Linear Support Vector Machine (LSVM)

Introduction

Support vector machine (SVM) is designed for binary classification. It finds a decision boundary that maximally separates data of the two classes. Let $Y = \{-1, +1\}$ be the label set.

A linear decision boundary is a linear hyper-plane $G$, defined as

$$G := \{x; \ x^T \beta + \beta_0 = 0\}, \tag{1}$$

where $\beta$ is a normal of $G$ (i.e., perpendicular to $G$), and $\beta_0$ is a bias term.

[Discussion] Why does a linear hyper-plane have the form (1)?

[Exercise] Prove $\beta$ is perpendicular to $G$ (i.e. perpendicular to every vector in $G$).

The signed distance from any point $x$ and $G$ is

$$d(x, G) = \frac{x^T \beta + \beta_0}{||\beta||}. \tag{2}$$

By ‘signed’, we mean $d(x, G) > 0$ if $x$ is on the side of $G$ where $\beta$ is pointing at, and $d(x, G) < 0$ if $x$ is on the other side of $G$.

If $||\beta|| = 1$, the signed distance is $d(x, G) = x^T \beta + \beta_0$.

[Exercise] Derive (2).

We can use $G$ as a linear decision boundary, e.g., let it classify every $x$ satisfying $d(x, G) > 0$ to class $y = +1$ and to every $x$ satisfying $d(x, G) < 0$ to class $y = -1$.

The two classes are linearly separable if they can be separated by some linear decision boundary.

Optimal Separating Hyperplane

If two classes are linearly separable, there often exists multiple $G$’s that can perfectly separate the two classes in a training sample. Which $G$ should we select?

The optimal separating hyperplane selects the $G$ with the maximum distances to training data. It finds such $G$ by solving the following optimization problem:

$$\max_{\beta, \beta_0, ||\beta||=1} M \quad \text{s.t.} \quad y_i (x_i^T \beta + \beta_0) \geq M, \ i = 1, \ldots, n. \tag{3}$$
Problem (3) can be interpreted as follows.

First, note $x_i^T \beta + \beta_0$ is the signed distance from $x_i$ to $G$ (under the constraint $||\beta|| = 1$).

Suppose we make the decision rule that any $x_i$ satisfying $x_i^T \beta + \beta_0 > 0$ is assigned to class $+1$ and any $x_i$ satisfying $x_i^T \beta + \beta_0 < 0$ is assigned to class $-1$. Then the true label $y_i$ in (3) will turn signed distance of correctly classified $x_i$ into an unsigned distance because

- [correctly classified] If $x_i^T \beta + \beta_0 > 0$ and $y_i = +1$, then $y_i(x_i^T \beta + \beta_0) = |x_i^T \beta + \beta_0| > 0$.
- [correctly classified] If $x_i^T \beta + \beta_0 < 0$ and $y_i = -1$, then $y_i(x_i^T \beta + \beta_0) = |x_i^T \beta + \beta_0| > 0$.
- [falsely classified] If $x_i^T \beta + \beta_0 > 0$ but $y_i = -1$, then $y_i(x_i^T \beta + \beta_0) < 0$. To satisfy constraint $y_i(x_i^T \beta + \beta_0) \geq M$, the learner will prioritize on finding a $G$ that can correctly classify $x_i$.

Finally, by maximizing $M$, the learner maximizes the smallest distance from any instance to $G$.

**Duality and Support Vector Classifier**

We can simplify (3) in three steps:

(1) remove $||\beta|| = 1$ by replacing $y_i(x_i^T \beta + \beta_0) \geq M$ with $y_i(x_i^T \beta + \beta_0) \geq M||\beta||$.

(2) replace $y_i(x_i^T \beta + \beta_0) \geq M||\beta||$ with $y_i(x_i^T \beta + \beta_0) \geq C$ with fixed $C = M||\beta||$.

(3) fix $C = 1$.

Then we have a new optimization problem

$$
\min_{\beta, \beta_0} \frac{1}{2}||\beta||^2 \\
\text{s.t. } y_i(x_i^T \beta + \beta_0) \geq 1, \ i = 1, \ldots, n.
$$

The optimal $\beta$’s in (3) and (4) will have the same direction but different norms, because (4) is still maximizing $M$ but at an additional cost of minimizing $||\beta||$. This is good enough because if two normals have the same direction, they are supporting the same separating hyperplane $G$.

