0.1 Support Vector Classifier (Soft-Margin Linear SVM)

Max margin classifier finds a slab such that one class lies on its one side while another class lies on the other. This is based on an assumption that two classes are linearly separable.

What if two classes are not linearly separable? One simple solution is to ask the slab to tolerate mis-behaved instances (see Figure 1). The modified method will thus maximize soft-margin, as compared with the original method that maximizes hard-margin.\(^1\)

How to model ‘tolerance’ mathematically? We can introduce a \(\epsilon_i\) slack variable for each \((x_i, y_i)\), and modify the original constraint \(y_i(x_i^T \beta + \beta_0) \geq M\) into

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

or

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

The modified constraint means \(x_i\) does not need to lie on the correct side of the slab; it can either lie inside the slab or lie on the wrong side of the slab. (see Figure 1)

![Figure 1: Separable Case (left) and Non-Separable Case (right)](image)

Clearly, we want the tolerance to be as small as possible e.g. it is desirable to minimize \(\sum_{i=1}^{n} \epsilon_i\).

Adding this concern to the (cleaned) optimal separating hyperplane, we have

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

s.t. \(y_i(x_i^T \beta + \beta_0) \geq 1 - \epsilon_i, \, \epsilon_i \geq 0, \, \forall i.\)

(3)

where \(C\) is a hyper-parameter controlling the degree of tolerance.

[Discussion] What happens if \(C = \infty\)?

How to solve (3)?

---

\(^1\)Hard-margin does not tolerate mis-behaved instances, while soft-margin does.

\(^2\)Both (1, 2) introduce tolerance. In practice (2) is preferred as it results in convex optimization problem.
Again, we can apply the Lagrange multiplier to get a dual problem. The Lagrange function is

\[ L(\beta, \beta_0, \epsilon, \mu) = \frac{1}{2}||\beta||^2 + C \sum_{i=1}^{n} \epsilon_i - \sum_{i=1}^{n} \alpha_i[y_i(x_i^T\beta + \beta_0) - (1 - \epsilon_i)] - \sum_{i=1}^{n} \mu_i\epsilon_i. \] (4)

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

\[ \beta = \sum_{i=1}^{n} \alpha_i y_i x_i \] (5)

\[ 0 = \sum_{i=1}^{n} \alpha_i y_i \] (6)

\[ \alpha_i = C - \mu_i, \] (7)

and \( \alpha_i, \mu_i, \epsilon_i \geq 0, \forall i. \)

Plugging (5), (6) and (7) back to (4), we have the Wolfe dual objective function

\[ L_D(\beta, \beta_0, \alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j. \] (8)

[Exercise] Verify (8).

We will maximize \( L_D \) under several constraints, including \( 0 \leq \alpha_i \leq C \) from (7), \( \sum_{i=1}^{n} \alpha_i y_i = 0 \) from (6), and three from the KKT conditions:

\[ \alpha_i[y_i(x_i^T\beta + \beta_0) - (1 - \epsilon_i)] = 0, \] (9)

\[ \mu_i\epsilon_i = 0, \] (10)

\[ y_i(x_i^T\beta + \beta_0) - (1 - \epsilon_i) \geq 0, \forall i. \] (11)

This is a simpler convex optimization problem.

From (5) we see the solution is (again) a linear combination of training instances:

\[ \hat{\beta} = \sum_{i=1}^{n} \hat{\alpha}_i y_i x_i, \] (12)

and, by (9), coefficient \( \hat{\alpha}_i \) is non-zero only if

\[ y_i(x_i^T\beta + \beta_0) - (1 - \epsilon_i) = 0. \] (13)

What we can say about instances with non-zero coefficients?

[1] if \( \epsilon_i = 0 \), then \( y_i(x_i^T\beta + \beta_0) = 1 \) thus \( x_i \) lies on the margin; so (10, 7) imply \( 0 < \hat{\alpha}_i < C \).

[2] if \( \epsilon_i > 0 \), then \( y_i(x_i^T\beta + \beta_0) \leq 1 \) thus \( x_i \) lies inside the slab\(^3\), so (10,7) imply \( \hat{\alpha}_i = C \).

Finally, we say \( x_i \) is a support vector if \( \hat{\alpha}_i \neq 0 \). This method is called support vector classifier.

[Reading] ELS, Chapter 12.2

\(^3\)This could mean \( x_i \) is on the right side of the boundary or on the wrong side of the boundary.
0.2 KKT Conditions

KKT conditions guarantee that solution of a constrained problem equals to the solution of its Lagrange dual problem (which is often easier to solve). Let’s see how they are developed.

