Support Vector Machines

September 23, 2020

Abstract

Topics: SVM and its primal problem; dual problem; non-separable case; kernel trick; SMO algorithm.

1 Background: geometry of Euclidean spaces

We define some geometric concepts in an Euclidean space $\mathbb{R}^n$. The length of a point or vector $\mathbf{x} \in \mathbb{R}^n$ is $\|\mathbf{x}\|_2$ or $\|\mathbf{x}\|$ by default. The distance between two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ is $\|\mathbf{x}_1 - \mathbf{x}_2\|$. The inner product between two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ is $\langle \mathbf{x}_1, \mathbf{x}_2 \rangle$ or $\mathbf{x}_1^\top \mathbf{x}_2$.

A hyperplane in $\mathbb{R}^n$ is a generalization of a straight line in the 2-dimensional Euclidean space, defined by the set of vectors $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{w}^\top \mathbf{x} + b = 0\}$, where $\mathbf{w}$ is the normal direction of the hyperplane (a vector perpendicular to the plane) and $b/\|\mathbf{w}\|$ is the distance from the origin to the plane. A hyperplane can be identified by the tuple $(\mathbf{w}, b)$. A half-space derived from a hyperplane $(\mathbf{w}, b)$ is the set of vectors $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{w}^\top \mathbf{x} + b \geq 0\}$ or $\{\mathbf{x} \in \mathbb{R}^n : \mathbf{w}^\top \mathbf{x} + b \leq 0\}$.

A set $A \in \mathbb{R}^n$ is called convex, if for any two elements $\mathbf{x}_1$ and $\mathbf{x}_2$ in the set, and for any real number $0 \leq \lambda \leq 1$, $\lambda \mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2$ also in A. That is, if the straight line that connects any two vectors in A lies entirely in A, then A is a convex set. For example, the hyperplane $A = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{w}^\top \mathbf{x} + b = 0\}$ is a convex set. A half space is also a convex set. The unit sphere $\{\mathbf{x} : \|\mathbf{x}\| = 1\}$ is not a convex set. A function f defined on a convex set A is convex if $f(\lambda \mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1 - \lambda) f(\mathbf{x}_2)$ for any $\mathbf{x}_1$, $\mathbf{x}_2$ in A.

2 SVM in the primal: a large margin classifier

Given training samples $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^m$, let the label $y^{(i)} \in \{-1, 1\}$[1] Assume that the data are linearly separable. That is, there is a hyperplane $(\mathbf{w}, b)$ such that

$$\begin{cases} \mathbf{w}^\top \mathbf{x}^{(i)} + b < 0 & y^{(i)} = -1 \\ \mathbf{w}^\top \mathbf{x}^{(i)} + b > 0 & y^{(i)} = 1. \end{cases} \tag{1}$$

Alternatively, we can say $y^{(i)}(\mathbf{w}^\top \mathbf{x}^(i) + b) > 0$. Then there can be multiple such hyperplanes. Which one should one choose to classify test data? Figure 1 shows the situation.

Another motivation of using a large margin classifier is to obtain confident classications. The probability that $\mathbf{x}$ is in class 1 can be obtained by the sigmoid function $\sigma(\mathbf{w}^\top \mathbf{x} + b)$ and if $\mathbf{w}^\top \mathbf{x} + b$ is much larger than 0, the probability is close to 1, giving a high confidence level that $\mathbf{x}$ belongs to class 1. Similarly, if $\mathbf{w}^\top \mathbf{x} + b$ is much less than 0, then $\sigma(\mathbf{w}^\top \mathbf{x} + b)$ will be close to 0 and give a confidence prediction that $\mathbf{x}$ is in class 0.

The quantity $\hat{\gamma}^{(i)} = y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 0$ is the "functional margin" of classifying the training example $(\mathbf{x}^{(i)}, y^{(i)})$, as it is evaluated based on the function $h(\mathbf{x}; \mathbf{w}, b) = \mathbf{w}^\top \mathbf{x} + b$. Note that the functional margin is non-negative since we assume that the training data are linearly separable. The larger the functional margin, the more confidence the classification would be. We want the

[1] We let -1, rather than 0, denote the negative class label for mathmatical convenience.

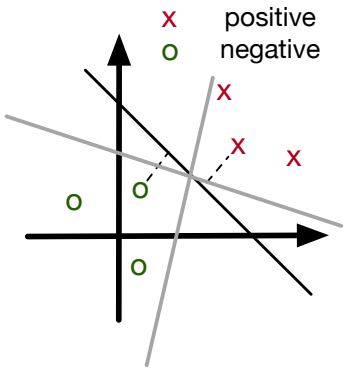

Figure 1: Multiple hyperplanes can classify the training data into two classes, but which one will perform better on the unseen test data? The solid line is the large margin classifier that has a lower probability to make mistakes on the unseen test data.

functional margins of all training examples to be large and let the smallest functional margin be $\hat{\gamma} = \min_{1 \le i \le m} \hat{\gamma}^{(i)}$. The following optimization problem is for learning a large margin classifier:

$$\begin{aligned} &\max_{\mathbf{w},b,\hat{\gamma}} && \hat{\gamma} \\ &\text{s.t.} && y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \ge \hat{\gamma}, i = 1, \ldots, m. \end{aligned} \tag{2}$$

Note that although the term $\mathbf{w}$ and b does not appear in the objective function, but they influence the objective through the inequality constraints.

However, one can scale $\mathbf{w}$ and b by a factor of $\kappa > 1$ so that the functional margin will be enlarged without changing the hyperplane geometricaly ($\kappa\mathbf{w}$ points in the same direction as $\mathbf{w}$ does, and distance from the hyperplane to the origin remains $b/\|\mathbf{w}\| = \kappa b/\kappa\|\mathbf{w}\|$). This cause the issue of hyperplane identification. To eliminate such ambiguity, fix $\|\mathbf{w}\| = 1$ so that $\gamma = \hat{\gamma}/\|\mathbf{w}\|$, leading to the following optimization problem:

$$\begin{aligned} &\max_{\mathbf{w},b,\gamma} && \gamma \\ &\text{s.t.} && y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \ge \gamma, i = 1, \ldots, m, \\ & && \|\mathbf{w}\| = 1. \end{aligned} \tag{3}$$

The term $\gamma = \hat{\gamma}/\|\mathbf{w}\|$ is the minimum of the m "geometric margins" $\gamma^{(i)} = \frac{y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)}{\|\mathbf{w}\|}$. Figure 2 shows the geometry of the derivation. Let the projection of $\mathbf{x}^{(i)}$ onto the hyperplane be $\text{proj}(\mathbf{x}^{(i)})$. Then the distance $\|\mathbf{x}^{(i)} - \text{proj}(\mathbf{x}^{(i)})\|$ is the geometric margin $\gamma^{(i)}$ since $\text{proj}(\mathbf{x}^{(i)}) + \gamma^{(i)}\frac{\mathbf{w}}{\|\mathbf{w}\|} = \mathbf{x}^{(i)}$ and $\gamma^{(i)}$ scales the unit length vector $\frac{\mathbf{w}}{\|\mathbf{w}\|}$ to obtain the vector $\mathbf{x}^{(i)} - \text{proj}(\mathbf{x}^{(i)})$.

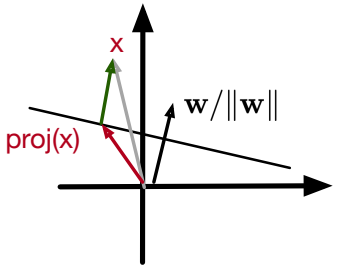

Figure 2: $\mathbf{x}$ is a training example, $\text{proj}(\mathbf{x})$ is the projection of $\mathbf{x}$ onto the hyperplane, and $\mathbf{w}/\|\mathbf{w}\|$ is the unit length vector perpendicular to the hyperplane. The geometric margin of classifying $\mathbf{w}$ using $\mathbf{w}$ is the length of the green arrow, which is in parallel to $\mathbf{w}/\|\mathbf{w}\|$.

The feasible solution $(\mathbf{w}, b)$ of the above optimization problem must satisfy the m inequality constraints and the equality constraint $\|\mathbf{w}\| = 1$. The m inequality constriants define m halfspaces

and satisfying all m constraints means that the solution is in the intersection of these halfspaces, which is a convex set. However, the equality constraint defines a unit hyper-sphere (the surface of the unit ball without the interior of the ball) in the n-dimensional space. The hyper-sphere is not convex (try linking two different points on the sphere using a straight line without leaving the sphere).