Further, we can interpret $1/||\beta||$ as the thickness of a slab centered at $G$ and excluding all $x$. It is also known as the margin of $G$, and (4) is thus called max-margin classifier.

To solve (4), we can first apply the Lagrange Multiplier and convert it into an unconstrained optimization problem. The new objective function is

$$
J(\beta, \beta_0) = \frac{1}{2}||\beta||^2 - \sum_{i=1}^{n} \alpha_i[y_i(x_i^T \beta + \beta_0) - 1],
$$

where $\alpha_i \geq 0$ for $i = 1, \ldots, n$.

Since $J$ is a quadratic function. We can apply the critical point method and have

$$
\frac{\partial}{\partial \beta} J(\beta, \beta_0) = 0 \quad \Rightarrow \quad \beta = \sum_{i=1}^{n} \alpha_i y_i x_i.
$$
\[ \frac{\partial}{\partial \beta_0} J(\beta, \beta_0) = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0. \quad (7) \]

**Exercise** Derive (6) and (7).

Plugging (6, 7) back to (5), we eliminate \( \beta, \beta_0 \) and have a new objective function of \( \alpha \)'s

\[ J_D(\alpha_i) = \sum_{i=1}^{n} \alpha_i - \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{1}{2} \alpha_i \alpha_j y_i y_j x_i^T x_j, \quad (8) \]

which will be minimized with the constraints

\[ \alpha_i \geq 0 \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0. \quad (9) \]

Problem (8, 9) is the **Wolfe dual** of problem (5). It is a simpler convex optimization task and suggests an insight that the optimal \( \beta \) is a linear combination of training instances.

The Lagrange Multiplier does not guarantee the optimal \( \alpha \) lead to the optimal \( \beta \). For the latter to also be optimal, the solution must satisfy the Karush-Kuhn-Tucker (KKT) conditions, which include (6), (7), (9) and the following condition

\[ \alpha_i [y_i (x_i^T \beta + \beta_0) - 1] = 0, \quad i = 1, \ldots, n. \quad (10) \]

Condition (10) gives an important insight. Consider two cases:

1. if \( \alpha_i > 0 \), then \( y_i (x_i^T \beta + \beta_0) - 1 = 0 \). So \( x_i \) is correctly classified and lies on the margin.
2. if \( y_i (x_i^T \beta + \beta_0) - 1 \neq 0 \), which means \( x_i \) is not on the margin, then \( \alpha_i = 0 \).

Recall the optimal model satisfies \( \beta_* = \sum_i \alpha_i x_i \). The above analysis suggests that only instances on the margin are used to express \( \beta_* \). These instances are called the support vectors, and the classifier based on (8, 9) is called the support vector classifier (SVC).

SVC is also known as hard-margin (linear) SVM. It is ‘hard’ because it assumes data are linearly separable and does not tolerate misclassified instances. It is ‘linear’ because its assumes a linear separating hyperplane. In the following, we will see the so-called ‘soft-margin’ linear SVM that relaxes the first assumption. In later lectures, we will see the commonly called ‘SVM’ that relaxes the second assumption using kernel methods.

**Soft-Margin Linear SVM**

Soft-margin linear SVM does not assume data are linearly separable. It learns a linear decision boundary that tolerates ‘misbehaved’ instances, i.e., instances that are misclassified or lies inside the margin (Fig 1). It implements such tolerance by introducing a slack variable \( \epsilon_i \) for each instance \((x_i, y_i)\), such that the original hard-margin constraint \( y_i (x_i^T \beta + \beta_0) \geq M \) becomes soft-margin constraint

\[ y_i (x_i^T \beta + \beta_0) \geq M (1 - \epsilon_i). \quad (11) \]

In (11), if \( \epsilon > 1 \) then we can have \( y_i (x_i^T \beta + \beta_0) \) can be negative which means \( x_i \) is misclassified; if \( 1 > \epsilon > 0 \) then \( y_i (x_i^T \beta + \beta_0) \) can be smaller than \( M \) which means \( x_i \) lies in the slab; if \( \epsilon = 0 \), then soft-margin is equivalent to hard-margin.

