Lecture 10
Support Vector Machines II

22 February 2016

Taylor B. Arnold
Yale Statistics
STAT 365/665

Yale

Notes:

- Problem 3 is posted and due this upcoming Friday
- There was an early bug in the fake-test data; fixed as of 2016-02-20

Today:

- Optimization theory behind support vector machines
- More examples

Recall that we settled on the following definition for a the support vector machine:

$$\begin{aligned} \max_{||\beta||_2=1} \quad & M \\ \text{s.t.} \quad & y_i(x_i^t\beta + \beta_0) > M - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i > 0, \sum_i \xi_i \leq \text{Constant}. \end{aligned}$$

This defines a margin around the linear decision plane of width and tries to minimize the number of errors (ξ) for points that are on the wrong side of the margin.

We then re-parameterized this by setting the margin to 1 but allowing the size of β to grow:

$$\begin{aligned} \min \quad & \frac{1}{2}||\beta||_2^2 \\ \text{s.t.} \quad & y_i(x_i^t\beta + \beta_0) > 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i > 0, \sum_i \xi_i \leq \text{Constant}. \end{aligned}$$

This defines a margin around the linear decision plane of width $\frac{1}{||\beta||}$, and tried to minimize the number of errors ξ of points that are on the wrong side of the margin.

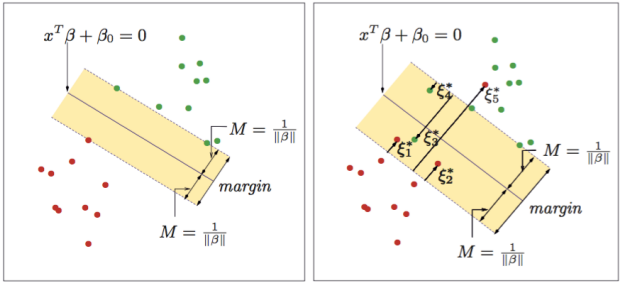


FIGURE 12.1. *Support vector classifiers. The left panel shows the separable case. The decision boundary is the solid line, while broken lines bound the shaded maximal margin of width* $2M = 2/\|\beta\|$. *The right panel shows the nonseparable (overlap) case. The points labeled* ξ_j^* *are on the wrong side of their margin by an amount* $\xi_j^* = M\xi_j$; *points on the correct side have* $\xi_j^* = 0$. *The margin is maximized subject to a total budget* $\sum \xi_i \leq$ *constant. Hence* $\sum \xi_j^*$ *is the total distance of points on the wrong side of their margin.*

Notice that we can rewrite

$$\begin{aligned} \min \quad & \frac{1}{2}||\beta||_2^2 \\ \text{s.t.} \quad & y_i(x_i^t\beta + \beta_0) > 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi_i > 0, \sum_i \xi_i \leq \text{Constant} \end{aligned}$$

With a constant $C > 0$, which depends only on the constant in the original formulation, as:

$$\begin{aligned} \min \quad & \frac{1}{2}||\beta||_2^2 + C \cdot \sum_i \xi_i \\ \text{s.t.} \quad & y_i(x_i^t\beta + \beta_0) > 1 - \xi_i, \quad \xi_i > 0, \quad i = 1, \ldots, n. \end{aligned}$$

By noticing that the second form will find a $\widehat{\beta}$ that minimizes $||\beta||_2^2$ such that $\sum_i \xi_i \leq \sum_i \widehat{\xi}_i$.

The Lagrangian

Given a constrained optimization problem:

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } g_j(x) = 0, \qquad j = 1, \ldots, K \end{aligned}$$

We can define the primal Lagrangian function as:

$$\mathcal{L}_P = f(x) - \sum_{j=1}^{K} \lambda_j g_j(x)$$

What does this function look like?