Lemma 15.2 of the textbook UML[2] shows the above optimization problem is equivalent to the following one:

$$\begin{aligned} &\max_{\mathbf{w},b} && \frac{1}{\|\mathbf{w}\|} \\ &\text{s.t.} && y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1, i = 1, \ldots, m, \end{aligned} \tag{4}$$

or this one:

$$\begin{aligned} &\min_{\mathbf{w},b} && \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{s.t.} && y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1, i = 1, \ldots, m, \end{aligned} \tag{5}$$

since $f(x) = x^2$ is a monotonically increasing function for $x > 0$. We want to have the square of $\|\mathbf{w}\|$ ($\|\mathbf{w}\|^2$) rather than the norm $\|\mathbf{w}\|$ in the objective function, $\|\mathbf{w}\|^2$ is a convex objective function, while $\|\mathbf{w}\|$ is a concave function. The term $\|\mathbf{w}\|^2$ also makes deriving the dual problem (to be defined later) easier and helps introduce the kernel trick. Along with the convex intersection of m half-spaces, the optimization is convex. The optimization problem is called the primal problem of SVM and the parameters $(\mathbf{w}, b)$ are called the "primal" variables, in contrast to the dual problem and dual variables to be motivated and defined next.

3 SVM in the dual

To gain more insight of SVM (sparsity in the model and kernel trick), we find the dual problem of the primal problem.

3.1 Background: constrained optimization and Lagrangian methods

For a light treatment of this background, refer to Appendix E of PRML.

Unlike gradient descent for minimizing *unconstrained* negative log-likelihood loss function as when training logistic regression, the SVM optimization problem is a minimization problem, constrained by m inequalities derived from m training examples. The primal problem can be solved using stochastic gradient descent (Section 15.5 of UML). With additional constraints, special care must be taken in gradient descent, since a descent step may move the parameters outside of the feasible region, defined as the set of parameter values that satisfy the constraints. We care about the following "primal" optimization problem that encompasses the SVM primal as a special case:

$$\begin{aligned} &\min_{\mathbf{w} \in \mathbb{R}^n} && f(\mathbf{w}) \\ &\text{s.t.} && g_i(\mathbf{w}) \geq 0, i = 1, \ldots, m, \end{aligned} \tag{6}$$

where m is the number of inequality constraints g_i. $f(\mathbf{w})$ is a convex function in $\mathbf{w}$, such as $\frac{1}{2}\|\mathbf{w}\|^2$ in SVM. $g_i(\mathbf{w})$ are concave functions in $\mathbf{w}$ for all i, so that $-g_i(\mathbf{w})$ are convex functions. For SVM, $g_i(\mathbf{w}) = y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) - 1$ is linear (and thus both convex and concave) in $\mathbf{w}$.

3.1.1 Simple motivating examples

Lagrangian method is invented for constrained optimization problems. To motivate the method, let's look at the following tiny optimization problem in the 2-dimensional Euclidean space $\mathbb{R}^2$ with

[2] Downloadable:https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-theory-algorithms.pdf

a single equality constraint[3]:

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^2} \quad & f(\mathbf{w}) = \tfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & g(\mathbf{w}) = w_1 w_2 - 1 = 0. \end{aligned} \tag{7}$$

See Figure 3 for a visualization of the objective function and the constraint in the space $\mathbf{w} = [w_1, w_2]$. At the minimizer $\mathbf{w}^*$ of the objective function $f(\mathbf{w})$, the hyperbola touch the coutour when $\|\mathbf{w}\| = 1$, but how can one find $\mathbf{w}^*$.

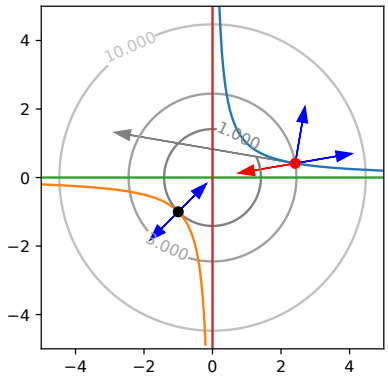

Figure 3: This figure shows the contour lines of the objective function $\frac{1}{2}\|\mathbf{w}\|$. The equality constraint generates two hyperbola as the feasible region. At the stationary point in the quadrant of $\{\mathbf{w} : w_1 < 0, w_2 < 0\}$, the two gradients ∇f and ∇g are parallel. In the quadrant $\{\mathbf{w} : w_1 > 0, w_2 > 0\}$, the projection of $-\nabla f$ on the the tangent line of $g(\mathbf{w})$ at the red point is a non-zero vector, indicating that $f(\mathbf{w})$ can be further reduced by moving the red point along the hyperbola in the direction of the projection with infinitestimal step size.

It turns out that some geometric conditions can help us locate $\mathbf{w}^*$. The normal direction to the hyperbola at any point in $\mathbb{R}^2$, say $\mathbf{w}^0 = [w_1^0, w_2^0]^\top$, can be computed as

$$\nabla g(\mathbf{w})|_{\mathbf{w}^0} = \frac{\partial g(\mathbf{w})}{\partial \mathbf{w}}|_{\mathbf{w}^0} = [w_2^0, w_1^0]^\top, \tag{8}$$

and the normal direction to the contour $\{\mathbf{w} : f(\mathbf{w}) = c\}$ for any constant c at $\mathbf{w}^0$ is

$$\nabla f(\mathbf{w})|_{\mathbf{w}^0} = \frac{\partial f(\mathbf{w})}{\partial \mathbf{w}}|_{\mathbf{w}^0} = \frac{1}{2}\frac{\partial \mathbf{w}^\top \mathbf{w}}{\partial \mathbf{w}}|_{\mathbf{w}^0} = [w_1^0, w_2^0]^\top. \tag{9}$$

Observe that at a point where ∇f is not parallel to ∇g (such as the red point in Figure 3), the projection of ∇f onto the vector $\nabla g^\perp$ that is perpendicular to ∇g (or tangent to the line $g(\mathbf{w}) = c$ at $\mathbf{w}^0$) is a non-zero vector:

$$\nabla f = \lambda \nabla g + \mu \nabla g^\perp, \quad \mu \neq 0. \tag{10}$$

Since $\langle \nabla g, \nabla g^\perp \rangle = 0$, we have

$$\langle \nabla f, \nabla g^\perp \rangle = \lambda \langle \nabla g, \nabla g^\perp \rangle + \mu \langle \nabla g^\perp, \nabla g^\perp \rangle = \mu \|\nabla g^\perp\|^2, \tag{11}$$

so that $\mu = \langle \nabla f, \nabla g^\perp \rangle / \|\nabla g^\perp\|^2$. By moving from $\mathbf{w}^0$ to a ponit $\mathbf{w}^1$ in the direction (or the opposite direction) of $\nabla g^\perp$ with an infinitesimal step size so that $\mathbf{w}^1$ is still on the hyperbola $g(\mathbf{w}) = c$, we obtain a new solution $\mathbf{w}^1$ that further reduce $f(\mathbf{w}^0)$ to $f(\mathbf{w}^1)$ while staying on the hyperbola[4].

At the point $\mathbf{w}^*$ where ∇g and ∇f are parallel (such as the black point in Figure 3), the projection of ∇f onto $\nabla g^\perp$ is zero and no movement of $\mathbf{w}^*$ can reduce the value $f(\mathbf{w})$. The point

[3] You will see that this is a special case for the more general case with inequality constraint $g(\mathbf{w}) \geq 0$

[4] A formal proof of this will need numerical optimization theory.

$\mathbf{w}^*$ is called a *stationary point*. We obtain the important condition of the partial derivatives of f and g at a stationary point $\mathbf{w}^*$:

$$\nabla_{\mathbf{w}} f|_{\mathbf{w}^*} = \lambda \nabla_{\mathbf{w}} g|_{\mathbf{w}^*}. \tag{12}$$

Definition 3.1 *The parameter $\lambda \in \mathbb{R}$ is the dual variable for the primal problem in Eq. (7).*

The above example does not include inequality constraint like the SVM primal problem. Let's look at a second example with an inequality constraint:

$$\begin{array}{ll} \min_{\mathbf{w}\in\mathbb{R}^2} & f(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} & g(\mathbf{w}) = w_1 + w_2 - c \geq 0 \end{array} \tag{13}$$

There are two possible cases for the location of the optimal solution.

