Chapter 1 Support Vector Machine

Support Vector Machine (SVM) is also a binary classification model, often introduced as an alternative to logistic regression, since—as we have discussed in the previous chapter—logistic regression has several notable limitations. SVM provides an elegant framework that allows us to transfer powerful results and intuition from optimization theory into machine learning.

1.1 Conditioned Optimization

Unlike the unconstrained optimization problems we studied earlier, the support vector machine is originally formulated as a **constrained optimization problem**. This distinction is fundamental. In mathematics, unconstrained optimization is defined as:

$$\operatorname*{argmin}_{x} f(x) \tag{1.1}$$

When we deal with constrained optimization, we add an auxiliary constraint $h(x) = 0$ (where both $f(x)$ and $g(x)$ are differentiable). Considering the geometric meaning of the gradient, we observe:

1. For any point on the surface defined by $g(x)$, the gradient $\nabla g(x)$ must be orthogonal to the surface.
2. For any critical point x^* that achieves a local minimum, the gradient $\nabla f(x^*)$ must also be orthogonal to the surface.

Note *The second property can be understood by analogy with electrostatics: near a conductor, the electric field must be orthogonal to the surface. Otherwise, there would be a component of the field (gradient) along the constrained sub-hypersurface, violating equilibrium.*

Extending this analogy: in the electrostatic case, such a tangential component would induce currents along the conductor's surface. In the optimization setting, a tangential component corresponds to a gradient lying along the constraint manifold, implying that the objective function can still be further optimized without violating the constraint.

Moreover, the charge distribution problem in electrostatics can itself be interpreted as an energy minimization process: the charges rearrange to minimize the system's electrostatic energy. In this sense, the equilibrium state of charges on a conductor can be seen as an optimization problem, which strengthens the analogy with constrained optimization in mathematics.

Combining the two conditions above, we obtain the **necessary** condition for a local minimum point, as shown in Figure 1.1:

$$\exists \lambda \quad \text{s.t.} \quad \nabla f(x^*) + \lambda \nabla h(x^*) = 0 \tag{1.2}$$

Remark In some contour examples, the constraint condition does not cover the entire space. In such cases, the above equation may *not* always be satisfied.

Now we introduce the *Lagrangian function*, which plays a central role in optimization theory:

$$L(x, \lambda) := f(x) + \underbrace{\lambda}_{\text{Lagrange Multiplier}} h(x) \tag{1.3}$$

Extending to the case of multiple constraints, the Lagrangian is defined as:

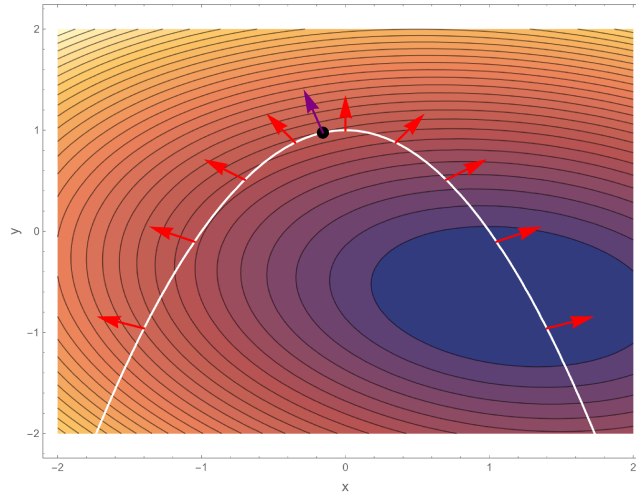$$L(x, \lambda) := f(x) + \sum_{i=1}^{k} \lambda_i h_i(x) \tag{1.4}$$

Figure 1.1: Illustration of the necessary condition for constrained local minima.

Theorem 1.1

If x^* is a local minimum point, then there exists a set of multipliers $\lambda = (\lambda_1, \ldots, \lambda_k)$ such that

$$\begin{cases} \nabla_x L(x^*, \lambda) = 0, \\ \nabla_\lambda L(x^*, \lambda) = 0. \end{cases} \tag{1.5}$$

The first condition corresponds to the stationarity (minimum) condition, while the second condition enforces the constraints.

Corollary 1.1

In multi-constrained problems, the minimum condition becomes

$$\nabla f(x^*) + \sum_{i=1}^{k} \lambda_i \nabla h_i(x^*) = 0. \tag{1.6}$$

Remark In higher dimensions, the constrained submanifold may possess certain free variables that can be chosen arbitrarily while still satisfying the minimum condition. Specifically, if $\nabla f(x^*)$ cannot be expressed as a linear combination of $\nabla h_1(x^*)$ and $\nabla h_2(x^*)$, it is possible to continue optimizing along the intersection submanifold defined by $h_1 = 0$ and $h_2 = 0$, as shown in Figure 1.2. Therefore, it must be emphasized that the above equation provides only a **necessary condition**, not a sufficient one.

