COS 511: Theoretical Machine Learning

Lecturer: Rob Schapire
Scribe: Jian Min Sim

Lecture #12
March 14, 2013

1 Bounding the Margin

We are continuing the proof of a bound on the generalization error of AdaBoost from the last lecture. We have shown previously, with probability at least $1-\delta$ over the choice of S, for all $f \in \mathcal{F}$,

$$E_D[f] \leq \hat{E}_S[f] + 2\hat{\mathcal{R}}_S(\mathcal{F}) + \mathcal{O}\left(\sqrt{\frac{\ln(1/\delta)}{m}}\right) \tag{1.1}$$

where $\hat{E}_S[f]$ represents the sample average. Additionally, we also want to show that if we can get a large margin from AdaBoost, this will translate to a low generalization error. We want to use (1.1) to prove a bound on the generalization error in terms of the margins.

First of all, $\mathcal{H}$ will denote a weak hypothesis space with VC-dimension of d, and recall the convex hull of $\mathcal{H}$ which we defined to be:

$$co(\mathcal{H}) = \left\{ f : f(x) = \sum_{t=1}^{T} a_t h_t(x) \mid T \geq 1, a_t \geq 0, \sum_{t=1}^{T} a_t = 1, h_1, \ldots, h_T \in \mathcal{H} \right\} \tag{1.2}$$

Theorem 1. *With training set of size m, for all $f \in co(\mathcal{H})$, and for all margin, $\Theta > 0$,*

$$Pr_D[yf(x) \leq 0] \leq \hat{Pr}_S[yf(x) \leq \Theta] + \tilde{\mathcal{O}}\left(\sqrt{\frac{d/\Theta^2 + \ln 1/\delta}{m}}\right)$$

with probability at least $1-\delta$.

Proof. As shown in the last lecture, we state some of the relationships known:

$$\hat{\mathcal{R}_S}(\mathcal{H}) \leq \tilde{\mathcal{O}}\left(\sqrt{d/m}\right) \tag{1.3}$$

$$\mathcal{M} = \{(x,y) \mapsto yf(x) : f \in co(\mathcal{H})\} \tag{1.4}$$

$$\hat{\mathcal{R}}_S(\mathcal{M}) = \hat{\mathcal{R}}_S(co(\mathcal{H})) = \hat{R}_S(\mathcal{H}) \text{ (Rademacher complexity)} \tag{1.5}$$

We also need to note, from last lecture, that:

$$Pr_D[yf(x) \leq 0] = E_D[1\{yf(x) \leq 0\}] \tag{1.6}$$

$$\hat{Pr}_S[yf(x) \leq \Theta] = \hat{E}_S[1\{yf(x) \leq \Theta\}] \tag{1.7}$$

From Figure 1.1, we can arrive at:

$$1\{u \leq 0\} \leq \phi(u) \leq 1\{u \leq \Theta\} \tag{1.8}$$

where $\phi(u)$ is the Lipschitz function shown in the figure

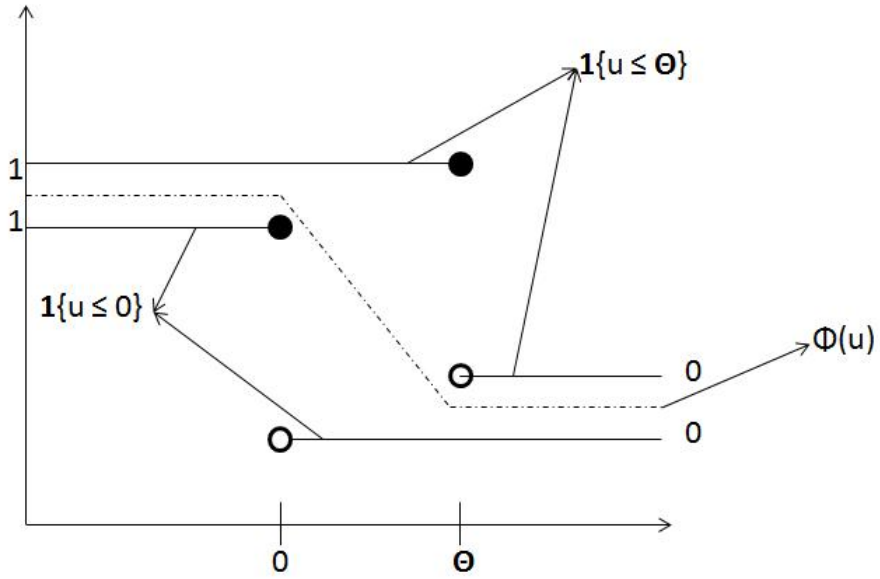

Figure 1.1: Visualization of the indicator functions and the Lipschitz function, $\phi(u)$, where Lipschitz constant, $L_\phi = \frac{1}{\Theta}$ (i.e. slope of line).