- When the half-space $g(\mathbf{w}) = w_1 + w_2 - c \geq 0$ includes the origin (e.g., when $c = -3$), the optimal solution is obviously $\mathbf{w}^* = [0, 0]^\top$. This solution is called "non-binding" as it is not touching the boundary of the hyperplane that defines the half-space, so that $\mathbf{w}^*$ satisfies $g(\mathbf{w}^*) > 0$ (the inequality constraint is inactive). This is like an unconstrained optimization problem and one can solve for $\mathbf{w}^*$ by solving the equation $\nabla f(\mathbf{w}) = 0$.
- When the half-space $g(\mathbf{w}) = w_1 + w_2 - c \geq 0$ does not include the origin (e.g., when $c = 1$), the optimal solution $\mathbf{w}^*$ is located at where the contour of $f(\mathbf{w})$ touches the hyperplane (called a "binding" stationary point). This is similar to the optimization problem with equality constraint since $\mathbf{w}^*$ satisfies $g(\mathbf{w}^*) = c$ (the inequality constraint is active).

  At this point, we must have $\nabla f(\mathbf{w}) = \lambda \nabla g(\mathbf{w})$ for some $\lambda \geq 0$ (the dual variable for the optimization problem Eq. (13)). λ must be non-negative since the gradient ∇f goes in the same direction of ∇g.

We combine the above two cases into the following equation (called "first-order condition"):

$$\nabla f(\mathbf{w}) = \lambda \nabla g(\mathbf{w}), \quad \lambda \geq 0. \tag{14}$$

Case 1 is when $\lambda = 0$ (the inequality constraint is inactive) and case 2 is when $\lambda > 0$ (the inequality constraint is active).

In both cases, we also have the so-called "complementary slackness" condition:

$$\lambda g(\mathbf{w}) = 0. \tag{15}$$

In case 1, $\lambda = 0$ so the constraint $g(\mathbf{w}) \geq 0$ is inactive and $g(\mathbf{w})$ is free to take value 0 or a positive number; in case 2, $\lambda \neq 0$ and the constraint $g(\mathbf{w}) \geq 0$ is active and $g(\mathbf{w})$ is forced to take value 0.

Combining the first-order condition, the complementary slackness, the primal and dual constraints, we have the following KKT (Karush-Kuhn-Tucker) conditions:

$$\begin{aligned} \nabla_{\mathbf{w}} f(\mathbf{w}) &= \lambda \nabla_{\mathbf{w}} g(\mathbf{w}), && (16) \\ \lambda &\geq 0, && (17) \\ g(\mathbf{w}) &\geq 0, && (18) \\ \lambda g(\mathbf{w}) &= 0. && (19) \end{aligned}$$

The subscript $\mathbf{w}$ in $\nabla f(\mathbf{w})$ and $\nabla g(\mathbf{w})$ is to stress that the partial derivative is taken with respect to $\mathbf{w}$, not λ.

We define the Lagrangian function as

$$L(\mathbf{w}, \lambda) = f(\mathbf{w}) - \lambda g(\mathbf{w}), \tag{20}$$

(No need to understand this: $L(\mathbf{w}, \lambda)$ is a convex function in $\mathbf{w}$, if $f(\mathbf{w})$ is convex and $g(\mathbf{w})$ is concave (so that $-\lambda g(\mathbf{w})$ is convex).)

Using the Lagrangian function, the first-order condition Eq.(31) can be re-written as

$$\nabla_{\mathbf{w}} f(\mathbf{w}) - \lambda \nabla_{\mathbf{w}} g(\mathbf{w}) = \nabla_{\mathbf{w}} L(\mathbf{w}, \lambda) = 0. \tag{21}$$

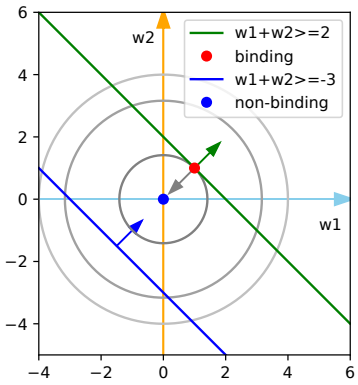

Figure 4: This figure shows the contour lines of the objective function $\frac{1}{2}\|\mathbf{w}\|^2$. There are two optimization problems, each with an inequality constraint $g(\mathbf{w}) = w_1 + w_2 - c \geq 0$. Two half-spaces as the feasible regions are shown with normal directions attached. The two inequality constraints lead to one inactive constraint and one active constraint in the two optimization problems, respectively.

A stationary point $\mathbf{w}^* \in \mathbb{R}^n$ and $\lambda^* \in \mathbb{R}$ of an inequality-constrained optimization problem can be found by solving the $n+1$ equations Eq. (31) and Eq. (25). Continuing the second constrained optimization example, the KKT condition is

$$\begin{aligned} \nabla_{\mathbf{w}} f(\mathbf{w}) = \mathbf{w} &= \lambda[1,1]^\top = \lambda \nabla_{\mathbf{w}} g(\mathbf{w}) && (22)\\ \lambda &\geq 0, && (23)\\ g(\mathbf{w}) = w_1 + w_2 - c &\geq 0, && (24)\\ \lambda g(\mathbf{w}) = \lambda(w_1 + w_2 - c) &= 0. && (25) \end{aligned}$$

The first-order condition says that $w_1 = w_2$ and they are $[1,1]^\top$ scaled by the dual variable λ. Depending on the value of c, if the inequality constraint is inactive ($w_1 + w_2 - c > 0$), then $\lambda = 0$ and one stationary point is $\mathbf{w}^* = \lambda[1,1]^\top = [0,0]^\top$, which is shown as the blue point at the origin in Figure 4; if the inequality constraint is active ($w_1 + w_2 - c = 0$), then a stationary point is $[c/2, c/2]$ (one such point is shown as the red dot in Figure 4).

So why the definition of the Lagrangian function? Besides representing the first-order condition in the KKT conditions more compactly, the function allows us to find the dual problem of the primal problem Eq. (6) In most real-world cases, the KKT equations are too large and complicated to be solved analytically. The "dual" problem may allow some numerical optimization algorithm to find a stationary point more efficiently. The dual problem also leads to an infinitely dimensional data representation via the "kernel trick".

3.1.2 Dual problems

We go back to the more general primal Eq. (6) with m constraints. Let $\lambda_i \geq 0$ be the dual variable corresponding to the constraint $g_i(\mathbf{w}) \geq 0$ and $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_m]^\top \succeq \mathbf{0}$ be the vector of all λ's. Accordingly, let $\mathbf{g}(\mathbf{w}) = [g_1(\mathbf{w}), \ldots, g_m(\mathbf{w})]^\top$ be g_i, $i = 1, \ldots, m$, stacked in a column vector.

Definition 3.2 *The Lagrangian function of the primal problem Eq. (6) is*

$$L(\mathbf{w}, \boldsymbol{\lambda}) = f(\mathbf{w}) - \sum_{i=1}^{m} \lambda_i g_i(\mathbf{w}) = f(\mathbf{w}) - \boldsymbol{\lambda}^\top \mathbf{g}(\mathbf{w}), \quad \boldsymbol{\lambda} \succeq 0. \tag{26}$$

A possibly more efficient strategy to solve the primal problem Eq. (6) could be to formulate and solve the corresponding *dual problem* to obtain the stationary point $\boldsymbol{\lambda}^*$ in the dual space, and then recovery the stationary point $\mathbf{w}^*$ in the primal space.

Definition 3.3 *The dual objective function is defined as*

$$\tilde{L}(\boldsymbol{\lambda}) = \min_{\mathbf{w} \in \mathbb{R}^n} L(\mathbf{w}, \boldsymbol{\lambda}) \tag{27}$$

for any $\boldsymbol{\lambda} \succeq 0$. *That is, fix any non-negative vector* $\boldsymbol{\lambda}$, *search for a* $\mathbf{w}$ *that minimizes the Lagrangian function* $L(\mathbf{w}, \boldsymbol{\lambda})$.

The primal objective function $f(\mathbf{w})$, the dual objective function $\tilde{L}(\boldsymbol{\lambda})$, and the Lagrangian function $L(\mathbf{w}, \boldsymbol{\lambda})$ are connected in the Figure 6.

