COS 511: Foundations of Machine Learning

Rob Schapire
Scribe: Wil Brady

Lecture # 12
March 16, 2006

1 Support Vector Machines Continued

As discussed in the previous class, given a set of examples labeled positive and negative that are separable, we wish to find the hyperplane with largest margin.

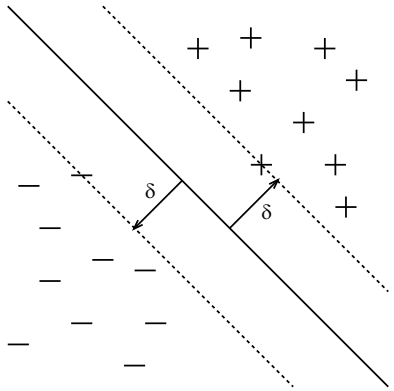

Figure 1

Margin of example $(\mathbf{x}_i, y_i)$ is the distance between the example and the hyperplane. We let δ be the smallest margin. Thus we have the problem:

$$\begin{aligned} &\max \delta \\ &\text{such that } ||\mathbf{v}||_2 = 1 \qquad y_i(\mathbf{v} \cdot \mathbf{x}_i) \geq \delta \qquad \forall i \end{aligned} \tag{1}$$

Now if we divide $y_i(\mathbf{v} \cdot \mathbf{x}_i) \geq \delta$ through by δ and let $\mathbf{w} = \frac{\mathbf{v}}{\delta}$, we then get $y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 1$ and $||\mathbf{w}|| = 1/\delta$. Thus our maximization problem above can be rewritten as a minimization problem:

$$\begin{aligned} &\min \frac{1}{2}||\mathbf{w}||^2 \\ &\text{such that } y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 1 \qquad \forall i \end{aligned} \tag{2}$$

Note the $\frac{1}{2}$ and the square do not affect where $||\mathbf{w}||$ is minimized. Also note that $y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 1$ can be rewritten as $y_i(\mathbf{w} \cdot \mathbf{x}_i) - 1 \geq 0$.

Now we define the Lagrangian

$$L(\mathbf{w}, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{w}||^2 - \sum_i \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x}_i) - 1]$$

where $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_m)$ are called the Lagrange multipliers. Also, let $b_i(\mathbf{w}) = y_i(\mathbf{w} \cdot \mathbf{x}_i) - 1$.

Claim: the problem

$$\min_{\mathbf{w}} \max_{\boldsymbol{\alpha} \geq 0} L(\mathbf{w}, \boldsymbol{\alpha}) \tag{3}$$

is the same as problem(2).

proof: View this as a game where

$$\begin{aligned}&\text{Player 1 first chooses } \mathbf{w} \text{ to } \min L(\mathbf{w}, \boldsymbol{\alpha})\\ &\text{Player 2 then chooses } \boldsymbol{\alpha} \geq 0 \text{ to } \max L(\mathbf{w}, \boldsymbol{\alpha})\end{aligned}$$

If player 1 chooses $\mathbf{w}$ such that $b_i(\mathbf{w}) < 0$ for some i, then player 2 chooses $\alpha_i = \infty$ which will yield $L(\mathbf{w}, \boldsymbol{\alpha}) = \infty$. Thus we see player 1 will choose $b_i(\mathbf{w}) \geq 0$ for all i. So we see that player 1 will choose $\mathbf{w}$ that will satisfy the constraint of problem(2) - $y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 1$.

Continuing with the game, if player 1 chooses $\mathbf{w}$ so that $b_i(\mathbf{w}) = 0$ then α_i is irrelevant. If $\mathbf{w}$ is chosen so that $b_i(\mathbf{w}) > 0$ then $\alpha_i = 0$. So in either case we have $\alpha_i b_i(\mathbf{w}) = 0$. So now we have reduced $L(\mathbf{w}, \boldsymbol{\alpha}) = \frac{1}{2}||\mathbf{w}||^2$ but with $\mathbf{w}$ satisfying $y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 1$, which is exactly problem(2). □

