2^+ > H \\ \alpha_2^+, & L \le \alpha_2^+ \le H \\ L, & \alpha_2^+ < L \end{cases}$$

In each iteration, we have to update b accordingly so as to verify if the convergence criterion is satisfied. According to **Corollary** 3, when $0 < \alpha_1^+ < C$, we have

$$\begin{aligned}
b_1^+ &= y^{(1)} - \alpha_1^+ y^{(1)} K_{11} - \alpha_2^+ y^{(2)} K_{21} - \sum_{i=3}^{m} \alpha_i y^{(i)} K_{i1} \\
&= -E_1 + \alpha_1 y^{(1)} K_{11} + \alpha_2 y^{(2)} K_{21} + b - \alpha_1^+ y^{(1)} K_{11} - \alpha_2^+ y^{(2)} K_{21} \\
&= -E_1 - y^{(1)} K_{11}(\alpha_1^+ - \alpha_1) - y^{(2)} K_{21}(\alpha_2^+ - \alpha_2) + b
\end{aligned}$$

Similarly, when $0 < \alpha_2^+ < C$, another choice to compute b^+ is

$$b_2^+ = -E_2 - y^{(1)} K_{12}(\alpha_1^+ - \alpha_1) - y^{(2)} K_{22}(\alpha_2^+ - \alpha_2) + b$$

Therefore, when $0 < \alpha_1^+ < C$ and $0 < \alpha_2^+ < C$ both hold, we have $b_1^+ = b_2^+$ and we can choose one of them as b^+. When α_1^+ and α_2^+ are on the bound (i.e., $\alpha_1 = 0$ or $\alpha_1 = C$ and $\alpha_2 = 0$ or $\alpha_2 = C$), all values between b_1^+ and b_2^+ satisfy the KKT conditions and we can let $b^+ = (b_1^+ + b_2^+)/2$.