The Lagrangian Dual

The Lagrangian dual function is then given as the infimum of $\mathcal{L}_P$ as function of the λ_j over values of x:

$$\begin{aligned}\mathcal{L}_D(\lambda) &= \inf_x \mathcal{L}_P(x, \lambda) \\ &= \inf_x \left\{ f(x) - \sum_{j=1}^{K} \lambda_j g_j(x) \right\}.\end{aligned}$$

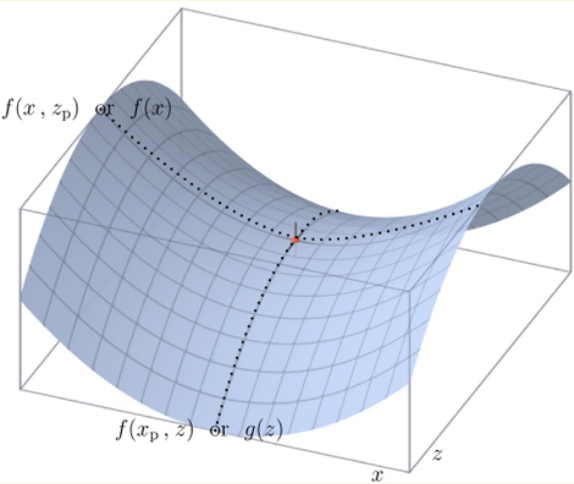
And the **dual problem** is to find the maximum of the dual function over all choices of λ:

$$\lambda^* = \arg\max_{\lambda} \mathcal{L}_D(\lambda).$$

The optimal value of the primal problem, x^*, can be reconstructed by working backwards:

$$x^* = \arg\min_{x} \mathcal{L}_P(x, \lambda^*).$$

For a better understanding of the dual problem we can visualize the Lagrangian solution as a saddle point[1]



[1] www.convexoptimization.com

It turns out that this is a very good framework for working with support vector machines. We can define the Lagrangian function as:

$$\mathcal{L}_P = \frac{1}{2}||\beta||_2^2 + C\sum_i \xi_i - \sum_i \alpha_i \left[y_i(x_i^t\beta + \beta_0) - (1 - \xi_i)\right] - \sum_i \mu_i\xi_i$$

Where α_i and μ_i are the Lagrangian multipliers.

Aside:

Technically, the theory of Lagrangian multipliers only apply when the constraints on the solution are equality constraints rather than inequality constraints. The larger theory needed for the general case uses the *Karush-Kuhn-Tucker* (KKT) conditions. These add additional constraints on top of those presented here. Following the Elements of Statistical Learning, we will not worry with those details here as they are more an annoyance than an interesting conceptual difference.

To construct the dual function, we need to take partial derivatives with respect to the primal variables: β, β_0, and ξ_i. If we plug these into the primal problem, we get the dual function.

For β_j:

$$\begin{aligned}
\frac{\partial \mathcal{L}_P}{\partial \beta_j} &= \frac{\partial}{\partial \beta_j}\left\{\frac{1}{2}||\beta||_2^2 + C\cdot\sum_i \xi_i - \sum_i \alpha_i\left[y_i(x_i^t\beta+\beta_0) - (1-\xi_i)\right] - \sum_i \mu_i\xi_i\right\} \\
&= \frac{\partial}{\partial \beta_j}\left\{\frac{1}{2}||\beta||_2^2 - \sum_i \alpha_i y_i(x_i^t\beta)\right\} \\
&= \beta_j - \sum_i \alpha_i y_i x_{i,j}
\end{aligned}$$

For β_j:

$$\begin{aligned}\frac{\partial \mathcal{L}_P}{\partial \beta_j} &= \frac{\partial}{\partial \beta_j}\left\{\frac{1}{2}||\beta||_2^2 + C\cdot\sum_i \xi_i - \sum_i \alpha_i\left[y_i(x_i^t\beta+\beta_0) - (1-\xi_i)\right] - \sum_i \mu_i\xi_i\right\} \\ &= \frac{\partial}{\partial \beta_j}\left\{\frac{1}{2}||\beta||_2^2 - \sum_i \alpha_i y_i(x_i^t\beta)\right\} \\ &= \beta_j - \sum_i \alpha_i y_i x_{i,j}\end{aligned}$$

Setting this equal to zero, and writing the equation simultaneously for all β_j, we get:

$$\beta = \sum_i \alpha_i y_i x_i$$

This necessary condition for the solution of the support vector machine is of independent interest. It says that β can be written as a linear combination of the data points x_i. Any i such that α_i is non-zero is called a support vector.

