COS 511: Foundations of Machine Learning

Rob Schapire
Scribe: John H. White, IV

Lecture # 10
3/6/2003

1 Boosting

Theorem 1 *With probablity* $1-\delta$, $\forall f \in co(H)$, *and* $\forall \theta > 0$ *then*

$$\Pr_D[yf(x) \le 0] \le \Pr_S[yf(x) \le \theta] + O\left(\frac{1}{\sqrt{m}}\sqrt{\frac{\ln(m)\ln(|H|) + \ln\frac{1}{\delta}}{\theta^2}}\right)$$

Proof:

Previous Lecture

The main idea of the argument is to show that:

$$\exists g \in C_N \text{ s.t. for "most" } x,\ |f(x) - g(x)| \le \tfrac{\theta}{2}$$

and also that, with probability at least $1-\delta$,

$$\Pr_D\left[yg(x) \le \frac{\theta}{2}\right] \le \Pr_S\left[yg(x) \le \frac{\theta}{2}\right] + \epsilon_\theta. \quad (1)$$

Then

$$\begin{aligned}
\Pr_D[yf(x) \le 0] &\approx \Pr_D\left[yg(x) \le \frac{\theta}{2}\right] & (2)\\
&\approx \Pr_S\left[yg(x) \le \frac{\theta}{2}\right] & (3)\\
&\approx \Pr_S[yf(x) \le \theta] & (4)
\end{aligned}$$

This Lecture

$$\begin{aligned}
\Pr_D[yf(x) \le 0] &\le \Pr_{D,g}\left[yf(x) \le 0 \wedge yg(x) > \frac{\theta}{2}\right] + \Pr_{D,g}\left[yg(x) \le \frac{\theta}{2}\right] & (5)\\
&\le e^{-\frac{N\theta^2}{8}} + E_g\left[\Pr_D\left[yg(x) \le \frac{\theta}{2}\right]\right] & (6)\\
&\le e^{-\frac{N\theta^2}{8}} + E_g\left[\Pr_S\left[yg(x) \le \frac{\theta}{2}\right]\right] + \epsilon_\theta & (7)\\
&\le \Pr_S[yf(x) \le \theta] + 2e^{-\frac{N\theta^2}{8}} + \sqrt{\frac{\ln\left[\frac{(N+1)|H|^N}{\delta}\right]}{m}} & (8)\\
&\le \Pr_S[yf(x) \le \theta] + O\left(\frac{1}{\sqrt{m}}\sqrt{\frac{\ln(m)\ln(|H|) + \ln\frac{1}{\delta}}{\theta^2}}\right) & (9)
\end{aligned}$$

The justification of going from equation 6 to 7 is the following:

$$\Pr_D\left[yg(x)\leq\frac{\theta}{2}\right]\leq\Pr_S\left[yg(x)\leq\frac{\theta}{2}\right]+\epsilon_\theta \tag{10}$$

The justification of going from equation 7 to 8 is the following:

$$\begin{aligned}E_g\left[\Pr_S\left[yg(x)\leq\frac{\theta}{2}\right]\right] &= \Pr_{S,g}\left[yg(x)\leq\frac{\theta}{2}\wedge yf(x)>\theta\right]+\Pr_{S,g}\left[yg(x)\leq\frac{\theta}{2}\wedge yf(x)\leq\theta\right]\\ &\leq e^{-\frac{N\theta^2}{8}}+\Pr_S\left[yf(x)\leq\theta\right] \end{aligned}\tag{11}$$

using the same argument as used in the last lecture.

The justification of going from equation 8 to 9 is by defining N as follows:

$$N=\frac{4}{\theta^2}\ln\left(\frac{m}{\ln|H|}\right) \tag{12}$$

■

In boosting we can prove the following:

$$\Pr_S\left[yf(x)\leq\theta\right]\leq\prod_{t=1}^{T}\sqrt{2\epsilon_t^{1-\theta}\left(1-\epsilon_t\right)^{1+\theta}} \tag{13}$$

So if $\epsilon_t\leq\frac{1}{2}-\gamma$ then

$$\Pr_S\left[yf(x)\leq\theta\right]\leq\prod_{t=1}^{T}\sqrt{2\epsilon_t^{1-\theta}\left(1-\epsilon_t\right)^{1+\theta}}\leq\left(\sqrt{\left(1-2\gamma\right)^{1-\theta}\left(1+2\gamma\right)^{1+\theta}}\right)^T. \tag{14}$$

The base of the exponential in this last term is strictly less than 1 when $\theta<\gamma$. So $\Pr_S\left[yf(x)\leq\theta\right]$ goes to 0 as $T\to\infty$ when $\theta<\gamma$.