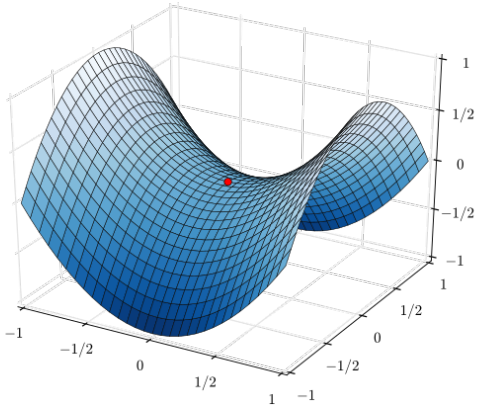

Figure 5: The surface is the values of the Lagrangian function $L(w, \lambda)$ for scalars $w \in \mathbb{R}$ and $\lambda \in \mathbb{R}$. The red dot is the saddle point or stationary point where $\partial_w L = \partial_\lambda L = 0$. By fixing a λ value and minimizing $L(w, \lambda)$ with respect to w, we obtain a point on the surface. By tracing this point along the λ axis, we can draw a concave function $\tilde{L}(\lambda) = \max_w L(w, \lambda)$.

No need to prove the following listed items, which are listed only for completeness. Properties of the dual objective function:

- $\tilde{L}(\boldsymbol{\lambda})$ is a concave function in $\boldsymbol{\lambda}$ in the domain $\boldsymbol{\lambda} \succeq \mathbf{0} : \tilde{L}(\ \boldsymbol{\lambda}) \geq -\infty\}$.
- (Weak duality) For any fixed $\boldsymbol{\lambda} \succeq \mathbf{0}$, $\tilde{L}(\ \boldsymbol{\lambda}) = \min_{\mathbf{w}} L(\mathbf{w},\ \boldsymbol{\lambda}) \leq L(\mathbf{w}, \boldsymbol{\lambda}) = f(\mathbf{w}) - \boldsymbol{\lambda}^\top \mathbf{g}(\mathbf{w}) \leq f(\mathbf{w})$ for any $\mathbf{w}$, since $\boldsymbol{\lambda} \geq 0$ and $\mathbf{g}(\mathbf{w}) \geq 0$. This implies that

$$\max_{\boldsymbol{\lambda} \succeq \mathbf{0}} \tilde{L}(\boldsymbol{\lambda}) \leq \min_{\mathbf{w} \in \mathbb{R}^n} f(\mathbf{w}). \tag{28}$$

- (Strong duality) When

$$\max_{\boldsymbol{\lambda} \succeq \mathbf{0}} \tilde{L}(\boldsymbol{\lambda}) = \min_{\mathbf{w} \in \mathbb{R}^n} f(\mathbf{w}), \tag{29}$$

  we have strong duality and

$$\max_{\boldsymbol{\lambda} \succeq \mathbf{0}} \min_{\mathbf{w} \in \mathbb{R}^n} L(\mathbf{w}, \boldsymbol{\lambda}) = \min_{\mathbf{w} \in \mathbb{R}^n} \max_{\boldsymbol{\lambda} \succeq \mathbf{0}} L(\mathbf{w}, \boldsymbol{\lambda}). \tag{30}$$

The following KKT conditions point to a connection between the optimal solution $\boldsymbol{\lambda}^*$ to the dual problem and the optimal solution $\mathbf{w}^*$ to the primal problem, stated in Theorem **??**.

Definition 3.4 *The KKT conditions for the primal problem Eq.(6) are*

$$\begin{aligned}
\nabla_{\mathbf{w}} f(\mathbf{w}) &= \sum_{i=1}^{m} \lambda_i \nabla_{\mathbf{w}} g_i(\mathbf{w}), && (31)\\
\boldsymbol{\lambda} &\succeq \mathbf{0}, && (32)\\
\mathbf{g}(\mathbf{w}) &\succeq \mathbf{0}, && (33)\\
\lambda_i g_i(\mathbf{w}) &= 0, \quad i = 1, \ldots, m. && (34)
\end{aligned}$$

Definition 3.5 *The dual (optimization) problem is defined as*

$$\begin{aligned}
\max \quad & \tilde{L}(\boldsymbol{\lambda}) \\
\text{s.t.} \quad & \boldsymbol{\lambda} \geq \mathbf{0}.
\end{aligned} \tag{35}$$

Theorem 3.6 *If $\mathbf{w}^*$ is the optimal solution to the primal problem Eq. (6). Besides, assume that $f(\mathbf{w})$ and $-g_i(\mathbf{w})$ are convex and differentiable at $\mathbf{w}^*$ Then any $\boldsymbol{\lambda}^*$ that satisfies the above KKT conditions is an optimal solution of the dual problem Eq. (35).*

With strong duality, the solution λ^* of the dual Eq. (35) satisfies $\tilde{L}(\lambda^*) = f(\mathbf{w}^*)$, so that by maximizing the concave function $\tilde{L} = \min_{\mathbf{w}\in\mathbb{R}^n} L(\mathbf{w}, \boldsymbol{\lambda})$ over $\boldsymbol{\lambda} \succeq \mathbf{0}$,

$$\tilde{L}(\boldsymbol{\lambda}^*) = \max_{\boldsymbol{\lambda}\succeq\mathbf{0}} \left\{ \min_{\mathbf{w}\in\mathbb{R}^n} L(\mathbf{w}, \boldsymbol{\lambda}) \right\}, \tag{36}$$

one can find the solution to the primal problem Eq. (6).

3.2 Dual problem for SVM

Equipped with the background, we use the Lagrangian method to find the dual of SVM. The primal problem of SVM can be re-written using the following inequality-constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & f(\mathbf{w}) = \tfrac{1}{2}\|\mathbf{w}\|^2 \\ \text{s.t.} \quad & g_i(\mathbf{w}) = y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) - 1 \geq 0, i = 1, \ldots, m. \end{aligned} \tag{37}$$

The primal variables are the parameters $\mathbf{w}$ and b. The m inequality constraints need to be satisfied simultaneously and thus define the intersection of m hyperplanes. With Lagrangian multipliers (or dual variables) α_i, $i = 1, \ldots, m$, we have the following Lagrangian function:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = f(\mathbf{w}) - \sum_{i=1}^{m} \alpha_i g_i(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i (y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) - 1). \tag{38}$$

Let $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m] \succeq \mathbf{0}$, meaning that $\boldsymbol{\alpha}$ is 'element-wisely greater than ($\succeq$) the all-zero vector $\mathbf{0}$ ($\alpha_i \geq 0$ for any i). the dual objective function $\tilde{L}(\boldsymbol{\alpha}) \triangleq \min_{\mathbf{w}\in\mathbb{R}^n, b} L(\mathbf{w}, b, \boldsymbol{\alpha})$ and the dual problem is

$$\max \quad \tilde{L}(\boldsymbol{\alpha}) \tag{39}$$

$$\text{s.t.} \quad \boldsymbol{\alpha} \succeq \mathbf{0} \tag{40}$$

The KKT conditions of the above primal problem are

$$\nabla_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i \nabla_{\mathbf{w}} [y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) - 1] = 0 \tag{41}$$

$$-\sum_{i=1}^{m} \alpha_i \nabla_b [y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) - 1] = 0 \tag{42}$$

$$\alpha_i \geq 0 \tag{43}$$

$$y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1, \quad i = 1, \ldots, m, \tag{44}$$

$$\alpha_i (y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) - 1) = 0 \quad i = 1, \ldots, m. \tag{45}$$

We can find the stationary point $(\mathbf{w}^*, b^*)$ that satisfies the KKT conditions as the solution to dual problem of SVM. But since the optimization problems satisifies the strong duality

$$\min_{\mathbf{w},b} \max_{\boldsymbol{\alpha}\succeq\mathbf{0}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}\succeq\mathbf{0}} \min_{\mathbf{w},b} L(\mathbf{w}, b, \boldsymbol{\alpha}), \tag{46}$$

the solution $\boldsymbol{\alpha}^*$ to the dual problem finds us the solution $\mathbf{w}^*$ to the primal problem.