For β_0, the derivative is given as:

$$\begin{aligned}\frac{\partial \mathcal{L}_P}{\partial \beta_0} &= \frac{\partial}{\partial \beta_0}\left\{\frac{1}{2}||\beta||_2^2 + C\cdot\sum_i \xi_i - \sum_i \alpha_i\left[y_i(x_i^t\beta+\beta_0) - (1-\xi_i)\right] - \sum_i \mu_i\xi_i\right\} \\ &= \frac{\partial}{\partial \beta_0}\left\{-\sum_i \alpha_i y_i \beta_0\right\} \\ &= -\sum_i \alpha_i y_i\end{aligned}$$

Which when set to zero gives:

$$0 = \sum_i \alpha_i y_i$$

This explains why the term β_0 is often called the bias of the support vector machine.

Finally, the derivative with respect to ξ_i is given as:

$$\begin{aligned}
\frac{\partial \mathcal{L}_P}{\partial \xi_i} &= \frac{\partial}{\partial \xi_i}\left\{\frac{1}{2}||\beta||_2^2 + C\cdot\sum_i \xi_i - \sum_i \alpha_i\left[y_i(x_i^t\beta+\beta_0) - (1-\xi_i)\right] - \sum_i \mu_i\xi_i\right\} \\
&= \frac{\partial}{\partial \xi_i}\left\{C\cdot\sum_i \xi_i + \sum_i \alpha_i(1-\xi_i) - \sum_i \mu_i\right\} \\
&= C - \alpha_i - \mu_i
\end{aligned}$$

Finally, the derivative with respect to ξ_i is given as:

$$\begin{aligned}\frac{\partial \mathcal{L}_P}{\partial \xi_i} &= \frac{\partial}{\partial \xi_i}\left\{\frac{1}{2}||\beta||_2^2 + C\cdot\sum_i \xi_i - \sum_i \alpha_i\left[y_i(x_i^t\beta+\beta_0) - (1-\xi_i)\right] - \sum_i \mu_i\xi_i\right\} \\ &= \frac{\partial}{\partial \xi_i}\left\{C\cdot\sum_i \xi_i + \sum_i \alpha_i(1-\xi_i) - \sum_i \mu_i\right\} \\ &= C - \alpha_i - \mu_i\end{aligned}$$

Setting this equal to zero we see that:

$$\alpha_i = C - \mu_i.$$

We now want to plug this into the Lagrangian primal function. We first use the fact that $\alpha_i = C - \mu_i$ to unite the trailing terms with respect to α_i:

$$\begin{aligned}
\mathcal{L}_D &= \frac{1}{2}||\beta||_2^2 + C\sum_i \xi_i - \sum_i \alpha_i \left[y_i(x_i^t\beta + \beta_0) - (1-\xi_i)\right] - \sum_i \mu_i\xi_i \\
&= \frac{1}{2}||\beta||_2^2 + \sum_i (C-\mu_i)\xi_i - \sum_i \alpha_i y_i x_i^t\beta - \sum_i \alpha_i y_i\beta_0 + \sum_i \alpha_i(1-\xi_i) \\
&= \frac{1}{2}||\beta||_2^2 + \sum_i \alpha_i\xi_i - \sum_i \alpha_i y_i x_i^t\beta - \sum_i \alpha_i y_i\beta_0 + \sum_i \alpha_i(1-\xi_i) \\
&= \frac{1}{2}||\beta||_2^2 + \sum_i \alpha_i - \sum_i \alpha_i y_i x_i^t\beta - \sum_i \alpha_i y_i\beta_0
\end{aligned}$$