When using boosting there are usually two approaches. The first is to use a pretty good weak learning algorithm and then boost it to make it better. The second is to use a truly weak learning algorithm and then boost it to make it better. For instance, when learning to classify emails as spam, we might use weak classifiers that make predictions based on the presence of key words or phrases (e.g., "if *buy now* occurs in the email, then predict that it's spam").

2 Support Vector Machines (SVM)

The idea of support vector machines is due to Vapnik. The idea of an SVM is to learn a hyperplane that classifies the data in the best way. The data for an SVM is

$$S=\{(\vec{x}_1,y_1),\ldots,(\vec{x}_m,y_m)\} \tag{15}$$

$$\vec{x}_i\in\mathbb{R}^n \tag{16}$$

$$\|\vec{x}_i\|_2 \leq 1 \tag{17}$$

$$y_i \in \{+1, -1\} \tag{18}$$

The set S contains all the training examples with each example coming from $\mathbb{R}^n$ and being labeled by y_i. Equation 17 means that the Euclidean length of each example is less than or equal to 1. An SVM can be visuallized by the following figure.

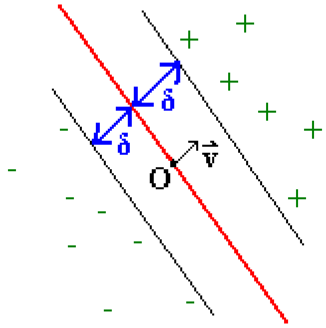

Fig. 1

As Fig. 1 shows an SVM learns the red hyperplane that separates the positive and negative examples. This hyperplane passes through the origin O and has the normal vector $\vec{v}$. The goal of an SVM is to maximize the distance from the plane to each of the examples. This distance is labeled by δ. An SVM then classifies a point $\vec{x}$ as follows

$$\begin{aligned}
\vec{v} \cdot \vec{x} &> 0 \quad \text{means the point } \vec{x} \text{ is above the hyperplane} && (19)\\
\vec{v} \cdot \vec{x} &= 0 \quad \text{means the point } \vec{x} \text{ is on the hyperplane} && (20)\\
\vec{v} \cdot \vec{x} &< 0 \quad \text{means the point } \vec{x} \text{ is below the hyperplane} && (21)
\end{aligned}$$

One interesting observation about SVM's is that dot products are very prevalent. The sign of $\vec{v} \cdot \vec{x}$ determines which side of the hyperplane a point is on. Also the magitude of $\vec{v} \cdot \vec{x}$ gives the distance from the hyperplane to the point. As with boosting SVM's have a margin. The margin is defined as $y\,(\vec{v} \cdot \vec{x})$ and this is positive if the point is classified correctly. Also you want all the margin values to be large as described by $y\,(\vec{v} \cdot \vec{x}) \geq \delta$. Assuming the data is linearly separable the goal of an SVM is to find $\vec{v}$ and maximize $\delta > 0$ such that $\|\vec{v}\|_2 = 1$ and $y\,(\vec{v} \cdot \vec{x}) \geq \delta$. When this inequality holds with equality, $\vec{x}$ is called a "support vector". It is also interesting to note that if a linear threshold function can be found with δ then the $VCdim \leq \frac{1}{\delta^2}$.

3 SVM vs Boosting

	SVM	Boosting
Examples	$\vec{x} \in \mathbb{R}^n$ $\|\vec{x}\|_2 \leq 1$	$\vec{h}(x) \equiv (h_1(x), h_2(x), \ldots, h_h(x))_{h \in H}$ $\left\|\vec{h}(x)\right\|_\infty = \max_j \|h_j(x)\| \leq 1$
Weights	$\|\vec{v}\|_2 = 1$	$\vec{a} = (a_1, a_2, \ldots)$ $\|\vec{a}\|_1 = \sum a_j = 1$
Prediction	$sign(\vec{v} \cdot \vec{x})$	$sign\left(\sum_t a_t h_t(x)\right) = sign\left(\vec{a} \cdot \vec{h}(x)\right)$
Margin	$y(\vec{v} \cdot \vec{x})$	$y \sum_t a_t h_t(x) = y\vec{a} \cdot \vec{h}(x)$