3.2.1 A game theory interpretation

Consider the following optimization problem as a game between two players Alice and Bob.

$$\min_{\mathbf{w},b} \max_{\boldsymbol{\alpha}\succeq\mathbf{0}} L(\mathbf{w}, b, \boldsymbol{\alpha}) = \min_{\mathbf{w},b} \max_{\boldsymbol{\alpha}\succeq\mathbf{0}} \left\{ \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i (y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) - 1). \right\} \tag{47}$$

Alice controls the parameters $(\mathbf{w}, b)$ and Bob controls $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_m]^\top$. For any value $(\mathbf{w}, b)$ that Alice chooses, Bob plug that $(\mathbf{w}, b)$ into the function $L(\mathbf{w}, b, \boldsymbol{\alpha})$ and chooses the best $\boldsymbol{\alpha}$ that maximizes L given the current $(\mathbf{w}, b)$ value. The opportunity for Bob to make $L(\mathbf{w}, b, \boldsymbol{\alpha}) = \infty$ is when Alice chooses a $(\mathbf{w}, b)$ that misclassifies a training example:

$$y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) - 1) < 0, \tag{48}$$

so that the corresponding α_i is chosen to be ∞ to make $-\alpha_i y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) - 1) = \infty$. Alice is aware of the game that Bob is playing, so Alice is smart to pick $(\mathbf{w}, b)$ carefully so that maximum possible value of L achieved by Bob is minimized (and $< \infty$).

3.2.2 Deriving the dual SVM

Let's find the dual objective function $\tilde{L}(\boldsymbol{\alpha})$ first. Fixing any $\boldsymbol{\alpha}$, the function $L(\mathbf{w}, b, \boldsymbol{\alpha})$ is convex in the primal variables $(\mathbf{w}, b)$, since $\frac{1}{2}\|\mathbf{w}\|^2$ is convex and $-\alpha_i y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) - 1)$ are linear (and thus convex) in $(\mathbf{w}, b)$ for all $i = 1, \ldots, m$, and the sum of multiple convex functions is convex. Minimizing $L(\mathbf{w}, b, \boldsymbol{\alpha})$ with respect to $\mathbf{w}$ and b can be done by setting

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y^{(i)} \mathbf{x}^{(i)} = \mathbf{0}, \tag{49}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = \sum_{i=1}^{m} \alpha_i y^{(i)} = 0. \tag{50}$$

Note that $\mathbf{0}$ is an all-zero vector and 0 is a scalar.

As a result, $\mathbf{w} = \sum_{i=1}^{m} \alpha_i y^{(i)} \mathbf{x}^{(i)}$, which is a linear combination of the training feature vectors. The condition $\sum_{i=1}^{m} \alpha_i y^{(i)} = 0$ is called the "balancing" equation: the labels $y^{(i)} \in \{-1, +1\}$ are weighted by α_i so that $\sum_{i:y^{(i)}=1} \alpha_i = \sum_{i:y^{(i)}=-1} \alpha_i$.

Plugging $\mathbf{w} = \sum_{i=1}^{m} \alpha_i y^{(i)} \mathbf{x}^{(i)}$ in $L(\mathbf{w}, b, \boldsymbol{\alpha})$ and make use of the balancing equation, we obtain

$$\frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i (y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) - 1) \tag{51}$$

$$= \frac{1}{2}\langle \mathbf{w}, \mathbf{w} \rangle - \sum_{i=1}^{m} \alpha_i (y^{(i)}(\left\langle \mathbf{w}, \mathbf{x}^{(i)} \right\rangle + b) - 1) \tag{52}$$

$$= \frac{1}{2}\left\langle \sum_{i=1}^{m} \alpha_i y^{(i)} \mathbf{x}^{(i)}, \sum_{j=1}^{m} \alpha_j y^{(j)} \mathbf{x}^{(j)} \right\rangle - \sum_{i=1}^{m} \alpha_i \left(y^{(i)} \left(\left\langle \sum_{j=1}^{m} \alpha_j y^{(j)} \mathbf{x}^{(j)}), \mathbf{x}^{(i)} \right\rangle + b \right) - 1 \right) \tag{53}$$

$$= \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} \left\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \right\rangle - \sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} \left\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \right\rangle + \sum_{i=1}^{m} \alpha_i \tag{54}$$

$$= \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} \left\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \right\rangle \tag{55}$$

Therefore, the dual problem is

$$\begin{array}{ll} \max_{\boldsymbol{\alpha}} & L(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j y^{(i)} y^{(j)} (\mathbf{x}^{(i)})^\top \mathbf{x}^{(j)} \\ \text{s.t.} & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0, \\ & \alpha_i \geq 0, i = 1, \ldots, m. \end{array} \tag{56}$$

This is called the dual of the primal problem and the Lagrangian multipliers $\alpha_1, \ldots, \alpha_m$ are called the dual variables. We will see an algorithm (SMO) that solves this quadratic optimization problem later. Once the optimal $\boldsymbol{\alpha}$ is found, a training example $(\mathbf{x}^{(i)}, y^{(i)})$ with $\alpha_i > 0$ is called a support vector. To classify a test example $\mathbf{x}$ using the optimal dual variables $\boldsymbol{\alpha}$,

$$h(\mathbf{x}; \boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i y^{(i)} \left\langle \mathbf{x}^{(i)}, \mathbf{x} \right\rangle + b. \tag{57}$$

We can see that only the support vectors participate in the clasification of the test example. α_i measures the importance of the i-th training example, and $\langle \mathbf{x}^{(i)}, \mathbf{x} \rangle$ "roughly" measures the similarity between $\mathbf{x}^{(i)}$ and $\mathbf{x}$. These two factors are multiplied with the label of the i-th training example to make contribution to the classification output $h(\mathbf{x}; \boldsymbol{\alpha})$.

3.3 Kernel tricks

The dual problem of SVM Eq. (56) requires only the evaluation of the inner product $\langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle$ for any pair of training examples $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$. Similarly, during classification, the SVM model only need to evaluate the inner product $\langle \mathbf{x}^{(i)}, \mathbf{x} \rangle$ for any test example $\mathbf{x}$. If one has not access to the features of $\mathbf{x}$, there is no problem. More generally, we can define the inner product in a even higher-dimensional space (*feature space*) where $\mathbf{x}$ are mapped to, and replace the inner product in the $\mathbb{R}^n$ where $\mathbf{x}$ are sitting in with the new inner product. Optimizing SVM using an inner product without knowing explicitly computing the mapping $\psi(\mathbf{x})$, which can of infinite dimensionality, is called the "kernel trick".

3.3.1 Kernel functions

So why the mapping ψ and the kernel trick? The mapping from $\mathbb{R}^n$ to a higher dimensional feature space enable more expressive decision boundaries to make more accurate classification possible. As the following example shows.

Example 3.7 *Let the data be in the 1-dimensional space $\mathbb{R}$: $x^{(1)} = -1$, $x^{(2)} = 0$, and $x^{(3)} = 1$. Their labels are $y^{(1)} = 1$, $y^{(2)} = -1$, and $y^{(3)} = 1$. There is no linear hyperplane in $\mathbb{R}$ that can separate the data into two classes perfectly. However, if we map the data via the function $\psi(x) = (x, x^2)$ to the 2-dimensional space $\mathbb{R}^2$, the hyperplane in $\mathbb{R}^2$ $\mathbf{w} = [0, 1]^\top$ and $b = -1/2$ can separate the mapped samples.*

A kernel function is $\mathtt{k}(\mathbf{x}, \mathbf{z})$ refers to the inner product in some Euclidean space where $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ are mapped to. Semantically, a kernel function measures the similarity between $\mathbf{x}^{(i)}$ and $\mathbf{x}^{(j)}$ using the inner product in the mapped space:

$$\mathtt{k}(\mathbf{x}, \mathbf{z}) = \langle \psi(\mathbf{x}), \psi(\mathbf{z}) \rangle . \tag{58}$$

If $\mathbf{x}$ and $\mathbf{z}$ are similar, we expect $\mathtt{k}(\mathbf{x}, \mathbf{z})$ to be large (i.e., $\psi(\mathbf{x}$ and $\psi(\mathbf{z})$ are well-aligned in the mapped space).