We now want to plug this into the Lagrangian primal function. We first use the fact that $\alpha_i = C - \mu_i$ to unite the trailing terms with respect to α_i:

$$\begin{aligned}
\mathcal{L}_D &= \frac{1}{2}||\beta||_2^2 + C\sum_i \xi_i - \sum_i \alpha_i \left[y_i(x_i^t\beta + \beta_0) - (1-\xi_i)\right] - \sum_i \mu_i\xi_i \\
&= \frac{1}{2}||\beta||_2^2 + \sum_i (C-\mu_i)\xi_i - \sum_i \alpha_i y_i x_i^t\beta - \sum_i \alpha_i y_i\beta_0 + \sum_i \alpha_i(1-\xi_i) \\
&= \frac{1}{2}||\beta||_2^2 + \sum_i \alpha_i\xi_i - \sum_i \alpha_i y_i x_i^t\beta - \sum_i \alpha_i y_i\beta_0 + \sum_i \alpha_i(1-\xi_i) \\
&= \frac{1}{2}||\beta||_2^2 + \sum_i \alpha_i - \sum_i \alpha_i y_i x_i^t\beta - \sum_i \alpha_i y_i\beta_0
\end{aligned}$$

And the last term drops out:

$$\begin{aligned}
\mathcal{L}_D &= \frac{1}{2}||\beta||_2^2 + \sum_i \alpha_i - \sum_i \alpha_i y_i x_i^t\beta - \beta_0 \sum_i \alpha_i y_i \\
&= \frac{1}{2}||\beta||_2^2 + \sum_i \alpha_i - \sum_i \alpha_i y_i x_i^t\beta
\end{aligned}$$

Now, notice that since $\beta = \sum_i \alpha_i y_i x_i$, we have that:

$$\begin{aligned} ||\beta||_2^2 &= \sum_j \beta_j^2 \\ &= \sum_{i'} \sum_i (\alpha_i y_i x_i)^t (\alpha_{i'} y_{i'} x_{i'}) \\ &= \sum_{i'} \sum_i \alpha_i \alpha_{i'} y_i y_{i'} x_i^t x_{i'} \end{aligned}$$

Also see that we can rewrite the last term in our dual function as:

$$\begin{aligned} \sum_i \alpha_i y_i x_i^t \beta &= \sum_{i'} \sum_i \alpha_i \alpha_{i'} y_i y_{i'} x_i^t x_{i'} \\ &= ||\beta||_2^2. \end{aligned}$$

Finally, the dual function can be written as:

$$\begin{aligned}\mathcal{L}_D &= \frac{1}{2}||\beta||_2^2 + \sum_i \alpha_i - \sum_i \alpha_i y_i x_i^t \beta \\ &= \sum_i \alpha_i - \frac{1}{2}||\beta||_2^2 \\ &= \sum_i \alpha_i - \frac{1}{2} \cdot \sum_{i'} \sum_i \alpha_i \alpha_{i'} y_i y_{i'} x_i^t x_{i'}.\end{aligned}$$

Which we want to maximize under the constraints (the first is from the KKT conditions, the second from the partial derivative of the bias):

$$0 \leq \alpha_i \leq C, \quad \sum_i \alpha_i y_i = 0.$$

To what extent can we make some sense of this equation? First notice the duality of the problem if we flip the ± 1 labeling of the classes y_i:

$$\mathcal{L}_D = \sum_i \alpha_i - \frac{1}{2} \cdot \sum_{i'} \sum_i \alpha_i \alpha_{i'} \, y_i y_{i'} \, x_i^t x_{i'}$$