From (1.6),

$$Pr_D[yf(x) \leq 0] = E_D[1\{yf(x) \leq 0\}] \leq E_D[\phi(yf(x)] \tag{1.9}$$

From (1.7),

$$Pr_D[yf(x) \leq \Theta] = \hat{E}_S[1\{yf(x) \leq \Theta\}] \geq \hat{E}_S[\phi(yf(x))] \tag{1.10}$$

By applying (1.1), for all $f \in co(\mathcal{H})$,

$$E_D[\phi(yf(x))] \leq \hat{E}_S[\phi(yf(x))] + 2\hat{\mathcal{R}}_S(\phi \circ \mathcal{M}) + \mathcal{O}\left(\sqrt{\frac{\ln 1/\delta}{m}}\right) \tag{1.11}$$

Now, the only thing is to compute $2\hat{\mathcal{R}}_S(\phi \circ \mathcal{M})$. Recall from previous lecture, the Talagrand's Lemma, which states that for any hypothesis set H of real-valued functions, the following inequality holds if $\phi : \mathbb{R} \to \mathbb{R}$ is a Lipschitz function[1]:

$$\hat{\mathcal{R}}_S(\phi \circ \mathcal{F}) \leq L_\phi \hat{\mathcal{R}}_S(\mathcal{F}) \tag{1.12}$$

From (1.12),

[1] ϕ is Lipschitz continuous means that $\forall u, v, |\phi(u) - \phi(v)| \leq L_\phi |u - v|$, where L_ϕ is some Lipschitz constant

$$\begin{aligned}\hat{\mathcal{R}}_S(\phi \circ \mathcal{M}) &\leq L_\phi \hat{\mathcal{R}}_S(\mathcal{M}) \\ &= 1/\Theta \cdot \mathcal{R}(\mathcal{H}) \\ &\leq 1/\Theta \cdot \tilde{\mathcal{O}}\left(\sqrt{d/m}\right) \end{aligned} \tag{1.13}$$

From (1.9),

$$\begin{aligned}Pr_D[yf(x) \leq 0] &= E_D[1\{yf(x) \leq 0\}] \\ &\leq E_D[\phi(yf(x))] \\ &\leq \hat{E}_S[\phi(yf(x))] + \tilde{\mathcal{O}}\left(\sqrt{\frac{d/\Theta^2 + \ln 1/\delta}{m}}\right) \text{ (using 1.13)} \\ &\leq \hat{Pr}_S[yf(x) \leq \Theta] + \tilde{\mathcal{O}}\left(\sqrt{\frac{d/\Theta^2 + \ln 1/\delta}{m}}\right) \text{ (using 1.10)}\end{aligned}$$

□

2 Support Vector Machines (SVM)

We start off with m labeled examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{-1, +1\}$, and our goal is to find a hypothesis that is consistent with all the m examples. We are assuming for now that the data is linearly separable and $\| \mathbf{x}_i \|_2 \leq 1$.

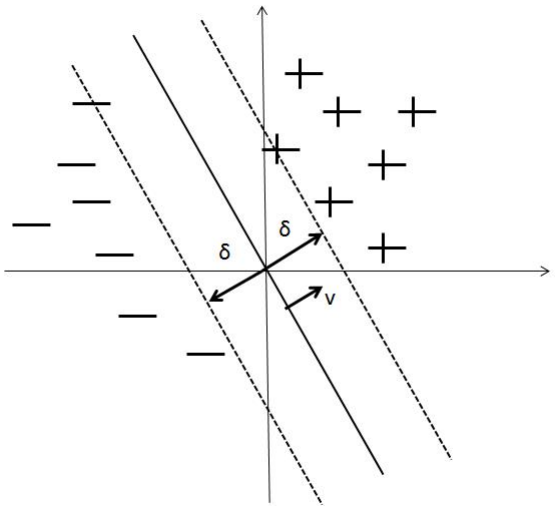

Figure 2.1: Labeled examples in a plane in which a separating hyperplane (i.e. line) is shown to be passing through the origin.