Example 1.1 Consider the following constrained optimization problem:

$$\underset{x_1, x_2}{\operatorname{argmin}} \ x_1 + x_2 \tag{1.7}$$

subject to the constraint

$$(x_1 - 1)^2 + x_2^2 - 1 = 0 \tag{1.8}$$

Solution $1 - \sqrt{2}$.

Note *The Lagrange multiplier method can provide critical points, but these do not always correspond to the minimum solution.*

Note *In convex settings, any local minimum is also a global minimum.*

A new problem arises: what if we have an inequality constraint instead of a strict equality, i.e., $g(x) \leq 0$?

1.2 Inequality optimization

We assume the feasible set of x is compact; otherwise, the existence of a minimum may be problematic.
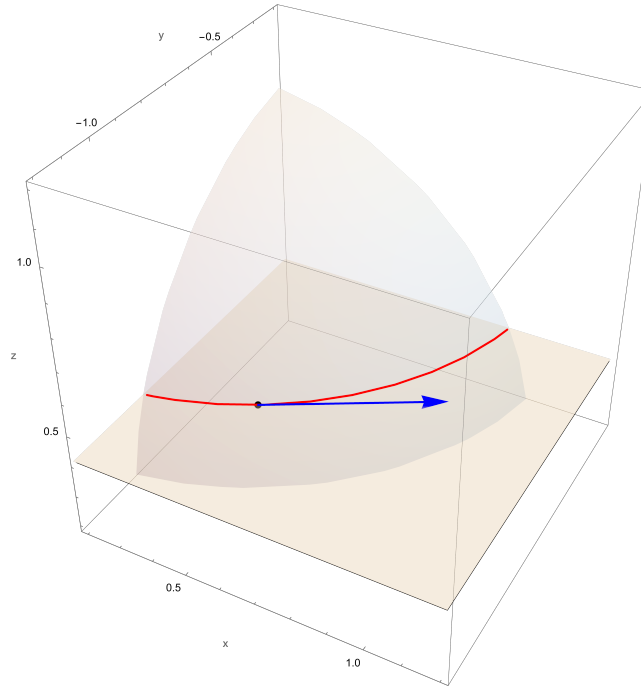
Figure 1.2: Illustration of optimization along the intersection of two constraint manifolds.

By gradient analysis, we obtain:

1. For any point on the surface $g(x) = 0$, $\nabla g(x)$ is orthogonal to the surface and points outward from the feasible region $g(x) \leq 0$. This is straightforward, since $\nabla g(x)$ indicates the steepest direction of increase.

2. For a local minimum point x^*:

   (a). If x^* lies on the surface $g(x) = 0$ (in other words, the constraint is *active*), then $-\nabla f(x^*)$ must be aligned with $\nabla g(x^*)$. Equivalently, $\exists \mu > 0$ such that

   $$\nabla f(x^*) + \mu \nabla g(x^*) = 0$$

   (b). If x^* does not lie on the surface, i.e., $g(x) < 0$ (the constraint is *inactive*), we simply require $\nabla f(x^*) = 0$. In the unified form above, this corresponds to $\exists \mu = 0$ such that

   $$\nabla f(x^*) + \mu \nabla g(x^*) = 0$$

Note *How should we understand the meaning of an "active" constraint? When the inequality constraint is not tight (i.e., $g(x) < 0$), it does not affect gradient descent optimization. In this case, the constraint is effectively absent, which is why we call it* inactive.

Again, we define the Lagrangian as

$$L(x, \mu) = f(x) + \mu g(x) \tag{1.9}$$

where $g(x)$ denotes the inequality constraint function. Following the same steps as in the equality-constrained case, we obtain:

Theorem 1.2

If x^* is a local minimum point, then

$$\begin{cases} \nabla L(x^*, \mu) = 0, \\ g(x^*) \leq 0, \\ \mu \geq 0, \\ \mu \cdot g(x^*) = 0. \end{cases} \tag{1.10}$$

The last condition (the *complementary slackness*) unifies the boundary and interior cases, since either $g(x^*) = 0$

(active constraint) or $\mu = 0$ (inactive constraint) must hold.