The function only depends on the sign of $y_i y_{i'}$. Also note that it only depends on the data that serve as support vectors:

$$\mathcal{L}_D = \sum_i \alpha_i - \frac{1}{2} \cdot \sum_{i'} \sum_i \alpha_i \alpha_{i'} \, y_i y_{i'} \, x_i^t x_{i'} \, .$$

Perhaps most importantly though, notice that only the outer product XX^t effects the final results:

$$\mathcal{L}_D = \sum_i \alpha_i - \frac{1}{2} \cdot \sum_{i'} \sum_i \alpha_i \alpha_{i'} \, y_i y_{i'} \, x_i^t x_{i'} \, .$$

As $x_i^t x_{i'}$ is the (i, i')'th element of XX^t. This is a measurement of how similar x_i and $x_{i'}$ are to one another (if scaled to both have length one, it is the cosine of the angle between them).

Perhaps most importantly though, notice that only the outer product XX^t effects the final results:

$$\mathcal{L}_D = \sum_i \alpha_i - \frac{1}{2} \cdot \sum_{i'} \sum_i \alpha_i \alpha_{i'} y_i y_{i'} x_i^t x_{i'} .$$

As $x_i^t x_{i'}$ is the (i, i')'th element of XX^t. This is a measurement of how similar x_i and $x_{i'}$ are to one another (if scaled to both have length one, it is the cosine of the angle between them).

So, we can see that:

1. There is a penalty for including two similar x_i's with the same class label
2. There is a benefit for including two similar x_i's with different class labels

Both of which actually make sense for a classification algorithm.

Taking a step back now, how does logistic regression and support vector machines compare?

1. Both separate the plane into two half-spaces which attempt to split the classes as well as possible
2. However, logistic regression is (primarily) concerned with the correlation matrix X^tX between the variables and support vector machines only care about the similarity matrix XX^t between observations

The Kernel Trick

In the case of logistic and linear regression I have shown how basis expansion can be used to add non-linear effects into a linear model. One observation that makes support vector machines attractive is that it is possible to mimic basis expansion without ever having to actually project into a higher dimensional space.

The Kernel Trick, cont.

Assume that we have a mapping h of samples x into a higher dimensional space. We can re-write the dual function using inner product notation:

$$\mathcal{L}_D = \sum_i \alpha_i - \frac{1}{2} \cdot \sum_{i'} \sum_i \alpha_i \alpha_{i'} y_i y_{i'} < h(x_i), h(x_{i'}) >.$$

It quickly becomes apparent that we only need a fast way of calculating inner products in the space of h, which may not require actually determining and calculating h itself.

The Kernel Trick, cont.

The projected inner product $< h(x_i), h(x_{i'}) >$ is usually written directly as $K(x_i, x_{i'})$ for a function K called the kernel. Popular choices include:

1. **Linear:** $K(x, x') =< x, x' >$
2. **Polynomial:** $K(x, x') = (1+ < x, x' >)^d$
3. **Radial:** $K(x, x') = exp(-\gamma ||x - x'||^2)$
4. **Sigmoid:** $K(x, x') = tanh(\kappa_1 < x, x' > +\kappa_2)$

Notice that these all require approximately the same effort to calculate as the linear kernel.

Finishing the optimization

Now, we can re-write the optimization problem as:

$$\begin{aligned} \max \quad & 1^t\alpha - \alpha^t K\alpha \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C \end{aligned}$$

For a suitable matrix K, called the kernel matrix. This is a quadratic program with box constraints, and can be solved fairly efficiently by general purpose solvers.

More information

I try to provide additional references for all of my lectures on the class website. For today's material (and Wednesday's) I would like to make a particular point to mention two references:

- Elements of Statistical Learning, Sections 12.1-12.4
- Convex Optimization, S. Boyd, Chapter 5 (5.5 in particular)

These contain many more details than I have time to cover, and assume a deeper background in statistics / convex calculus.