Let’s start with a standard form of optimization problem

\[
\begin{align*}
\min & \quad f_0(x) \\
\text{s.t.} & \quad f_i(x) \leq 0, \quad i = 1, \ldots, m \\
& \quad h_j(x) = 0, \quad j = 1, \ldots, p.
\end{align*}
\]  

(14)

Any \( x \) satisfying all constraints is a feasible point of the problem. Let \( p^* \) be the optimal \( f_0(x) \).

Let’s augment the objective function with a weighted sum of the constraint functions i.e.

\[
L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{j=1}^{p} \nu_j h_i(x),
\]

(15)

where \( \lambda_i, \nu_j \) are Lagrange multipliers, and \( \lambda = \{ \lambda_i \} \) and \( \nu = \{ \nu_j \} \) are dual variables.

The Lagrange dual function is

\[
g(\lambda, \nu) = \min_L L(x, \lambda, \nu).
\]

(16)

An important result is Lagrange dual function lower bounds the optimal function value i.e.

\[
g(\lambda, \nu) \leq p^*,
\]

(17)

for any \( \lambda \geq 0 \) and any \( \nu \).

[Exercise] Verify (17).

A natural question is what is the best lower bound obtainable from the Lagrange dual function. This leads to the following optimization problem

\[
\begin{align*}
\max & \quad g(\lambda, \nu) \\
\text{s.t.} & \quad \lambda \geq 0.
\end{align*}
\]

(18)

(18) is the Lagrange dual problem associated with (14) – the primal problem.

Any \( (\lambda, \nu) \) satisfying \( \lambda \geq 0 \) and \( g(\lambda, \nu) > -\infty \) is a dual feasible.

Solution to (18), denoted by \( (\lambda^*, \nu^*) \), is called dual optimal.

[Remark] (18) is a convex optimization problem, whether or not (14) is convex. This is a potential benefit of working with dual problem instead of original problem.

Let \( d^* \) denote the optimal value of \( g(\lambda, \nu) \). The difference \( p^* - d^* \) is optimal duality gap.

If \( d^* = p^* \), we say strong duality holds. This does not hold in general; this often holds if the original problem is convex.

[Discussion] Is (3) convex?

\[4\text{We write } \lambda \geq 0 \text{ to indicate } \lambda \text{ is element-wise non-negative.}\]
Now, we are ready to introduce the KKT conditions.
Suppose strong duality holds. Suppose $x^*$ is primal optimal\(^5\) and $\lambda^*, \upsilon^*$ is dual optimal\(^6\).
We have
\[
 f_0(x^*) = g(\lambda^*, \upsilon^*) \\
= \inf_x f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{j=1}^p \upsilon_j h_i(x) \\
\leq f_0(x^*) + \sum_{i=1}^m \lambda_i f_i(x^*) + \sum_{j=1}^p \upsilon_j h_i(x^*) \\
\leq f_0(x^*).
\]

[Exercise] verify (19).
Since the left-hand-side must equal the right-hand-side, we conclude the two inequalities in (19) hold with equality. This gives two interesting insights:
(1) $x^*$ is a minimizer of $L(x, \lambda, \upsilon)$\(^7\)
(2) $\sum_{i=1}^m \lambda_i f_i(x^*) = 0$, which implies each $\lambda_i f_i(x^*) = 0$\(^8\)
In addition, since $L(x, \lambda, \upsilon)$ is minimized at $x^*$, its gradient should vanish at $x^*$ i.e.
\[
\nabla f_0(x^*) + \sum_{i=1}^m \lambda_i \nabla f_i(x^*) + \sum_{j=1}^p \upsilon_j \nabla h_i(x^*) = 0.
\]

Putting all together, we see the primal optimal point $x^*$ should satisfy
\[
f_i(x^*) \leq 0 \tag{21}
\]
\[
h_i(x^*) = 0 \tag{22}
\]
\[
\lambda_i \geq 0 \tag{23}
\]
\[
\lambda_i f_i(x^*) = 0 \tag{24}
\]
\[
\nabla f_0(x^*) + \sum_{i=1}^m \lambda_i \nabla f_i(x^*) + \sum_{j=1}^p \upsilon_j \nabla h_i(x^*) = 0 \tag{25}
\]
The above are called KKT conditions.

[Exercise] Verify (9) and (10) are indeed KKT conditions.

[Discussion] Where do we apply condition (20) in analyzing linear SVM?

[Reading] CXO, Chapter 5.1, 5.2, 5.5

\(^5\)It achieves optimum in (14).
\(^6\)It achieves optimum in (18).
\(^7\)This justifies why it makes sense to solve $\min_x L(x, \lambda, \upsilon)$ and plug result back to $L$.
\(^8\)This is called the complementary slackness condition.