Example 3.8 *Linear kernel: if $\psi(\mathbf{x}) = \mathbf{x} \in \mathbb{R}^n$, the identity mapping, then the inner product in $\mathbb{R}^n$ is a kernel function associated with the identity mapping.*

Polynomial kernel: if $\mathtt{k}(\mathbf{x}, \mathbf{z}) = (\mathbf{w}^\top \mathbf{z})^2$ and $\mathbf{x}, \mathbf{x} \in \mathbb{R}^2$, then we can recover the mapping ψ as $\psi(\mathbf{x}) = [x_1 x_1, x_1 x_2, x_2 x_1, x_2 x_2]$, since

$$(\mathbf{w}^\top \mathbf{z})^2 = (\sum_{i=1}^{2} x_i z_i)^2 = \sum_{i=1}^{2} \sum_{j=1}^{2} (x_i z_i)(x_j z_j) = \sum_{i=1}^{2} \sum_{j=1}^{2} (x_i x_j)(z_i z_j) = \langle \psi(\mathbf{x}), \psi(\mathbf{z}) \rangle . \tag{59}$$

The function ψ may map $\mathbf{x}$ to an infinite-dimensional Euclidean space.

Example 3.9 *Gaussian kernel is defined as $\mathtt{k}(\mathbf{x}, \mathbf{z}) = \exp\{-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{2\sigma^2}\}$, where $\sigma > 0$ is the so-called "bandwidth" and controls how sensitive the exponential function should be to the distance $\|\mathbf{x} - \mathbf{z}\|^2$ (the larger the σ the less sensitive). The associated mapping ψ maps $\mathbf{x}$ to an infinite dimensional space.*

For a more concrete example, let $\mathbf{x} = [1, 1]^\top$ and $\mathbf{z} = [2, -3]^\top$, with a Gaussian kernel with bandwidth $\sigma = 1$, $\mathtt{k}(\mathbf{x}, \mathbf{z}) = \exp(-(\|[1, 1]^\top - [2, -3]^\top\|^2)/2) = \exp(-((1-2)^2 + (1+3)^2)/2) = \exp(-17/2)$.

Definition 3.10 *A kernel function $\mathtt{k}$ is valid if there is a function ψ so that $\mathtt{k}(\mathbf{x}, \mathbf{z}) = \langle \psi(\mathbf{x}), \psi(\mathbf{z}) \rangle$.*

For the Gaussian kernel, it may be hard to write down the ψ function that explicitly gives the coordinate of the mapped data $\psi(\mathbf{x}) \in \mathbb{R}^\infty$. Luckily, the Mercer's Theorem states that any function $\mathtt{k}(\mathbf{x}, \mathbf{z})$ symmetric in its two arguments, including $\mathtt{k}(\mathbf{x}, \mathbf{z}) = \exp\{-\frac{\|\mathbf{x}-\mathbf{z}\|^2}{\sigma^2}\}$ and $\mathtt{k}(\mathbf{x}, \mathbf{z}) = (\mathbf{w}^\top \mathbf{z})^2$, can be verified to be a valid kernel function if any Gram matrix it defines is positive semi-definite.

First define the Gram matrix for any data $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\} \subset \mathbb{R}^n$ using a symmetric function $\mathtt{k}$ is $K \in \mathbb{R}^{m \times m}$ with $K_{ij} = \mathtt{k}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$.

Theorem 3.11 *$\mathtt{k}$ is a valid kernel function, if and only if, for any data $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\} \subset \mathbb{R}^n$, the Gram matrix is positive semi-definite (PSD).*

For a proof, see Lemma 16.2 of UML. The implication of the Mercer's theorem is that, one does not have to derive the ψ function explicitly for a symmetric function $\mathtt{k}$ and verify that $\mathtt{k}$ does compute an inner product in some Euclidean space where ψ maps the data to. Rather, one can prove that any Gram matrix is PSD to claim that the function $\mathtt{k}$ is indeed a kernel function. (Note: not every symmetric function defines a kernel function).

3.3.2 Kernel trick

The dual problem of SVM with a kernel function is

$$\begin{array}{ll} \max_{\boldsymbol{\alpha}} & L(\boldsymbol{\alpha}) = \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y^{(i)} y^{(j)} \mathtt{k}(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) \\ \text{s.t.} & \sum_{i=1}^m \alpha_i y^{(i)} = 0, \\ & \alpha_i \geq 0, i = 1, \ldots, m. \end{array} \tag{60}$$

and the prediction can be done by

$$h(\mathbf{x}; \boldsymbol{\alpha}) = \sum_{i=1}^m \alpha_i y^{(i)} \mathtt{k}(\mathbf{x}^{(i)}, \mathbf{x}) + b. \tag{61}$$

There is no need to explicitly define the mapping ψ to run SVM in the high-dimensional (or infinite dimensional) feature spaces, so long as one has a kernel function $\mathtt{k}$, validated by the Mercer's theorem.

3.4 Non-separable problems

There are two reasons that one wants to tolerate small portion of mistakes that the optimal hyperplane can make on the training data. First, the training data are usually noisy and a noisy positive example and a noisy negative example can sit closely to each other, leading to a small geometric margin, while the underlying data distribution of the two classes can be separated with a larger margin.

The other reason is more common: most training data are not linearly separable, even with the kernel trick in the high-dimensional space.

We formulate the following soft-SVM (Eq. (5) is called hard-SVM) to tolerate some errors on the training data:

$$\begin{array}{ll} \min_{\mathbf{w}, b, \boldsymbol{\xi}} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^m \xi_i \\ \text{s.t.} & y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1 - \xi_i, i = 1, \ldots, m \\ & \xi_i \geq 0, i = 1, \ldots, m. \end{array} \tag{62}$$

The newly introduced primal variable ξ_i is to allow the functional margin $y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b)$ to be less than 1. In contrast in hard-SVM Eq. (5), $y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) \geq 1$ for all $i = 1, \ldots, m$.

ξ_i is considered as the loss incurred by violating the maximum margin constraint:

$$\xi_i \geq 1 - y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b). \tag{63}$$

The soft-SVM optimizes a loss function called hinge loss:

$$J(\mathbf{x}; \mathbf{w}, b) = \max\{0, 1 - y(\mathbf{w}^\top \mathbf{x} + b)\}, \tag{64}$$

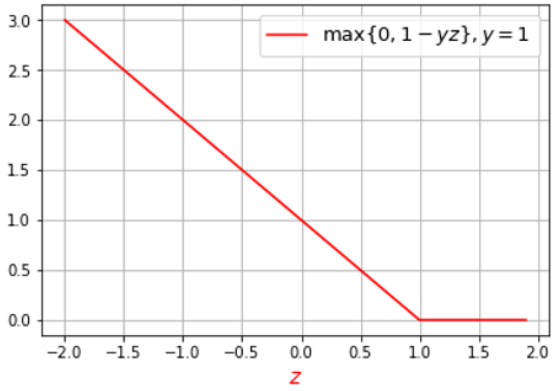

Figure 6: The hinge loss function. $z = \mathbf{w}^\top \mathbf{x}^{(i)} + b$.

3.4.1 Dual problem of soft-SVM

First we find the Lagrangian function

$$L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i(y^{(i)}(\mathbf{w}^\top \mathbf{x}^{(i)} + b) - 1 + \xi_i) - \sum_{i=1}^{m}\mu_i\xi_i. \tag{65}$$

with the $\boldsymbol{\alpha}$ being the same dual variables as in Eq. (56). $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_m]$ are Lagrangian multipliers for the m constraints $\xi_i \geq 0$. $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_m]^\top$ are m additional primal variables.

The Lagrangian is convex for any fixed $\boldsymbol{\alpha}$ and $\boldsymbol{\mu}$, so the dual objective $L(\boldsymbol{\alpha}, \boldsymbol{\mu}) = \min_{\mathbf{w},b,\boldsymbol{\xi}} L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \boldsymbol{\mu})$ can be found by setting the partial of L w.r.t. the primal variables:

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}), \boldsymbol{\mu}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m}\alpha_i y^{(i)}\mathbf{x}^{(i)} = \mathbf{0}, \tag{66}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}), \boldsymbol{\mu}}{\partial b} = \sum_{i=1}^{m}\alpha_i y^{(i)} = 0. \tag{67}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}), \boldsymbol{\mu}}{\partial \xi_i} = C - \alpha_i - \mu_i = 0 \tag{68}$$

Plugging the first two equations in to the Lagrangian function, we find the same dual objective as in hard-SVM:

$$\tilde{L}(\boldsymbol{\alpha}) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y^{(i)}y^{(j)}K_{ij} \tag{69}$$

Let's find out the constraints over the dual variables. Obviously, $\alpha_i \geq 0$ and $\mu_i \geq 0$ due to the KKT conditions. The constraint $C - \alpha_i - \mu_i = 0$ and $\mu_i \geq 0$ leads to $\alpha_i \leq C$ (and the equality holds if $\mu_i = 0$).

