Linear Support Vector Machines

David S. Rosenberg

1 The Support Vector Machine

For a linear support vector machine (SVM), we use the hypothesis space of affine functions

$$\mathcal{F} = \left\{ f(x) = w^T x + b \mid w \in \mathbf{R}^d, b \in \mathbf{R} \right\}$$

and evaluate them with respect to the **SVM loss function**, also known as the **hinge loss**. The hinge loss is a margin-based loss defined as $\ell(m) = (1-m)_+$, where $m = yf(x)$ is the margin for the prediction function f on the example (x, y), and $(x)_+ = x1(x \geq 0)$ denotes the "positive part" of x. The SVM traditionally uses an $\ell_2$regularization term, and the objective function is written as

$$J(w,b) = \frac{1}{2}||w||^2 + \frac{c}{n}\sum_{i=1}^{n}\left(1 - y_i\left[w^T x_i + b\right]\right)_+ .$$

Note that the w parameter is regularized, while the bias term b is not regularized. An alternative approach (which saves some writing), is to drop the b and add a a constant feature, say with the value 1, to the representation of x. With this approach, the bias term will be regularized along with the rest of the parameters.

Rather than the typical λ regularization parameter attached to the ℓ_2 penalty, for SVMs it's traditional to have a "c" parameter attached to the empirical risk component. The larger c is, the more relative importance we attach to minimizing the empirical risk compared to finding a "simple" hypothesis with small ℓ_2-norm.

2 Formulating SVM as a QP

The SVM optimization problem is

$$\min_{w\in\mathbf{R}^d, b\in\mathbf{R}} \frac{1}{2}||w||^2 + \frac{c}{n}\sum_{i=1}^{n}\left(1 - y_i\left[w^T x_i + b\right]\right)_+ . \tag{2.1}$$

This is an unconstrained optimization problem (which is nice), but the objective function is not differentiable, which makes it difficult to work with. We can formulate an equivalent problem with a differentiable objective, but we'll have to add new constraints to do so. Note that 2.1is equivalent to

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}||w||^2 + \frac{c}{n}\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & \xi_i \geq \left(1 - y_i\left[w^T x_i + b\right]\right)_+ , \end{aligned}$$

since the minimization will always drive down ξ_i until $\xi_i^* = \left(1 - y_i\left[w^T x_i + b\right]\right)_+$. We can now break up the inequality into two parts:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}||w||^2 + \frac{c}{n}\sum_{i=1}^{n}\xi_i \\ \text{subject to} \quad & \xi_i \geq 0 \text{ for } i = 1, \ldots, n \\ & \xi_i \geq \left(1 - y_i\left[w^T x_i + b\right]\right) \text{ for } i = 1, \ldots, n \end{aligned}$$

We now have a differentiable objective function in $d + 1 + n$ variables with $2n$ affine constraints. This is a quadratic program that can be solved by any off-the-shelf QP solver.

3 Compute the Lagrangian Dual

The Lagrangian for this formulation is

$$\begin{aligned} L(w, b, \xi, \alpha, \lambda) =& \frac{1}{2}||w||^2 + \frac{c}{n}\sum_{i=1}^{n}\xi_i + \sum_{i=1}^{n}\alpha_i\left(1 - y_i\left[w^T x_i + b\right] - \xi_i\right) - \sum_{i=1}^{n}\lambda_i\xi_i \\ =& \frac{1}{2}w^T w + \sum_{i=1}^{n}\xi_i\left(\frac{c}{n} - \alpha_i - \lambda_i\right) + \sum_{i=1}^{n}\alpha_i\left(1 - y_i\left[w^T x_i + b\right]\right). \end{aligned}$$