Since the hypothesis must be one that separates between the postive and negative labels as shown in Figure 2.1, there are in fact a lot of choices to choose from in this case. Intuitively, the natural idea is to choose the hyperplane that maximally separates between the positive and negative labels (i.e. to maximize δ). This is because we do not want the jiggering of a point to change the classification.

Assuming that the hyperplane passes through the origin and from Figure 2.1, $\mathbf{v}$ is defined to be the normal vector to the hyperplane such that $\| \mathbf{v} \|_2 = 1$ (i.e. L2-norm). Also the distance of a point $\mathbf{x}$ to the hyperplane is $\mathbf{v} \cdot \mathbf{x}$:

$$\mathbf{v} \cdot \mathbf{x} \begin{cases} > 0 & \text{if } \mathbf{x} \text{ above hyperplane} \\ = 0 & \text{if on hyperplane} \\ < 0 & \text{if } \mathbf{x} \text{ below hyperplane} \end{cases}$$

Hence, the problem formulation becomes:

$$\begin{aligned} &\text{find } \mathbf{v}, \delta \\ &\max \delta \\ &\text{s.t. } \| \mathbf{v} \|_2 = 1 \\ &\forall i \begin{cases} \mathbf{v} \cdot \mathbf{x}_i \geq \delta & \text{if } y_i = +1 \\ \mathbf{v} \cdot \mathbf{x}_i \leq \delta & \text{if } y_i = -1 \end{cases} \end{aligned} \tag{2.1}$$

in which (2.1) is mathematically equivalent to

$$y_i(\mathbf{v} \cdot \mathbf{x}_i) \geq \delta \tag{2.2}$$

Another way to view this problem is to think in terms of increasing our confidence of the margin. Since we do not have control of the number of samples but only the choices of the hyperplanes, we could control the complexities of the hyperplanes.

Recall that the VC-dimension of the linear threshold functions in $\mathbb{R}^n$ equals to n, and not $n+1$ because they go through the origin. Furthermore, the VC-dimension of the linear threshold functions with margin δ is smaller than $1/\delta^2$, which means that it is independent of n (to be proved as a corollary later).

Since the VC-dimension of linear threshold functions with margin δ is only dependent on δ^2, this means that the bigger the margin δ, the less complex the hyperplane gets. Hence, going back to the problem formulation in (2.1), we divide the terms in (2.2) by δ:

$$y_i(\frac{\mathbf{v}}{\delta} \cdot \mathbf{x}_i) \geq \frac{\delta}{\delta} = 1$$

If we let $\mathbf{w} = \frac{\mathbf{v}}{\delta}$,

$$\| \mathbf{w} \| = \frac{\| \mathbf{v} \|}{\delta} = \frac{1}{\delta}$$

We can then rewrite the problem formulation into a convex program:

$$\min \frac{1}{2} \| \mathbf{w} \|_2^2 \tag{2.3}$$

$$\text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i) \geq 1 \tag{2.4}$$

Since for all convex programs, there is an equivalent formulation, we can transform it to its dual maximization form using Lagrange multipliers α_i (to be shown in later lectures!):

$$\max \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$
$$\text{s.t. } \forall i : \alpha_i \geq 0$$

It turns out that:

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

and a lot of α_i will be zeros. In fact, α_i will be zero only if the corresponding example $(\mathbf{x}_i, y_i)$ is a support vector meaning that $y_i(\mathbf{w} \cdot \mathbf{v}) = 1$. Since $\mathbf{w}$ does not depend on the points that are not support vectors but only on a small subset of examples, we can also see, from problem 1 of homework 4, that $\text{err}(H) \leq \tilde{\mathcal{O}}\left(\frac{k+\ln 1/\delta}{m}\right)$, where k is the number of support vectors, δ is the margin, and error only depends on the k, and not the number of dimensions.