Now what if we reverse player roles:

$$\begin{aligned}&\text{Player 1 first chooses } \boldsymbol{\alpha} \geq 0 \text{ to } \max L(\mathbf{w}, \boldsymbol{\alpha})\\ &\text{Player 2 then chooses } \mathbf{w} \text{ to } \min L(\mathbf{w}, \boldsymbol{\alpha}).\end{aligned}$$

Then we have the problem

$$\max_{\boldsymbol{\alpha} \geq 0} \min_{\mathbf{w}} L(\mathbf{w}, \boldsymbol{\alpha}).$$

Logically it would seem being the second player is more advantageous, and this turns out to be true. That is we get

$$\max_{\boldsymbol{\alpha} \geq 0} \min_{\mathbf{w}} L(\mathbf{w}, \boldsymbol{\alpha}) \leq \min_{\mathbf{w}} \max_{\boldsymbol{\alpha} \geq 0} L(\mathbf{w}, \boldsymbol{\alpha}).$$

In the above it turns out we will have equality if $L(\mathbf{w}, \boldsymbol{\alpha})$ is convex in $\mathbf{w}$ and concave in $\boldsymbol{\alpha}$. In our case, we have equality

$$\max_{\boldsymbol{\alpha} \geq 0} \min_{\mathbf{w}} L(\mathbf{w}, \boldsymbol{\alpha}) = \min_{\mathbf{w}} \max_{\boldsymbol{\alpha} \geq 0} L(\mathbf{w}, \boldsymbol{\alpha}).$$

We call

$$\max_{\boldsymbol{\alpha} \geq 0} \min_{\mathbf{w}} L(\mathbf{w}, \boldsymbol{\alpha})$$

the convex dual.

(As a quick refresher, a function $f(x)$ is convex if it satisfies

$$f(\lambda x + (1 - \lambda)y) \leq \lambda f(x) + (1 - \lambda) f(y)$$

for all x, y. And a function is concave if

$$f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda) f(y)$$

for all x, y. See Figure 2.)

Let

$$\mathbf{w}^* = \arg \min_{\mathbf{w}} \max_{\boldsymbol{\alpha} \geq 0} L(\mathbf{w}, \boldsymbol{\alpha})$$

$$\boldsymbol{\alpha}^* = \arg \max_{\boldsymbol{\alpha} \geq 0} \min_{\mathbf{w}} L(\mathbf{w}, \boldsymbol{\alpha})$$

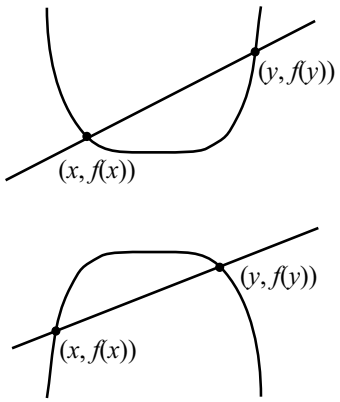

Figure 2: The top picture is convex, and the bottom picture concave

We see

$$\begin{array}{ccccccc} L(\mathbf{w}^*, \boldsymbol{\alpha}^*) & \leq & \max_{\boldsymbol{\alpha} \geq 0} L(\mathbf{w}^*, \boldsymbol{\alpha}) & & \min_{\mathbf{w}} L(\mathbf{w}, \boldsymbol{\alpha}^*) & \leq & L(\mathbf{w}^*, \boldsymbol{\alpha}^*) \\ & & \| & & \| & & \\ & & \min_{\mathbf{w}} \max_{\boldsymbol{\alpha} \geq 0} L(\mathbf{w}, \alpha) & = & \max_{\boldsymbol{\alpha} \geq 0} \min_{\mathbf{w}} L(\mathbf{w}, \boldsymbol{\alpha}) & & \end{array}$$

So $\boldsymbol{\alpha}^*$ maximizes $L(\mathbf{w}^*, \cdot)$ and $\mathbf{w}^*$ minimizes $L(\cdot, \boldsymbol{\alpha}^*)$. Thus we see the solution lies at a saddle point. From these facts we see at the solution we have the following conditions:

$$\frac{\partial L(\mathbf{w}, \boldsymbol{\alpha})}{\partial w_j} = 0$$

$$\forall i \quad b_i(\mathbf{w}) \geq 0 \quad \alpha_i \geq 0 \quad \alpha_i b_i(\mathbf{w}) = 0$$

The first condition is because $\mathbf{w}^*$ is minimum with no restrictions for all i. The second line of conditions are called the "complementary slackness conditions". And all the conditions are called the "KKT conditions" for Karush, Kuhn, and Tucker.

