

Support Vector Machines

Guy Lebanon

February 26, 2007

Support vector machines (SVM) are binary linear classifiers for data residing in $X \times \{+1, -1\}$. While X is arbitrary, the data is represented through a feature mapping $\Phi : X \rightarrow \mathcal{H}$, where \mathcal{H} is a Hilbert space (an inner product normed space). \mathcal{H} can be finite dimensional such as \mathbb{R}^d or infinite dimensional such as $L^p(\Omega)$. Representing the inner product in \mathcal{H} using $\langle \cdot, \cdot \rangle$ we have that $f(x) = \text{sign}\langle w, \Phi(x) \rangle$. We omit the bias or constant term above since it can be incorporated into the simpler notation above using $\Phi'(x) = (\Phi(x), 1), w' = (w, 1)$. Below, we will illustrate SVMs in the special case of $\mathcal{H} = \mathbb{R}^d$. The ideas, however, carry over to more general Hilbert space \mathcal{H} .

Linearly Separable Case

The linear classifier $f(x) = \text{sign}\langle w, \Phi(x) \rangle$ is parameterized by a weight vector which is normal to the decision boundary hyperplane. Since there is no bias, the decision hyperplane passes through the origin (it is actually a linear subspace) and any point $\Phi(x)$ can be represented as a sum of its projection onto the hyperplane and its perpendicular component $\Phi(x)_\perp + \Phi(x)_\parallel = \Phi(x)_\parallel + r \frac{w}{\|w\|}$. Since

$$\langle w, \Phi(x) \rangle = \langle w, \Phi(x)_\parallel \rangle + \langle w, r w / \|w\| \rangle = 0 + r \|w\| \Rightarrow r = \langle w, \Phi(x) \rangle / \|w\|,$$

we have that for correctly classified points (x_i, y_i) the distance to the hyperplane is $|r_i| = y_i \langle w, \Phi(x_i) \rangle / \|w\|$. The idea of support vector machines in the context of linearly separable data is to choose w that leads to the largest margin - defined as the distance of the closest data point to the hyperplane i.e.

$$w_{\text{svm}} = \arg \max_{w \in \mathbb{R}^d} \left(\|w\|^{-1} \min_{1 \leq i \leq n} y_i \langle w, \Phi(x_i) \rangle \right). \quad (1)$$

Since the direct solution of (1) is difficult, the SVM optimization problem is usually converted to an equivalent form that is easier to solve. We start by observing that rescaling the weight vector $w' = cw, c \in \mathbb{R}_+$ leaves the classifier $f(x)$ unchanged and does not change the distance r of points to the hyperplane. More importantly, it also leaves the objective function in (1) unchanged. By rescaling w so that the distance of the closest point to the hyperplane is 1 we get that $\min_{1 \leq i \leq n} y_i \langle w, \Phi(x_i) \rangle \geq 1$ with the minimum being actually achieved for one or more training points. This leads to the canonical or primal representation of SVM

$$w_{\text{svm}} = \arg \min_{w \in \mathbb{R}^d} \frac{1}{2} \|w\|^2 \quad \text{subject to} \quad y_i \langle w, \Phi(x_i) \rangle \geq 1 \quad \forall i. \quad (2)$$

Problem (2) is a quadratic program (minimization of a quadratic function subject to linear constraints) and is easier to solve than (1). However, it involves a large number of linear inequality constraints. The dual problem which is yet another equivalent SVM formulation is the easiest to solve computationally. It is obtained by forming the Lagrangian of the primal problem (2)

$$\mathcal{L}(w, \lambda) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^n \lambda_i (y_i \langle w, \Phi(x_i) \rangle - 1) \quad (3)$$

and maximizing the dual function $h(\lambda) = \inf_w \mathcal{L}(w, \lambda)$. To compute the dual function we optimize \mathcal{L} with respect to w and substitute the optimal value of \mathcal{L} w.r.t w