From our study of Lagrangian duality, we know that the original problem can now be expressed as

$$\inf_{w,b,\xi} \sup_{\alpha,\lambda\succeq 0} L(w,b,\xi,\alpha,\lambda).$$

Since our constraints are affine, by Slater's condition we have strong duality so long as the problem is feasible (i.e. so long as there is at least one point in the feasible set). The constraints are satisfied by $w = 0$ and $\xi_i = 1$ for $i = 1,\dots,n$, so **we have strong duality**. Thus we get the same result if we solve the following dual problem:

$$\sup_{\alpha,\lambda\succeq 0} \inf_{w,b,\xi} L(w,b,\xi,\alpha,\lambda).$$

As usual, we capture the inner optimization in the Lagrange dual objective: $g(\alpha,\lambda) = \inf_{w,\xi} L(w,b,\xi,\alpha,\lambda)$. Note that if $\frac{c}{n} - \alpha_i - \lambda_i \neq 0$, then the Lagrangian is unbounded below (by taking $\xi_i \to \pm\infty$) and thus the infimum is $-\infty$. For any given (α,λ), the function $(w,\xi) \mapsto L(w,b,\xi,\alpha,\lambda)$ is differentiable and convex, thus we have an optimal point if and only if all partial derivatives of L with respect to w, b, and ξ are 0:

$$\begin{aligned}
\partial_w L = 0 &\iff w - \sum_{i=1}^n \alpha_i y_i x_i = 0 \iff w = \sum_{i=1}^n \alpha_i y_i x_i && (3.1)\\
\partial_b L = 0 &\iff -\sum_{i=1}^n \alpha_i y_i = 0 \iff \sum_{i=1}^n \alpha_i y_i = 0 &&\\
\partial_{\xi_i} L = 0 &\iff \frac{c}{n} - \alpha_i - \lambda_i = 0 \iff \alpha_i + \lambda_i = \frac{c}{n} && (3.2)
\end{aligned}$$

Note that one of the conditions is $\alpha_i + \lambda_i = \frac{c}{n}$, which agrees with our previous observation that if $\alpha_i + \lambda_i \neq \frac{c}{n}$ then L is unbounded below.

Substituting these conditions back into L, the second term disappears, while the first and third terms become

$$\begin{aligned}
\frac{1}{2} w^T w &= \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j\\
\sum_{i=1}^n \alpha_i (1 - y_i \left[w^T x_i + b\right]) &= \sum_{i=1}^n \alpha_i - \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j x_j^T x_i - b \underbrace{\sum_{i=1}^n \alpha_i y_i}_{=0}.
\end{aligned}$$

In the last expression, we see that if $\sum_{i=1}^n \alpha_i y_i \neq 0$, then by sending b to $\pm\infty$, we can send the dual function to $-\infty$.

Putting it together, the dual function is

$$g(\alpha,\lambda) = \begin{cases} \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j x_j^T x_i & \sum_{i=1}^n \alpha_i y_i = 0,\ \alpha_i + \lambda_i = \frac{c}{n}, \text{ all } i \\ -\infty & \text{otherwise.} \end{cases}$$

Thus we can write the dual problem as

$$\begin{aligned} \sup_{\alpha,\lambda} & \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j x_j^T x_i \\ \text{s.t.} & \sum_{i=1}^n \alpha_i y_i = 0 \\ & \alpha_i + \lambda_i = \frac{c}{n},\ i = 1,\ldots,n \\ & \alpha_i, \lambda_i \geq 0,\ i = 1,\ldots,n \end{aligned}$$

We can actually eliminate the λ variables, replacing the last three constraints by $0 \leq \alpha_i \leq \frac{c}{n}$:

$$\begin{aligned} \sup_{\alpha} & \sum_{i=1}^n \alpha_i - \frac{1}{2}\sum_{i,j=1}^n \alpha_i\alpha_j y_i y_j x_j^T x_i \\ \text{s.t.} & \sum_{i=1}^n \alpha_i y_i = 0 \\ & \alpha_i \in \left[0, \frac{c}{n}\right]. \end{aligned}$$

When written in standard form, this has a quadratic objective in n unknowns and $2n+1$ constraints. The constraints of the form $\alpha_i \in \left[0, \frac{c}{n}\right]$ are called **box constraints**, and are particularly easy to handle in optimization.

Let (w^*, b^*, ξ^*) be a solution to the primal problem, and let (α^*, λ^*) be a solution to the dual problem. Then by strong duality and since everything is differentiable, the solutions must obey the KKT conditions. In particular, the solutions must satisfy the first order condition we derived in Eq. (3.1). So

$$w^* = \sum_{i=1}^n \alpha_i^* y_i x_i.$$

- Note that $w^* = \sum_{i=1}^n \alpha_i^* y_i x_i$ only depends on those examples for which $\alpha_i^* > 0$ (recall that $\alpha_i^* \geq 0$ by constraint). These examples are called **support vectors**.