Now to solve $\max_{\boldsymbol{\alpha} \geq 0} \min_{\mathbf{w}} L(\mathbf{w}, \boldsymbol{\alpha})$ we first solve $\min_{\mathbf{w}} L(\mathbf{w}, \alpha)$. Let x_{ij} be the jth component of the ith example, and we get

$$\frac{\partial L(\mathbf{w}, \boldsymbol{\alpha})}{\partial w_j} = w_j - \sum_i \alpha_i y_i x_{ij}.$$

Solving for $\mathbf{w}$ gives

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

and this minimizes $L(\mathbf{w}, \boldsymbol{\alpha})$. Now we plug in $\mathbf{w}$ into $L(\mathbf{w}, \alpha)$ to get

$$\max_{\boldsymbol{\alpha} \geq 0} \min_{\mathbf{w}} L(\mathbf{w}, \boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha} \geq 0} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j. \tag{4}$$

We can optimize this last equation using a number of techniques such as a hill climbing algorithm.

Now we note that we have $\alpha_i b_i(\mathbf{w}) = 0$ and if $\alpha_i \neq 0$ then $b_i(\mathbf{w}) = 0$, written out gives $y_i(\mathbf{w} \cdot \mathbf{x}_i) = 1$. So we see ith example is a support vector. Thus our solution only depends

on the support vectors. So if we have k support vectors, then by homework 2 problem 1 we see

$$\text{err} \leq O\left(\frac{k\ln(m) + \ln(1/\delta)}{m}\right).$$

Now if our examples are not separable, then we can introduce ξ_i as the distance we have to move an example which has been wrongly classified by our hyperplane — see figure below.

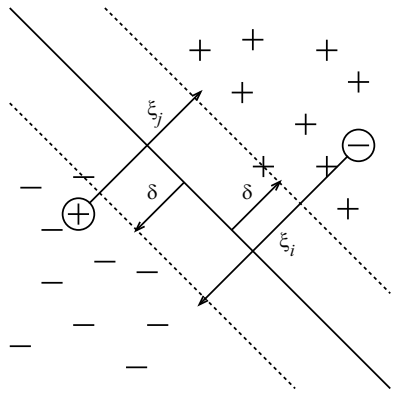

Figure 3

This gives the problem

$$\begin{array}{l} \min \frac{1}{2}||\mathbf{w}||^2 - C\sum_i \xi_i \\ \text{such that } y_i(\mathbf{w}\cdot\mathbf{x}_i) \geq 1 - \xi_i \qquad \xi_i \geq 0 \qquad \forall i \end{array}$$

where C is some fixed constant.

Now suppose we have examples where finding even an approximately separating hyperplane is not possible — see figure 4.

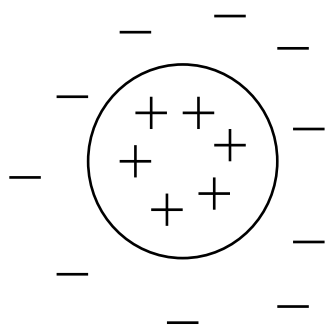

Figure 4

Then we map our examples into a higher dimensional space. Suppose $\mathbf{x} = (x_1, x_2)$. We do the following:

$$\mathbf{x} \mapsto (1, x_1, x_2, x_1^2, x_2^2, x_1x_2) = \boldsymbol{\psi}(\mathbf{x}).$$

This gives us a linear classifier (hyperplane) now of the form

$$a + bx_1 + cx_2 + dx_1^2 + ex_2^2 + fx_1x_2 = 0.$$

We see this example can be extended from the 2-dimensional space where the above $\mathbf{x}$ lives to any n-dimensional space and extended from the collection of degree 2 or less monomials $(x_1, x_2, x_1^2, x_2^2, x_1x_2)$ to all monomials of degree d or less. So we see that mapping to a higher dimension allows a much more general classifier (conic sections in the example above). However, we are adding $O(n^d)$ dimensions. This leads to computational and statistical problems. But support vector machines handles these problems nicely.

