SVM: Main Takeaways from Duality

David S. Rosenberg

Abstract

A traditional presentation on SVM can be a bit brutal, as it typically includes a development of convex optimization and Lagrangian duality. In this short note, we first recap the setup and the results derived in such a lecture, and in the last section we'll highlight the practical takeaways.

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 $\ell(m) = \max(0, 1 - m)$, where $m = yf(x)$ is the margin for the prediction function f on the example (x, y). The SVM traditionally uses an ℓ_2 regularization term, and the objective function is written as

$$J(w, b) = \frac{1}{2}||w||^2 + \frac{c}{n} \sum_{i=1}^{n} \max\left(0, 1 - y_i \left[w^T x_i + b\right]\right).$$

Note that the w parameter is regularized, while the bias term b is not regularized.

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 Lagrangian Duality: What we did

We reformulated the SVM optimization problem as a quadratic program, and then we found the dual optimization problem:

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

We noted that the primal problem satisfies Slater's condition, and thus we have strong duality. This allowed us to find a relationship between the primal optimal solution and the dual optimal solution:

$$\begin{aligned} w^* &= \sum_{i=1}^{n} \alpha_i^* y_i x_i \\ b^* &= y_j - x_j^T w^*, \end{aligned}$$

where j is any index for which $\alpha_i^* \in \left(0, \frac{c}{n}\right)$.

We then applied the complementary slackness conditions (guaranteed by strong duality) to derive the following relations between the margin of a training point (x_i, y_i) and the corresponding weight for that training point α_i^*, in the expression for w^*:

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

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

3 Key Takeaways

1. The solution $w^* = \sum_{i=1}^n \alpha_i^* y_i x_i$ is a linear combination of the training input vectors $x_1, \ldots, x_n$. People often say that w^* is **in the span of the data.** While this is **not** unique to SVMs, and there are much simpler ways to derive this result (such as with basic linear algebra via the representer theorem), we need this to understand some of the other takeaways.

2. While finding w^* to be in the span of the data is common with linear methods, with SVMs we find that the expansion is often **sparse** in the data. In other words, many of the α_i^*'s may be exactly 0. The complementary slackness conditions tell us exactly when this happens: it is guaranteed to happen for any training point with $y_i f^*(x_i) > 1$ (i.e. on the "good side of the margin") and may also happen with $y_i f^*(x_i) = 1$ (exactly on the margin). The x_i's that have nonzero coefficients (i.e. $\alpha_i^* > 0$) are called **support vectors.** The sparsity of the support vectors becomes more important when we introduce the "kernelized SVM", for which we need to store all the support vectors to make new predictions. So sparsity can be important when we have a very large training set.

3. The amount of weight we can put on any single example in the final solution is controlled by c, since $\alpha_i^* \in \left[0, \frac{c}{n}\right]$. So, in a certain sense, no single training point can have too much influence on the final solution. However, we shouldn't read too much into this. Note that a single training point can still dominate the expression for w^* just by being very far away from the other points in input space. To investigate: How does the influence of a single extreme training point on w^* change if we use square loss rather than hinge loss?