- Since $\alpha_i \in [0, \frac{c}{n}]$, we see that c controls the amount of weight we can put on any single example.

Note that we still don't have an expression for the optimal bias term b^*. We'll derive this below using complementary slackness conditions.

4 Consequences of Complementary Slackness

Let (w^*, b^*, ξ_i^*) and (α^*, λ^*) be optimal solutions to the primal and dual problems, respectively. For notational convenience, let's define $f^*(x) = x_i^T w^* + b^*$. By strong duality, we have the following **complementary slackness** conditions:

$$\alpha_i^* \left(1 - y_i f^*(x_i) - \xi_i^*\right) = 0 \tag{4.1}$$

$$\lambda_i^* \xi_i^* = \left(\frac{c}{n} - \alpha_i^*\right) \xi_i^* = 0 \tag{4.2}$$

We now draw many straightforward conclusions:

- As we noted above, ξ_i^* is the hinge loss on example i. When $\xi_i^* = 0$, we're either "at the margin" (i.e. $y_i f^*(x_i) = 1$) or on the "good side of the margin" ($y_i f^*(x_i) > 1$). That is

$$\xi_i^* = 0 \implies y_i f^*(x_i) \geq 1. \tag{4.3}$$

- By (4.2), $\alpha_i^* = 0$ implies $\xi_i^* = 0$, which by (4.3) implies $y_i f^*(x) \geq 1$.

- $\alpha_i^* \in \left(0, \frac{c}{n}\right)$ implies $\xi_i^* = 0$, by 4.2. Then by 4.1 we get $y_i f^*(x_i) = 1$. So the prediction is right on the margin.

- If $y_i f^*(x_i) < 1$ then the margin loss is $\xi_i^* > 0$, and (4.2) implies that $\alpha_i^* = \frac{c}{n}$.

- If $y_i f^*(x) > 1$ then the margin loss is $\xi_i^* = 0$, and 4.1 implies $\alpha_i^* = 0$.

- The contrapositive of the previous result is that $\alpha_i^* > 0$ implies $y_i f^*(x) \leq 1$. This seems to be all we can say for the specific case $\alpha_i^* = \frac{c}{n}$.

- We also can't draw any extra information about α_i^* for points exactly on the margin ($y_i f^*(x_i) = 1$).

We summarize these results below:

$$\begin{aligned}
\alpha_i^* = 0 &\implies y_i f^*(x_i) \geq 1 \\
\alpha_i^* \in \left(0, \frac{c}{n}\right) &\implies y_i f^*(x_i) = 1 \\
\alpha_i^* = \frac{c}{n} &\implies y_i f^*(x_i) \leq 1
\end{aligned}$$

$$\begin{aligned}
y_i f^*(x_i) < 1 &\implies \alpha_i^* = \frac{c}{n} \\
y_i f^*(x_i) = 1 &\implies \alpha_i^* \in \left[0, \frac{c}{n}\right] \\
y_i f^*(x_i) > 1 &\implies \alpha_i^* = 0
\end{aligned}$$

4.1 Determining b

Finally, let's determine b. Suppose there exists an i such that $\alpha_i^* \in \left(0, \frac{c}{n}\right)$. Then $\xi_i^* = 0$ by (4.2) and by (4.1) we get $y_i \left[x_i^T w^* + b^*\right] = 1$. Since $y_i \in \{-1, 1\}$, we can multiply on both sides by y_i to conclude that

$$b^* = y_i - x_i^T w^*.$$

With exact calculations, we would get the same b^* for any choice of i with $\alpha_i^* \in \left(0, \frac{c}{n}\right)$. With numerical error, however, some people suggest averaging over all eligible i's:

$$b^* = \text{mean}\left\{y_i - x_i^T w^* \mid \alpha_i^* \in \left(0, \frac{c}{n}\right)\right\}.$$

If there are no $\alpha_i^* \in \left(0, \frac{c}{n}\right)$, then we have a **degenerate SVM training problem**[1], for which $w^* = 0$, and we always predict the majority class.

[1] This is shown in Rifkin et al.'s "A Note on Support Vector Machine Degeneracy", an MIT AI Lab Technical Report.