1.3 General Case

In the most general case, we consider the optimization problem

$$\operatorname*{argmin}_{x} f(x) \tag{1.11}$$

subject to equality constraints $h_i(x) = 0$, $i \in [k]$, and inequality constraints $g_j(x) \leq 0$, $j \in [l]$.

The corresponding Lagrangian is defined as

$$L(x, \lambda, \mu) := f(x) + \sum_{i \in [k]} \lambda_i h_i(x) + \sum_{j \in [l]} \mu_j g_j(x) \tag{1.12}$$

In this setting, a local minimum x^* must satisfy a series of conditions known as the **Karush–Kuhn–Tucker (KKT) conditions**:

> **Theorem 1.3 (KKT Conditions)**
>
> *If x^* is a local minimum point, then there exist multipliers (λ, μ) such that*
>
> $$\begin{cases} \nabla L(x^*, \lambda, \mu) = 0, \\ h_i(x^*) = 0, & \forall i \in [k], \\ g_j(x^*) \leq 0, & \forall j \in [l], \\ \mu_j \geq 0, & \forall j \in [l], \\ \mu_j \cdot g_j(x^*) = 0, & \forall j \in [l]. \end{cases} \tag{1.13}$$
>
> ♡

In most cases, the above conditions cannot be solved explicitly in closed form.

1.4 Support Vector Machine

1.4.1 Hard Margin

Consider a linearly separable dataset $\{(x_i, y_i)\}$ with $y_i \in \{0, 1\}$, and a linear model

$$f(x) = w^\top x + b \tag{1.14}$$

Although introducing a regularization term can make the solution unique by adjusting the hyperparameter λ, this does not resolve the issue of determining which solution $f(x; \lambda)$ is preferable among all possible choices, since λ is itself arbitrary.

Our goal is for the model to generalize well to test data drawn from the same distribution as the training set. A direct and intuitive way to achieve this is to choose the separating hyperplane that maximizes the distance between the plane and the nearest datapoints.

> **Definition 1.1 (Support Vectors)**
>
> *The datapoints that lie closest to the separating hyperplane, such that the vectors orthogonal to the hyperplane connect the hyperplane to these datapoints, are called* support vectors.
>
> ♣

In the case of SVM, we aim to find the **Max-Margin Classifier**, i.e., the hyperplane that maximizes the minimum margin across all training points. Here, the *margin* refers to the distance from a point to the hyperplane, which is equivalently the norm of the vector associated with its support vector.

Remark Why does $w^\top x + b = 0$ represent a hyperplane? Consider the hyperplane through the origin, given by $w^\top x = 0$. Translating this plane until it passes through some point $x_0 \notin \{x \mid w^\top x = 0\}$, we obtain

$$w^\top (x - x_0) = 0$$

Setting $-w^\top x_0 = b$, we recover the general hyperplane equation
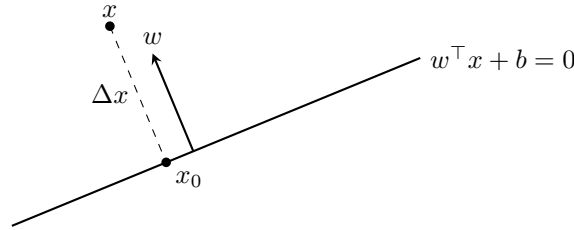
$$w^\top x + b = 0$$

Remark Note that the interpretation of b differs between the two cases $y = w^\top x + b$ and $0 = w^\top x + b$. In the latter, we can observe that $|b| = \|w\|\|x_0\|$, which corresponds to the offset determined by x_0.

The distance between a datapoint and the hyperplane is given by:

Theorem 1.4

$$d = \frac{|w^\top x + b|}{\|w\|}. \tag{1.15}$$

♡

Geometrically, we obtain:

1. Consider

$$\begin{cases} x_0 + \Delta x \dfrac{w}{\|w\|} = x, \\ w^\top x_0 + b = 0, \end{cases}$$

so that

$$w^\top x_0 + w^\top \Delta x \frac{w}{\|w\|} = w^\top x$$

$$-b + w^\top \Delta x \frac{w}{\|w\|} = w^\top x$$

$$\Delta x \|w\| = w^\top x + b$$

This matches exactly the expression we obtained in (1.15).

2. From the Lagrangian perspective:

$$\underset{x_0}{\mathrm{argmin}}\, \frac{1}{2}\|x - x_0\|^2 \quad \Rightarrow \quad L(x_0, \lambda) = \frac{1}{2}\|x - x_0\|^2 + \lambda(w^\top x_0 + b)$$

$$s.t.\ w^\top x_0 + b \qquad \nabla_{x_0} L(x_0, \lambda) = 0 \quad \Rightarrow \quad x_0 - x + \lambda w = 0$$

$$\nabla_x L(x_0, \lambda) = 0 \quad \Rightarrow \quad w^\top x_0 + b = 0$$

This leads to the same solution as above.