The dual problem is then

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & L(\boldsymbol{\alpha}) = \textstyle\sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y^{(i)}y^{(j)}K_{ij} \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{m}\alpha_i y^{(i)} = 0, \\ & C \geq \alpha_i \geq 0, i = 1, \ldots, m. \end{aligned} \tag{70}$$

$K_{ij} = \mathtt{k}(\mathbf{x}^{(i)}), \mathbf{x}^{(j)})$. The remaining KKT conditions (besides $\partial_{\mathbf{w}} L = \mathbf{0}$, $\partial_b L = 0$, and $\partial_{\boldsymbol{\xi}} L = \mathbf{0}$)

are:

$$\begin{aligned}
\alpha_i &\geq 0, i = 1, \ldots, m && (71)\\
y^{(i)}(\mathbf{w}^\top x^{(i)} + b) &\geq 1 - \xi_i, i = 1, \ldots, m && (72)\\
\alpha_i(y^{(i)}(\mathbf{w}^\top x^{(i)} + b) - 1 + \xi_i) &= 0, i = 1, \ldots, m && (73)\\
\mu_i &\geq 0, i = 1, \ldots, m && (74)\\
\xi_i &\geq 0, i = 1, \ldots, m && (75)\\
\mu_i \xi_i &= 0, i = 1, \ldots, m && (76)
\end{aligned}$$

We analyze the following cases:

- $\alpha_i = 0$: $\mu_i = C$ so that $\xi_i = 0$. $(\mathbf{x}^{(i)}, y^{(i)})$ is not a support vector and is classified with a margin at least 1.
- $C > \alpha_i > 0$: $\mu_i > 0$ so that $\xi_i = 0$. $(\mathbf{x}^{(i)}, y^{(i)})$ is a support vector correctly classified on the margin.
- $\alpha_i = C$: $\mu_i = 0$ so that $\xi_i \geq 0$. If $\xi_i = 0$, $(\mathbf{x}^{(i)}, y^{(i)})$ is a support vector on the margin; if $1 \geq \xi_i > 0$, $(\mathbf{x}^{(i)}, y^{(i)})$ is correctly classified between within the margin; if $\xi_i > 1$, $(\mathbf{x}^{(i)}, y^{(i)})$ is incorrectly classified.

The second case helps us find the value of b. For any $(\mathbf{x}^{(i)}, y^{(i)})$ that lies on the margin, $y^{(i)}(\mathbf{w}^\top x^{(i)} + b) = 1$ and thus

$$b = y^{(i)} - \mathbf{w}^\top x^{(i)} = y^{(i)} - \sum_{j=1}^{m} \alpha_i \mathbf{k}(\mathbf{x}^{(j)}, \mathbf{x}^{(i)}). \tag{77}$$

PRML (Eq. (7.37)) uses the average of the above b values of all support vectors with $0 < \alpha_i < C$.

3.5 SMO algorithm for soft-SVM

The dual problem is a quadratic optimization problem and there are off-the-shelf software for such problems. There is a more specialized algorithm called "Sequential Minimal Optimization" (SMO) for SVM designed by John Platt in 1998. The idea is that if the dual variables $\boldsymbol{\alpha}$ satisfy the KKT conditions of SVM, then the $\boldsymbol{\alpha}$ is the optimal solution to the following minimization problem equivalent to the dual problem Eq. (70):

$$\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & -\tilde{L}(\boldsymbol{\alpha}) = \tfrac{1}{2}\textstyle\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y^{(i)} y^{(j)} K_{ij} - \sum_{i=1}^m \alpha_i \\
\text{s.t.} \quad & \textstyle\sum_{i=1}^m \alpha_i y^{(i)} = 0, \\
& 0 \leq \alpha_i \leq C, i = 1, \ldots, m.
\end{aligned} \tag{78}$$

The problem is convex (K is PSD) so a local minimum is a global minimum. If there is any violation of the KKT conditions, then select two dual variables and construct a small quadratic optimization problem involving only the selected variables, with other dual variables held fixed. Minimizing the sub-problem will reduce the objective function $-\tilde{L}$. With only two dual variables, the solution can be found analytically and more efficiently. The algorithm continues with the next two selected dual variables until the KKT conditions are satisfied.

3.5.1 Optimizing the selected variables

Without loss of generality, assume that the selected dual variables are α_1 and α_2. The minimization problem that SMO solves is

$$\begin{aligned}
\min_{\alpha_1, \alpha_2} \quad & W(\alpha_1, \alpha_2) = \tfrac{1}{2} K_{11} \alpha_1^2 + \tfrac{1}{2} K_{22} \alpha_2^2 + y^{(1)} y^{(2)} K_{12} \alpha_1 \alpha_2 \\
& -(\alpha_1 + \alpha_2) + y^{(1)} \alpha_1 \textstyle\sum_{j=3}^m y^{(j)} \alpha_j K_{j1} + y^{(2)} \alpha_2 \sum_{j=3}^m y^{(j)} \alpha_j K_{j2} \\
\text{s.t.} \quad & \alpha_1 y^{(1)} + \alpha_2 y^{(2)} = -\textstyle\sum_{j=3}^m y^{(j)} \alpha_j = \rho, \\
& 0 \leq \alpha_i \leq C, i = 1, 2.
\end{aligned} \tag{79}$$

ρ is a constant due to the balancing equation $\sum_{j=1}^{m} y^{(j)}\alpha_j = 0$ and α_j, $j \geq 3$ are held fixed. As a result, α_1 and α_2 must be changed at the same time to satisfy the balancing equation. The Hessian of W with respect to α_1 and α_2 is the PSD matrix

$$\begin{bmatrix} y^{(1)}y^{(1)}K_{11} & y^{(1)}y^{(2)}K_{12} \\ y^{(2)}y^{(1)}K_{21} & y^{(2)}y^{(2)}K_{22} \end{bmatrix} = \begin{bmatrix} \langle y^{(1)}\psi(\mathbf{x}^{(1)}), y^{(1)}\psi(\mathbf{x}^{(1)})\rangle & \langle y^{(1)}\psi(\mathbf{x}^{(1)}), y^{(2)}\psi(\mathbf{x}^{(2)})\rangle \\ \langle y^{(1)}\psi(\mathbf{x}^{(2)}), y^{(1)}\psi(\mathbf{x}^{(1)})\rangle & \langle y^{(1)}\psi(\mathbf{x}^{(2)}), y^{(2)}\psi(\mathbf{x}^{(2)})\rangle \end{bmatrix} \quad (80)$$

due to the kernel function property. Therefore, the subproblem is convex in α_1 and α_2.

To further simplify the above problem, let

$$g_i = h(\mathbf{x}^{(i)}, \boldsymbol{\alpha}) = \sum_{j=1}^{m} \alpha_j y^{(j)} K_{ij} + b \quad (81)$$

$$v_i = \sum_{j=3}^{m} \alpha_j y^{(j)} K_{ij} = g_i - \sum_{j=1}^{2} \alpha_j y^{(j)} K_{ij} - b, \qquad i = 1, 2. \quad (82)$$

$W(\alpha_1, \alpha_2) = \frac{1}{2}K_{11}\alpha_1^2 + \frac{1}{2}K_{22}\alpha_2^2 + y^{(1)}y^{(2)}K_{12}\alpha_1\alpha_2 - (\alpha_1 + \alpha_2) + y^{(1)}\alpha_1 v_1 + y^{(2)}\alpha_2 v_2$.

Due to the equality constraint, picking a value for α_2 automatically determines the value for α_1, so we consider optimizing α_2 to minimizing the $W(\alpha_1, \alpha_2)$. First find $\alpha_1 = y^{(1)}(\rho - y^{(2)}\alpha_2)$ since $y^{(1)}y^{(1)} = y^{(2)}y^{(2)} = 1$. Substitute this into $W(\alpha_1, \alpha_2)$, we have

$$W(\alpha_2) = \frac{1}{2}K_{11}(\rho - y^{(2)}\alpha_2)^2 + \frac{1}{2}K_{22}\alpha_2^2 + y^{(2)}K_{12}(\rho - y^{(2)}\alpha_2)\alpha_2 \quad (83)$$

$$-(\rho - y^{(2)}\alpha_2)y^{(1)} - \alpha_2 + v_1(\rho - y^{(2)}\alpha_2) + v_2 y^{(2)}\alpha_2. \quad (84)$$

In this equation, α_2 is regarded as a variable to be optimized, and the quantities ρ and v_i are considered constants computed using the $\boldsymbol{\alpha}$ values before the update.

