Linear Support Vector Machines

Guy Lebanon

Support vector machines (SVM) are currently the best performing general purpose classifier. We describe in this note linear SVM. Non-linear SVMs will be described in a future note on kernels. We assume binary classification with $Y \in \{+1, -1\}$ and represent the classifier using the inner product notation $\hat{Y} = \text{sign}\langle w, X\rangle$ where $\langle x, z\rangle = x^\top z$ in vector notation. Note that a bias term may be included i.e. $\hat{Y} = \text{sign}(w_0 + \langle w, X\rangle)$ using the notation $\langle w, X\rangle$ if X is augmented with an always one component. Similarly, note that X may be a vector of non-linear features of the actual data X' i.e. $X_1 = X'_1 X'^2_2$ so the linear classifier in the space of X is really non-linear in the original data space of X'.

Linearly Separable Case

The linear classifier $f(X) = \text{sign}\langle w, X\rangle$ is parameterized by a weight vector which is normal (i.e. perpendicular) to the decision boundary which is a subspace or a linear hyperplane passing through the origin (note that as described above this does not preclude having a bias term). Any X can be represented as a sum of its projection onto the subspace and its perpendicular component $X = X_\perp + X_\parallel = X_\parallel + r\frac{w}{\|w\|}$. Since

$$\langle w, X\rangle = \langle w, X_\parallel\rangle + \langle w, rw/\|w\|\rangle = 0 + r\|w\| \quad \Rightarrow \quad r = \langle w, X\rangle/\|w\|,$$

we have that for correctly classified points $X^{(i)}, Y^{(i)}$ the distance to the hyperplane is $|r_i| = Y^{(i)}\langle w, X^{(i)}\rangle/\|w\|$. 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

$$w = \arg\max_{w\in\mathbb{R}^d} \left(\|w\|^{-1} \min_{1\le i\le n} Y^{(i)}\langle w, X^{(i)}\rangle \right). \tag{1}$$

The direct solution of (1) is difficult as the objective function is non-differentiable. We proceed by solving an equivalent optimization problem 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 subspace. More importantly, it also leaves the objective function (1) unchanged. By rescaling w so that the distance of the closest point to the hyperplane is 1 we get that $\min_{1\le i\le n} Y^{(i)}\langle w, X^{(i)}\rangle \ge 1$ with the minimum achieved for one or more training points

$$w = \arg\min_{w\in\mathbb{R}^d} \frac{1}{2}\|w\|^2 \quad \text{subject to} \quad Y^{(i)}\langle w, X^{(i)}\rangle \ge 1 \quad i = 1, \ldots, n. \tag{2}$$

A different way to see the equivalence between (1) and (2) is to note that as w gets closer to the origin (as we try to do by minimizing $\|w\|$) one or more of the constraints will be satisfied with equality rather than inequality in which case (1) and (2) are equivalent (i.e., at the solution one or more of the constraints must be active with equality).

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 optimizing the Lagrangian

$$\mathcal{L}(w, \lambda) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \lambda_i (Y^{(i)}\langle w, X^{(i)}\rangle - 1) \tag{3}$$

with respect to w first, substituting the solution into $\mathcal{L}$ and then optimizing with respect to λ (recall for a point to be a solution of the constrained optimization problem both $\nabla_w \mathcal{L}$ and $\nabla_\lambda \mathcal{L}$ need to be zero).

$$0 = \frac{\partial \mathcal{L}(w,\lambda)}{\partial w_i} \quad i = 1,\ldots,n \quad \Rightarrow \quad w^* = \sum_{j=1}^{n} \lambda_j^{(j)} X^{(j)} \tag{4}$$

$$\begin{aligned}\mathcal{L}(\lambda, w^*) &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i\lambda_j Y^{(i)}Y^{(j)}\langle X^{(i)}, X^{(j)}\rangle - \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i\lambda_j Y^{(i)}Y^{(j)}\langle X^{(i)}, X^{(j)}\rangle + \sum_{i=1}^{n}\lambda_i \\ &= \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i\lambda_j Y^{(i)}Y^{(j)}\langle X^{(i)}, X^{(j)}\rangle. \end{aligned} \tag{5}$$

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

$$f(X) = \text{sign}\langle w, X\rangle = \text{sign}\Big\langle \sum_{j=1}^{n} \lambda_j Y^{(j)} X^{(j)}, X \Big\rangle = \text{sign}\sum_{j=1}^{n} \lambda_j Y^{(j)}\langle X^{(j)}, X\rangle \quad \text{where} \tag{6}$$

$$\lambda = \arg\max_{\lambda\in\mathbb{R}^d} \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i\lambda_j Y^{(i)}Y^{(j)}\langle X^{(i)}, X^{(j)}\rangle \quad \text{subject to} \quad \lambda_i \geq 0. \tag{7}$$

Notice that (6) is also a quadratic program, but in contrast to (2) the constraints are much simpler. Also the number of variables and constrains are n rather than d (data dimensionality) which favors (6) further in high dimensional cases. Furthermore, the solution of (6) will have non-zero λ_j only for constraints that hold with equality at the optimum (the corresponding training points are called support vectors). Thus, many of the λ_j will converge early on to zero effectively reducing the dimensionality of (6).

Non-Separable Case

Thus far, we have assumed that the training data 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. In these cases we introduce ξ_i that measure the amount of violation in the constraints $Y^{(i)}\langle w, X^{(i)}\rangle \geq 1$:

$$(w,\xi) = \arg\min_{w\in\mathbb{R}^d, \xi\in\mathbb{R}^n} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \quad \text{subject to} \quad Y^{(i)}\langle w, X^{(i)}\rangle \geq 1-\xi_i, \quad \xi_i \geq 0 \quad i = 1,\ldots,n. \tag{8}$$

The parameter $C \geq 0$ is a regularization parameter controlling the relative importance of the two terms: maximizing margin for correctly classified examples and minimizing the misclassification penalty $\sum \xi_i$. Since we want small ξ_i (in order to minimize the objective function) we can easily determine $\xi_i = \max(0, 1 - Y^{(i)}\langle w, X^{(i)}\rangle)$ and remove ξ_i as optimization variables

$$w = \arg\min_{w\in\mathbb{R}^d} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\max(0, 1 - Y^{(i)}\langle w, X^{(i)}\rangle). \tag{9}$$

Proceeding as before the dual formulation leads to

$$\lambda = \arg\max_{\lambda\in\mathbb{R}^d} \sum_{i=1}^{n}\lambda_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i\lambda_j Y^{(i)} y_j \langle \Phi(X^{(i)}), \Phi(x_j)\rangle \quad \text{subject to} \quad 0 \leq \lambda_i \leq C \quad i = 1,\ldots,n \tag{10}$$

(with the classifier $f(X)$ given in terms of the dual variables λ as before).

Hinge Loss Interpretation

An interesting observation is that (9) can be interpreted as L_2 regularized margin based minimizier

$$w = \arg\min_{w} \sum_{i=1}^{n} l_{\text{hinge}}(Y^{(i)} \langle w, \Phi(X^{(i)}) \rangle) + c\|w\|^2$$

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

$$l_{\text{nll}}(z) = \log(1 + \exp(-z)) = -\log p(Y^{(i)}|X^{(i)}) = -\log \frac{\exp(y\langle w, x \rangle/2)}{\exp(-y\langle w, x/2 \rangle) + \exp(y\langle w, x \rangle/2)}$$

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