$$\forall i \quad 0 = \frac{\partial \mathcal{L}(w, \lambda)}{\partial w_i} \Rightarrow w^* = \sum_{j=1}^n \lambda_j y_j \Phi(x_j) \quad (4)$$

in \mathcal{L} to obtain

$$\begin{aligned} h(\lambda) &= \inf_w \mathcal{L}(w, \lambda) = \mathcal{L}(w^*, \lambda) = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \lambda_j \lambda_k y_j y_k \langle \Phi(x_j), \Phi(x_k) \rangle - \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \langle \Phi(x_i), \Phi(x_j) \rangle + \sum_{i=1}^n \lambda_i \\ &= \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \langle \Phi(x_i), \Phi(x_j) \rangle. \end{aligned} \quad (5)$$

We have thus shown, using convex duality that the equivalent dual formulation of SVM is

$$\lambda_{\text{svm}} = \arg \max_{\lambda \in \mathbb{R}^d} \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \langle \Phi(x_i), \Phi(x_j) \rangle \quad \text{subject to} \quad \lambda_i \geq 0. \quad (6)$$

Notice that (6) is also a quadratic program, but in contrast to (2) the constraints are much simpler. Instead of solving (2) and plugging in the resulting w_{svm} in $f(x) = \text{sigm}(w_{\text{svm}}, \Phi(x))$, we can solve (6) and use

$$\langle w, \Phi(x) \rangle = \left\langle \sum_{j=1}^n \lambda_j y_j \Phi(x_j), \Phi(x) \right\rangle = \sum_{j=1}^n \lambda_j y_j \langle \Phi(x_j), \Phi(x) \rangle. \quad (7)$$

to express the SVM classifier using the solution λ_{svm} of (6)

$$f(x) = \text{sign} \langle w_{\text{svm}}, \Phi(x) \rangle = \text{sign} \sum_{j=1}^n [\lambda_{\text{svm}}]_j y_j \langle \Phi(x_j), \Phi(x) \rangle. \quad (8)$$

Which of the two expressions above should we use? Besides the fact that the constraint in (6) is simpler than the constraints of (2) we have another fundamental difference. The optimization (2) involves d variables w while the optimization (6) involves n variables λ . For high dimensional problems λ is of lower dimensionality than w motivating the use of the dual problem. Furthermore, often many training examples x_i are not support vectors (closest to the SVM solution) and therefore the corresponding inequality constraints are inactive (they hold with a strict inequality) and the corresponding variables zero out $[\lambda_{\text{svm}}]_j = 0$. As a result (6) is often a much lower dimensional optimization problem ($\lambda_j \rightarrow 0$ early on in the optimization) and computing the classifier using the dual variables $f(x) = \text{sign} \sum_{j=1}^n [\lambda_{\text{svm}}]_j y_j \langle \Phi(x_j), \Phi(x) \rangle$ can be done by summing only over the support vectors.

Non-Separable Case

Thus far, we have assumed that the training data $\{(\Phi(x_i), y_i) : i = 1, \dots, n\}$ is linearly separable (can be correctly classified using a linear decision surface). We proceed as before in the separable case, only that this time some examples may be on the wrong side of the hyperplane. We scale w so that all correctly classified examples satisfy $y \langle w, \Phi(x_i) \rangle \geq 1$ with equality holding for the closest correctly classified points from the hyperplane. Due to the re-scaling of w the margin is again related inversely to $\|w\|^2$. However, (2) need to be changed to incorporate the miss-classified training points leading to a new primal problem

$$(w_{\text{svm}}^{\text{ns}}, \xi_{\text{svm}}^{\text{ns}}) = \arg \min_{w \in \mathbb{R}^d, \xi \in \mathbb{R}^n} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \quad \text{subject to} \quad y_i \langle w, \Phi(x_i) \rangle \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad \forall i. \quad (9)$$

Above, ξ_i are slack variables measuring the degree of deviation of a point from the correct side of the margin. The parameter $C \geq 0$ is a regularization parameter that control the relative size of the two quantities. It fulfills a role of controlling the bias variance trade-off in statistics. In the case $C \rightarrow \infty$ we obtain the separable formulation of SVM (zero tolerance for miss-classification of training points).