The SMO problem becomes the following single-variable optimization problem:

$$\begin{aligned} \min_{\alpha_2} \quad & W(\alpha_2) \\ \text{s.t.} \quad & \alpha_1 y^{(1)} + \alpha_2 y^{(2)} = \rho, \\ & 0 \leq \alpha_2 \leq C, i = 1, 2. \end{aligned} \quad (85)$$

The constraints $0 \leq \alpha_i \leq C, i = 1, 2$ are called box constraints, and the equality constraint $\alpha_1 y^{(1)} + \alpha_2 y^{(2)} = \rho$ has α_1 and α_2 on a straight line. The feasible regions for α_1 and α_2 are shown in Figure 7.

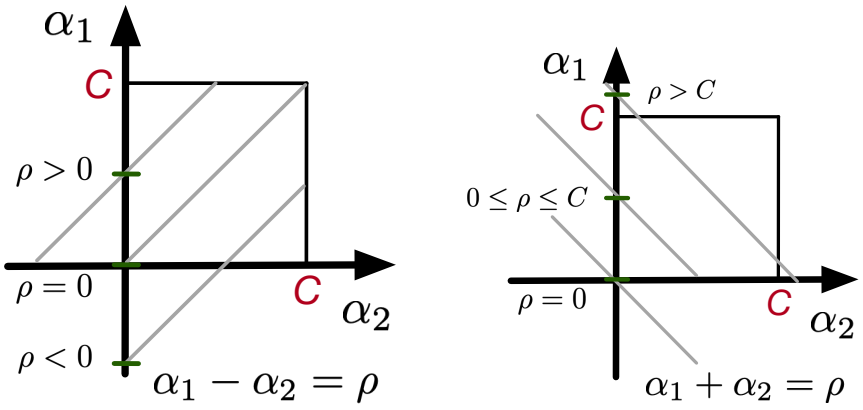

Figure 7:

- If $y^{(1)}y^{(2)} = -1$, then the linear constraint is represented as the straight lines in the left panel of Figure 7. If $\rho < 0$, then the line intersects the α_2-axis at some positive number $-\rho > 0$. Due to the box constraint $0 \leq \alpha_2 \leq C$ so that the updated α_2^{new} must $\leq H = \min\{C, C - \rho\} = \min\{C, C - (\alpha_1^{\text{old}} - \alpha_2^{\text{old}})\}$, where α_i^{old} are the values before the update. For similar reason,, the lower bound of α_2^{new} will be $L = \max\{0, -\rho\} = \max\{0, -(\alpha_1^{\text{old}} - \alpha_2^{\text{old}})\}$.

- If $y^{(1)}y^{(2)} = 1$, then the linear constraint is represented as the straight lines in the right panel of Figure 7. Due to the box constraint $0 \leq \alpha_2 \leq C$, the updated α_2^{new} will be $\leq H = \min\{C, \rho\} = \min\{C, \alpha_1^{\text{old}} + \alpha_2^{\text{old}}\}$ and $\geq L = \max\{0, \rho - C\} = \max\{0, \alpha_1^{\text{old}} + \alpha_2^{\text{old}} - C\}$.

Without considering the box and linear constraints, minimizing the above convex quadratic function can be done by setting 0 to the partial gradient of $W(\alpha_2)$ with respect to α_2.

$$\frac{\partial W(\alpha_2)}{\partial \alpha_2} \tag{86}$$

$$= K_{11}\alpha_2 + K_{22}\alpha_2 - 2K_{12}\alpha_2 - y^{(2)}K_{11}\rho + y^{(2)}K_{12}\rho + y^{(2)}(y^{(1)} - y^{(2)} + v_2 - v_1) \tag{87}$$

$$= \alpha_2(K_{11} + K_{22} - 2K_{12}) \tag{88}$$

$$+y^{(2)}(-K_{11}\rho + K_{12}\rho - y^{(2)} + y^{(1)} + v_2 - v_1). \tag{89}$$

$$(K_{11} + K_{22} - 2K_{12})\alpha_2^{\text{new}} = y^{(2)}(K_{11}\rho - K_{12}\rho + y^{(2)} - y^{(1)} - v_2 + v_1) \tag{90}$$

Let's simplify the right hand side,

$$y^{(2)}(K_{11}\rho - K_{12}\rho + y^{(2)} - y^{(1)} - v_2 + v_1) \tag{91}$$

$$= y^{(2)}(K_{11}\rho - K_{12}\rho + y^{(2)} - y^{(1)} \tag{92}$$

$$- (g_2 - \sum_{j=1}^{2} \alpha_j^{\text{old}} y^{(j)} K_{2j} - b) + (g_1 - \sum_{j=1}^{2} \alpha_j^{\text{old}} y^{(j)} K_{1j} - b)) \tag{93}$$

$$\tag{94}$$

Since ρ is a constant and can be computed as $\rho = y^{(1)}\alpha_1^{\text{old}} + y^{(1)}\alpha_2^{\text{old}}$. Plugging this in to the above equation, we have

$$(K_{11} + K_{22} - 2K_{12})\alpha_2^{\text{new}} = (K_{11} + K_{22} - 2K_{12})\alpha_2^{\text{old}} + y^{(2)}(g_1 - y^{(1)} - (g_2 - y^{(2)})) \tag{95}$$

We obtain the update rule for α_2 without constraint:

$$\alpha_2^{\text{new}} = \alpha_2^{\text{old}} + \frac{y^{(2)}(g_1 - y^{(1)} - (g_2 - y^{(2)}))}{K_{11} + K_{22} - 2K_{12}}. \tag{96}$$

Now clip α_2^{new} to the lower and upper bounds derived from the box and linear constraints over α_2, we have

$$\alpha_2^{\text{new,clipped}} = \begin{cases} H & \text{if } \alpha_2^{\text{new}} > H, \\ L & \text{if } \alpha_2^{\text{new}} < L, \\ \alpha_2^{\text{new}}, & \text{otherwise.} \end{cases} \tag{97}$$

With the updated α_2 value, you can find the new value for α_1 by

$$\alpha_1^{\text{new}} = y^{(1)}(\rho - y^{(2)}\alpha_2^{\text{new,clipped}}) = \alpha_1^{\text{old}} + y^{(1)}y^{(2)}(\alpha_2^{\text{old}} - \alpha_2^{\text{new,clipped}}). \tag{98}$$

3.5.2 Selecting dual variables

The SMO paper [**?**] devised a smart strategy to select the two dual variables α_1 and α_2 for optimization. The heuristic finds the variables that violate the KKT conditions and likely lead to the most reduction in the objective in Eq. (79). We don't use these sophisticated heuristics in our project but use a simpler one that works on simpler classification problems.

3.5.3 Pseudo codes of SMO

Algorithm 1: SMO

Given: training data $\{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{m}$, kernel function `k`, T (max iterations), max_passes (allows multiple passes without parameter change), τ (record_every), tol (a small positive tolerance).;
Initialization: set all dual variables α_i to zero; set bias $b = 0$.;
for $t = 1$ *to* T **do**
    num_passes = 0;
    while num_passes < max_passes **do**
        num_changes = 0;
        for $i = 1$ *to* m **do**
            if α_i violates any KKT conditions **then**
                Randomly sample $j \in \{1, \ldots, m\}$ so that $i \neq j$;
                Compute the lower bound L and upper bound H for the new α_j value;
                Compute the new α_j value;
                if the new α_j is not different from the old α_j by more than *tol* **then**
                    Go to the next i.
                end
                Compute the new value of α_i given the new α_j;
                Commit the two new values to the actual dual variables;
                Update the bias b;
                Increase the num_changes by 1;
            end
        end
        if num_changes == 0 **then**
            num_passes += 1 ; `// One pass without changing any parameters.`
        else
            num_passes = 0 ; `// At least one pair of α's are changed.`
        end
    end
    Record dual and primal objective function values and the model parameters (all dual variables and b) every τ iterations.
end
return The recorded objective function values and model parameters over the course of training.