2.1 Linearly Inseparable Data

So far, we have been talking about linearly separable data such as the one shown in Figure 2.1, where there exists some hyperplane that separates the positive and negative examples. What happens when the data is linearly inseparable? Consider the following case:

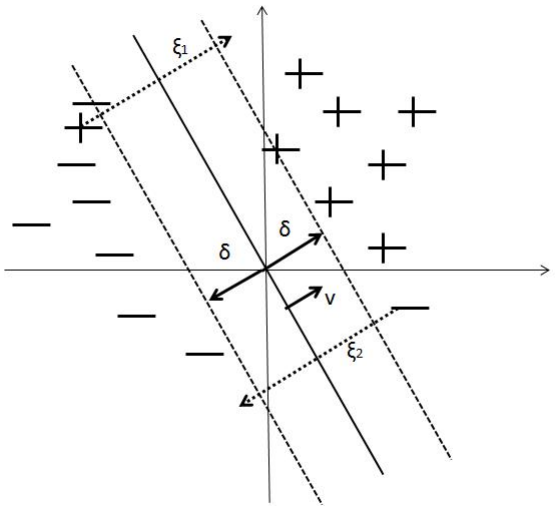

Figure 2.2: Linearly inseperable data on a plane with hypothesis as a line passing through the origin, where ξ_1 and ξ_2 represents the distances moved.

In this simple case, to make the data linearly separable, we can move the data, through the distance ξ_1 and ξ_2 to the correct side of the hyperplane. However, there is a penalty to be paid for the distances moved and we will need to incorporate it to the problem formulation, (2.4):

$$\min \frac{1}{2} \parallel \mathbf{w} \parallel_2^2 + C \sum_{i=1}^{m} \xi_i \text{ ,where } C\text{is a parameter user has to choose}$$
$$\text{s.t. } y_i(\mathbf{w} \cdot \mathbf{v}) \geq 1 - \xi_i$$
$$\xi_i \geq 0$$

The dual maximization form then becomes:

$$\max \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$$
$$\text{s.t. } \forall i : 0 \leq \alpha_i \leq C$$

However, what if it is not even close to being linearly separable? Consider the following case:

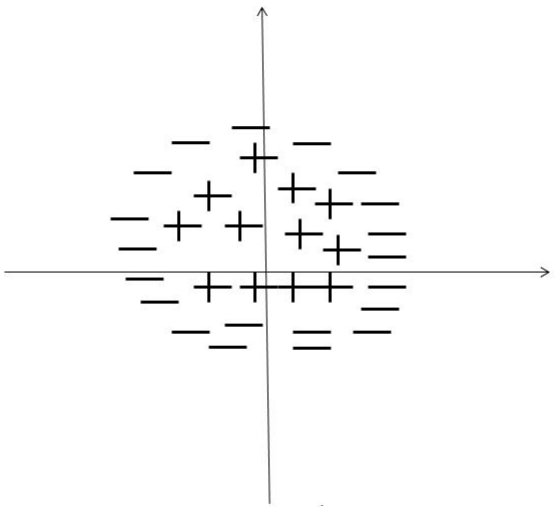

Figure 2.3: Linearly inseperable data on a plane.

Support Vector Machine can still be used in this case. To do this, we map the data to higher dimensional space (i.e. from 2 dimensional space to 6 dimensional space):

$$(x_1,x_2) = \mathbf{x} \mapsto \psi(\mathbf{x}) = (1, x_1, x_2, x_1^2, x_2^2, x_1 x_2) \tag{2.5}$$

However, how is this linearly separable now? Since the hyperplane equation is given by $\mathbf{w} \cdot \mathbf{x} = 0$, the hyperplane equation for the 6 dimensional space becomes:

$$a \cdot 1 + bx_1 + cx_2 + dx_1^2 + ex_2^2 + fx_1x_2 = 0 \tag{2.6}$$

which turns out to be the equation of the conic section in 2 dimensional space.
Graphically, the hyperplane in this example is:

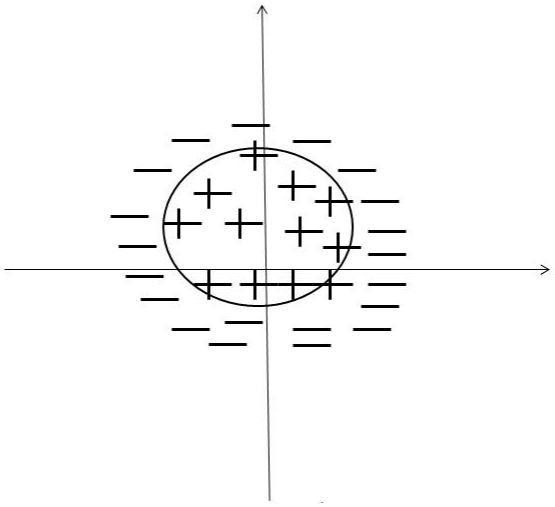

Figure 2.4: Linearly inseperable data on a plane with hypothesis as a circle in this case.

The above approach could be generalized to n dimensions:

$$\mathbf{x} \in \mathbb{R}^n \mapsto \psi(\mathbf{x})$$

However, the problem with this is that we have both computational and statistical issues.

- Computational Problem: We start with examples in n dimensions and add all terms up to degree d, then we are effectively mapping to $\mathcal{O}(n^d)$ dimensions. So, we might start in 100 dimensions, and end up with a trillion dimensions or more. So, even reading such a vector just once will take a very long time.
- Statistical Problem: There may be overfitting due to the addition of so many parameters.

For the statistical problem, Support Vector Machine is able to overcome it since VC-dimension $= 1/\delta^2$, and hence $\text{err}(H)$ does not depend on the number of dimensions.

From the dual maximization formulation, we can see that there is only a need to compute the inner product of pairs of instances because $\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$ and $H(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x}) = \text{sign}\left(\sum_{i=1}^{m} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{x}_i)\right)$. Hence, if we are able to compute the inner product efficiently, then we can overcome the computational problem as well.

We first consider adding constants to (2.5), which has no effect on the hyperplanes since the constants are arbitrary.

$$(x_1,x_2) = \mathbf{x} \mapsto \psi(\mathbf{x}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2) \tag{2.7}$$

Similarly,

$$(u_1,u_2) = \mathbf{u} \mapsto \psi(\mathbf{u}) = (1, \sqrt{2}u_1, \sqrt{2}u_2, u_1^2, u_2^2, \sqrt{2}u_1u_2) \tag{2.8}$$

For raising 2 dimensional space to 6 dimensional space, we just have to perform the inner product for $\psi(\mathbf{x})$ and $\psi(\mathbf{u})$:

$$\begin{aligned}\psi(\mathbf{x}) \cdot \psi(\mathbf{u}) &= 1 + 2x_1u_1 + 2x_2u_2 + x_1^2u_1^2 + x_2^2u_2^2 + 2x_1x_2u_1u_2 \\ &= (1 + x_1u_1 + x_2u_2)^2 \\ &= (1 + \mathbf{x} \cdot \mathbf{u})^2\end{aligned}$$

Hence, adding all terms up to degree d, the computation would be modified to become: $(1 + \mathbf{x} \cdot \mathbf{u})^d$. This elegant result means that we can implicitly compute inner product in the high dimensional space by performing all operations in the original low dimensional space and then simply raising to the power of d. This method is called the kernel trick. In general, a kernel is a function $K(\mathbf{x}, \mathbf{u})$ that implicitly computes the inner product $\psi(\mathbf{x}) \cdot \psi(\mathbf{u})$ for some mapping ψ. An example of this is the Radial Basis Kernel, $K(\mathbf{x}, \mathbf{u}) = \exp[-(\| \mathbf{x} - \mathbf{u} \|_2^2)]$. To apply the kernel trick method, we just have to replace $\mathbf{x}_i \cdot \mathbf{x}_j$ by $K(\mathbf{x}_i, \mathbf{x}_j)$ in the dual maximization formulation. It is also important to note the following:

Recall that when we used the assumption, $\| \mathbf{x}_i \|_2 \leq 1$, VC-dimension $\leq 1/\delta^2$. However, if $\| \mathbf{x}_i \|_2 \leq R$, VC-dimension $\leq R^2/\delta^2$. This means that there is a tradeoff between radius, R and δ, since the mapping ψ may increase δ, but might also increase R.