4 Solving an SVM

In order to solve an SVM you want to solve the following problem

Find $\vec{v}$ and max $\delta > 0$
s.t.
(1) $\|\vec{v}\|_2 = 1$

(2) $\forall i \;\; y_i(\vec{v} \cdot \vec{x}_i) \geq \delta \Leftrightarrow y_i \left(\underbrace{\frac{\vec{v}}{\delta}}_{\vec{w}} \cdot \vec{x} \right) \geq 1 \Leftrightarrow y_i(\vec{w} \cdot \vec{x}_i) \geq 1$

The above equation can be written into the following more simplified form to aid in solving it.

$\min \frac{1}{2} \|w\|^2$
s.t.
$\forall i \;\; y_i(\vec{w} \cdot \vec{x}_i) - 1 \geq 0$

The above form of the problem is the form most used in the literature. In order to solve this problem Lagrangians are used. This is because the solution to the problem occurs at the saddle point of the Lagrangian. The Lagrangian is defined as follows

$$L(\vec{w}, \vec{\alpha}) = \frac{1}{2} \|\vec{w}\|^2 - \sum_i \alpha_i \left[y_i (\vec{w} \cdot \vec{x}_i) - 1 \right] \tag{22}$$

In equation 22 the α_i's are call the Lagrange multipliers. The solution of the SVM is when $L(\vec{w}, \vec{\alpha})$ is minimized over $\vec{w}$ and maximized over $\alpha_1, \ldots, \alpha_m$ where $\alpha_i \geq 0$. To begin solving the SVM you first minimize over $\vec{w}$ by setting the partial derivative equal to zero as follows

$$\frac{\partial L}{\partial w_j} = w_j - \sum_i \alpha_i y_i x_{ij} = 0 \tag{23}$$

$$w_j = \sum_i \alpha_i y_i x_{ij} \tag{24}$$

$$\vec{w} = \sum_i \alpha_i y_i \vec{x}_i \tag{25}$$

Equation 25 shows that the solution must be a linear combination of the examples. So now to continue the solution equation 25 is substituted into equation 22 and the following is solved

$$\max_{\alpha_i \geq 0} \alpha_i \quad L = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \vec{x}_i \vec{x}_j \tag{26}$$

There are algorithm packages that exist to solve equation 26 so choose your favorite and the SVM is solved.

5 SVM Solution Observations

By using the Kuhn-Tucker conditions at the solution, $\alpha_i \left[y_i \left(\vec{w} \cdot \vec{x}_i \right) - 1 \right] = 0$. This means that $\alpha_i \neq 0 \Rightarrow y_i \left(\vec{w} \cdot \vec{x}_i \right) = 1 \Leftrightarrow$ "support vector". Because of this the solution $\vec{w}$ depends only on the support vectors. Also by using the solution to the homework it can be said with probablity at least $1 - \delta$, that the $err \leq \left(\frac{k \ln m + \ln \frac{1}{\delta}}{m-k} \right)$ where k is the number of support vectors. This analysis is very nice because the error doesn't depend upon the number of dimensions of the input.

6 What if the data is not linearly separable?

If the data is not linearly separable there are two main options.

6.1 Allow soft margins

Allowing soft margins means that if a training point is on the wrong side of the hyperplane then a cost will be applied to the point. This changes the problem into the following

$$\begin{aligned}
&\min \tfrac{1}{2} \left\| \vec{w} \right\|^2 + C \sum_i \xi_i \\
&\text{s.t.} \\
&\qquad \forall i \;\; y_i \left(\vec{w} \cdot \vec{x}_i \right) \geq 1 - \xi_i \\
&\qquad \forall i \;\; \xi_i \geq 0
\end{aligned}$$

In the above problem C is a parameter chosen by the user.

6.2 Increase Dimensionality

By increasing the dimensionality of the data the likelihood of the data becoming linearly separable increases dramatically. In the Lagrangian we only use dot products. Next lecture, we will show how to use this fact to avoid paying a high price for increasing the dimensionality of the data.