When $\Delta > 0$, the point x lies on the positive side of the hyperplane; when $\Delta = 0$, x lies exactly on the hyperplane; and when $\Delta < 0$, x lies on the negative side.

We define

$$\gamma_i = \frac{y_i(w^\top x_i + b)}{\|w\|} \tag{1.16}$$

to incorporate both the label information and the margin (distance to the hyperplane). If $\gamma_i > 0$, the point x_i is correctly classified.

Define the margin

$$\gamma = \min_{i \in [n]} \gamma_i \tag{1.17}$$

and the SVM optimization problem can be formalized as

$$\underset{w,b}{\mathrm{argmax}}\, \gamma, \quad s.t.\ \forall i,\ \frac{y_i(w^\top x_i + b)}{\|w\|} \geq \gamma \tag{1.18}$$

This formulation seems problematic, since γ is not independent. To resolve this, suppose (x_0, y_0) is a point achieving the minimum margin γ_i. Then

$$\gamma = \frac{y_0(w^\top x_0 + b)}{\|w\|} \tag{1.19}$$

Thus, the SVM objective can be reformulated as

$$\underset{w,b}{\text{argmax}} \ \frac{y_0(w^\top x_0 + b)}{\|w\|}, \quad \text{s.t.} \ \forall i, \ y_i(w^\top x_i + b) \geq y_0(w^\top x_0 + b). \tag{1.20}$$

We claim that we can make $y_0(w^\top x_0 + b)$ arbitrarily large without affecting the result.

Remark This is immediate: scaling w and b by the same factor λ does not change the optimization, since the numerator $y_i(w^\top x_i + b)$ and the denominator $\|w\|$ both scale by λ, and the factor cancels during simplification.

Thus, we may set $y_0(w^\top x_0 + b) = 1$. Define the *functional margin* as $y_i(w^\top x_i + b)$ and the *geometric margin* as $y_i(w^\top x_i + b)/\|w\|$. Since the functional margin alone has no intrinsic meaning, it is natural to rescale it by setting the norm of the support vector to unity.
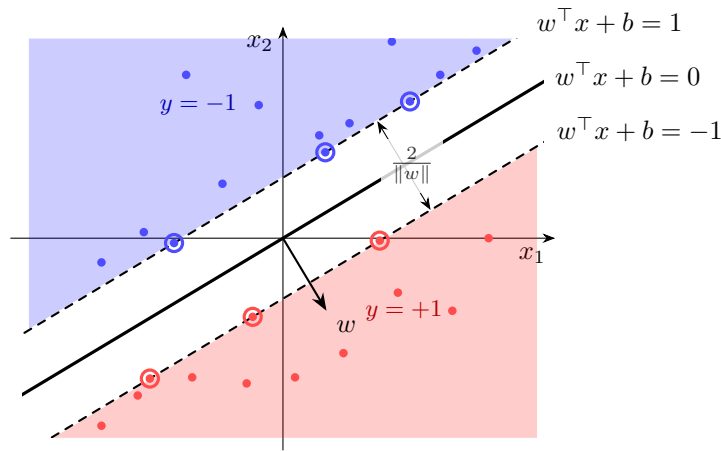

Figure 1.3: Support Vector Machine.

This yields the primal form of the SVM optimization problem:

$$\underset{w,b}{\text{argmax}} \ \frac{1}{\|w\|}, \quad \text{s.t.} \ y_i(w^\top x_i + b) \geq 1 \quad \forall i \tag{1.21}$$

which is equivalent to

$$\underset{w,b}{\text{argmin}} \ \frac{1}{2}\|w\|^2, \quad \text{s.t.} \ y_i(w^\top x_i + b) \geq 1 \quad \forall i \tag{1.22}$$

In other words, the goal is to separate the dataset correctly while ensuring a sufficient margin, and simultaneously minimizing the norm of w. This principle is known as **Structural Risk Minimization** (SRM).

Remark Unlike logistic regression, datapoint far from separate hyperplane won't contribute loss. That is, only small subset of dataset is active as a constraint.

> **Proposition 1.1**
>
> *SVM is a convex quadratic programming, and we can solve it in poltnomial time with standard package.* ♡

Problem 1.1

1. What if the dataset is not linearly separable?
2. Wht if γ too small because of outliers?