Above, we wrote the minimization problem in terms of w as well as ξ . However, since we want small ξ_i (in order to minimize the objective function) we can easily determine the ξ variables based on w

$$\xi_i = (1 - y_i \langle w, \Phi(x_i) \rangle)_+ \stackrel{\text{def}}{=} \max(0, 1 - y_i \langle w, \Phi(x_i) \rangle) = \begin{cases} 0 & y \langle w, \Phi(x_i) \rangle \geq 1 \\ |y_i - \langle w, \Phi(x_i) \rangle| = |1 - y_i \langle w, \Phi(x_i) \rangle| & y \langle w, \Phi(x_i) \rangle < 1 \end{cases}$$

Proceeding as before to the dual formulation, the Lagrangian of (9) is

$$\mathcal{L}(w, \xi, \lambda, \mu) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i - \sum_{i=1}^n \lambda_i (y_i \langle w, \Phi(x_i) \rangle - 1 + \xi_i) - \sum_{i=1}^n \mu_i \xi_i \quad \lambda_i, \mu_i \geq 0. \quad (10)$$

To find the dual function $h(\lambda, \mu) = \inf_{w, \xi} \mathcal{L}(w, \xi, \lambda, \mu)$ we need to set the $\nabla_w \mathcal{L} = 0$ and $\nabla_{\xi} \mathcal{L} = 0$ and substitute the resulting expressions in \mathcal{L} . $\nabla_w \mathcal{L} = 0$ is the same as the Lagrangian gradient we computed for the separable case. Differentiating w.r.t. ξ_i and setting it to 0 we obtain $\lambda_i = C - \mu_i$ which together with the constraint $\mu_i \geq 0$ leads to $\lambda_i \in [0, C]$. Substituting the resulting expressions in \mathcal{L} we obtain

$$h(\lambda, \mu) = \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \langle \Phi(x_i), \Phi(x_j) \rangle \quad (11)$$

which is the same dual function we obtained in the separable case. The dual problem $\max h(\lambda, \mu) = \max h(\lambda)$ no longer involves μ and is identical to the (6) except for the box constraints on λ

$$\lambda_{\text{svm}}^{\text{ns}} = \arg \max_{\lambda \in \mathbb{R}^d} \sum_{i=1}^n \lambda_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \langle \Phi(x_i), \Phi(x_j) \rangle \quad \text{subject to} \quad 0 \leq \lambda_i \leq C \quad \forall i. \quad (12)$$

As before, the obtained $\lambda_{\text{svm}}^{\text{ns}}$ can be translated to $w_{\text{svm}}^{\text{ns}}$ for use in the classifier using (4), or simply be used to compute the classifier via (8).

Hinge Loss Interpretation

An interesting observation is that the SVM problem (9) can be written as

$$\min_{w \in \mathbb{R}^d} \sum_{i=1}^n l_{\text{hinge}}(y_i \langle w, \Phi(x_i) \rangle) + c \|w\|^2$$

(recall that the relation between ξ and w) where $l_{\text{hinge}}(z) = (1 - z)_+$. Viewed in this way, we see a striking similarity between SVM and various penalized likelihood methods such as regularized logistic regression and boosting. The function $l_{\text{hinge}}(z)$, called the hinge loss, represents an empirical loss and replaces the logistic regression negative loglikelihood

$$l_{\text{nl}}(z) = \log(1 + \exp(-z)) = -\log p(y_i | x_i) = -\log \frac{\exp(y \langle w, \Phi(x) \rangle / 2)}{\exp(-y \langle w, \Phi(x) \rangle / 2) + \exp(y \langle w, \Phi(x) \rangle / 2)}$$

or Adaboost's exponential loss $l_{\text{exp}}(z) = \exp(-z)$. The term $\|w\|^2$ represents regularization penalty analogous or MAP under the log of a Gaussian prior.