\[ ^1 \text{We will derive this KKT condition later.} \]
Although soft-margin tolerates misbehaviors, it minimizes the misbehaviors by minimizing $\sum_i \epsilon_i$. Adding this and (11) back to (4), we obtain the optimization problem of soft-margin linear SVM:

$$\min_{\beta, \beta_0, \epsilon} \frac{1}{2} \| \beta \|^2 + C \sum_{i=1}^n \epsilon_i$$

s.t. $y_i(\mathbf{x}_i^T \beta + \beta_0) \geq 1 - \epsilon_i, \quad \epsilon_i \geq 0, \quad i = 1, 2, \ldots, n$ \hspace{1cm} (12)

where $C$ is a hyper-parameter controlling the degree of tolerance.

[Discussion] What happens if $C = \infty$?

Problem (12) can be solved in a similar way as solving hard-margin linear SVM. First, we derive a dual problem using the Lagrange multiplier; the Lagrange function is

$$L(\beta, \beta_0, \epsilon, \mu) = \frac{1}{2} \| \beta \|^2 + C \sum_{i=1}^n \epsilon_i - \sum_{i=1}^n \alpha_i [y_i(\mathbf{x}_i^T \beta + \beta_0) - (1 - \epsilon_i)] - \sum_{i=1}^n \mu_i \epsilon_i.$$ \hspace{1cm} (13)

Setting the derivatives w.r.t. $\beta, \beta_0$ and $\epsilon$ to zero, respectively, we have

$$\beta = \sum_{i=1}^n \alpha_i y_i x_i$$ \hspace{1cm} (14)

$$0 = \sum_{i=1}^n \alpha_i y_i$$ \hspace{1cm} (15)

$$\alpha_i = C - \mu_i,$$ \hspace{1cm} (16)

and $\alpha_i, \mu_i, \epsilon_i \geq 0$ for all $i$. Plugging these back to (13) gives the Wolfe dual objective function

$$L_D(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j.$$ \hspace{1cm} (17)

[Exercise] Derive (17).

The Wolfe dual sets a lower bound of the original problem, i.e.,

$$L_D(\alpha) = \min_{\beta, \beta_0, \epsilon} L(\beta, \beta_0, \epsilon, \mu) \leq L(\beta, \beta_0, \epsilon, \mu).$$ \hspace{1cm} (18)
Thus to find the optimal $L(\beta, \beta_0, \epsilon, \mu)$, we want to maximize $L_D(\alpha)$. We will do so under several constraints, including $0 \leq \alpha_i \leq C$ from (16), $\sum_{i=1}^{n} \alpha_i y_i = 0$ from (15), and the following three constraints derived from the KKT conditions:

$$\alpha_i [y_i (x_i^T \beta + \beta_0) - (1 - \epsilon_i)] = 0, \quad (19)$$

$$\mu_i \epsilon_i = 0, \quad (20)$$

$$y_i (x_i^T \beta + \beta_0) - (1 - \epsilon_i) \geq 0, \forall i. \quad (21)$$

This is a simpler convex optimization problem.

From (14) we see the solution is (again) a linear combination of training instances,

$$\hat{\beta} = \sum_{i=1}^{n} \hat{\alpha}_i y_i x_i, \quad (22)$$

and based on (19) $\hat{\alpha}_i$ is non-zero only if

$$y_i (x_i^T \beta + \beta_0) - (1 - \epsilon_i) = 0, \quad (23)$$

which implies that

(1) if $\epsilon_i = 0$, then $y_i (x_i^T \beta + \beta_0) = 1$ which means $x_i$ lies on the margin (and the right side).

(2) if $\epsilon_i > 0$, then $y_i (x_i^T \beta + \beta_0) \leq 1$ which means $x_i$ lies inside the margin (either on the right side or wrong side). In addition, (16,20) imply $\hat{\alpha}_i = C$.

In other words, only three types of instances will contribute to the decision boundary: (1) instances that are correctly classified and lie on the margin, (2) instances that are correctly classified and lie inside the margin, (3) instances that are misclassified. We call these instances support vectors, and the classification method soft-margin linear SVM.