For the statistical problem, we recall the VC-dimension is $1/\delta^2$ so is independent of the dimension of our example space. Thus going to larger dimensional example space does not necessarily hurt.

As for the computational problem, notice in equation (4) that we only need to compute the inner product of our example vectors to solve $\max_{\boldsymbol{\alpha}\geq 0} \min_{\mathbf{w}} L(\mathbf{w}, \alpha)$. So if we map our examples into a higher dimension space using $\boldsymbol{\psi}(\mathbf{x})$, our problem then becomes

$$\max_{\boldsymbol{\alpha}\geq 0} \min_{\mathbf{w}} L(\mathbf{w}, \boldsymbol{\alpha}) = \max_{\boldsymbol{\alpha}\geq 0} \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \boldsymbol{\psi}(\mathbf{x}_i) \cdot \boldsymbol{\psi}(\mathbf{x}_j). \tag{5}$$

So we only need to compute the inner product $\boldsymbol{\psi}(\mathbf{x}_i) \cdot \boldsymbol{\psi}(\mathbf{x}_j)$ to solve our problem. For the example of $\boldsymbol{\psi}(\mathbf{x})$ given previously, we add some well chosen constants to $\boldsymbol{\psi}(\mathbf{x})$ which will be absorbed in the coefficients of the linear classifier equation. Let

$$\boldsymbol{\psi}(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2).$$

So we see

$$\boldsymbol{\psi}(\mathbf{x}) \cdot \boldsymbol{\psi}(\mathbf{u}) = 1 + 2x_1u_1 + 2x_2u_2 + x_1^2u_1^2 + x_2^2u_2^2 + 2x_1x_2u_1u_2 = (1 + x_1u_1 + x_2u_2)^2 = (1 + (\mathbf{x} \cdot \mathbf{u}))^2.$$

The above is an example of a "kernel" mapping: $K(\mathbf{x}, \mathbf{u}) = \boldsymbol{\psi}(\mathbf{x}) \cdot \boldsymbol{\psi}(\mathbf{u})$. A "kernel" map is an inner product mapped into higher dimensions. Thus a kernel mapping helps us avoid computing the inner product in the higher dimensional space by using the lower dimensional inner product. Thus kernel mappings solve the problem of computing the higher dimensional inner product found in equation (5).

A further example of a kernel map would be the extension of our example above for any n-dimensional space and monomials of degree d, $(1 + (\mathbf{x} \cdot \mathbf{u}))^d$ called polynomial kernel. Also there is $\exp(-c||\mathbf{x} - \mathbf{u}||_2^2)$ which is called the radial basis kernel. For mapping into higher dimensions we can use any kernel map K satisfying the Mercer Conditions — symmetric and positive semidefinite.

One overlooked fact: We assumed $||\mathbf{x}_i|| \leq 1$ to get a VC-dimension of $1/\delta^2$. If $||\mathbf{x}_i|| \leq R$ we get a VC-dimension of R^2/δ^2. And we see if $||\mathbf{x}_i|| \leq 1$ then

$$||\boldsymbol{\psi}(\mathbf{x})||^2 = \boldsymbol{\psi}(\mathbf{x}) \cdot \boldsymbol{\psi}(\mathbf{x}) = (1 + \mathbf{x} \cdot \mathbf{x})^d \leq 2^d.$$

Thus we see $||\boldsymbol{\psi}(\mathbf{x})|| \leq 2^{d/2}$ and not $||\boldsymbol{\psi}(\mathbf{x})|| \leq 1$, which can cause our VC-dimension to be bigger than $1/\delta^2$. So, mapping to a higher dimensional space is likely to cause δ to increase, but it also may cause R to increase, which means it may or may not improve performance depending on how much δ increases relative to R.