COS 511: Foundations of Machine Learning

Rob Schapire
Scribe: Juhua Zhu

Lecture #11
March 11, 2003

1 Review of SVM - The Optimal Hyperplane

In the previous lecture, we introduced SVM. The simple linearly separable case is:
Given $(\vec{x}_1, y_1), \ldots, (\vec{x}_m, y_m)$, $\vec{x}_i \in \mathbb{R}^n$, $\|\vec{x}_i\|_2 \leq 1$, $y_i \in \{-1, +1\}$, we want to maximize the margin of a hyperplane classifier:

$$max \;\; \delta > 0 \tag{1}$$
$$s.t. \;\; \|\vec{v}\|_2 = 1 \tag{2}$$
$$y_i(\vec{v} \cdot \vec{x}_i) \geq \delta$$

which is equivalent to

$$min \;\; \frac{1}{2}\|\vec{w}\|_2^2 \tag{3}$$
$$s.t. \;\; y_i(\vec{w} \cdot \vec{x}_i) \geq 1 \tag{4}$$

The dual of the quadratic programming problem is

$$max \;\; \sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j (\vec{x}_i \cdot \vec{x}_j) \tag{5}$$
$$s.t. \;\; \alpha_i \geq 0 \tag{6}$$

Then the solution to this optimization problem has the form

$$\vec{w} = \sum_i \alpha_i y_i \vec{x}_i \tag{7}$$

2 SVM - Nonseparable Case

When the examples are not linearly separable, we may add slacks ξ_i to the optimization problem as described in the previous lecture. Or, we may take the examples, blow up to the higher-dimensional space, with the hope that in the higher-dimensional space, these examples are linearly separable.

Example 1:

If there is no hyperplane through origin to separate the examples, just take every example of $\vec{x} = (x_1, y_1)$ and map it to $(x_1, x_2, 1)$ by adding a dummy variable 1. Thus, the hyperplane through origin $ax_1 + bx_2 = 0$ becomes $ax_1 + bx_2 + c = 0$, which does not have

to go through origin.

Example 2:

If the examples are separable by a circle instead of by a line, say the circle is $(x_1-a_1)^2+(x_2-a_2)^2=c^2$, then just take every example of $\vec{x}=(x_1,y_1)$ and map it to $(1,x_1,x_2,{x_1}^2,{x_2}^2)$. By the way, if the examples are separated by hyperbolas or parabolas, we may add another cross term of x_1x_2 to the mapped vector. We denote this as $x \mapsto \psi(x)$.

Generalized to the case of n variables and degree d, $dim(\psi(x)) = O(n^d)$.

But, is it a terrible idea to go to a very high dimensional space? There is the problem of the "curse of dimensionality":
1. statistically, higher dimension means more data needed
2. computationally, even if the run-time is polynomial in dimension, it runs slowly

Well, for SVM, there is the miracle that it can get around these two problems. To address the first problem, there are two ways thinking of the errors of SVM:

- the errors only depend on the number of support vectors, not depend on the number of dimension
- $VCdim \leq 1/\delta^2$, where δ is the separation, again, not explicitly dependent on the number of dimensions.

To address the second problem, we should look into the way SVM works. The dependence of the x's and the examples is only through inner product. So we only need to worry about how to compute inner products efficiently. In the example above, the inner product in the higher dimensional space is:

$$\begin{aligned} \psi(\vec{x})\cdot\psi(\vec{u}) &= (1+u_1x_1+u_2x_2+{u_1}^2{x_1}^2+{u_2}^2{x_2}^2+u_1x_1u_2x_2) \\ &= 1+z_1+z_2+{z_1}^2+{z_2}^2+z_1z_2 \end{aligned} \tag{8}$$

where $z_1=u_1x_1$, $z_2=u_2x_2$.
Compare to

$$(1+z_1+z_2)^2 = 1+{z_1}^2+{z_2}^2+2z_1+2z_2+2z_1z_2 \tag{9}$$

To make them alike, we add the constant of 2 and then have

$$\begin{aligned} \psi(\vec{x})\cdot\psi(\vec{u}) &= (1+2u_1x_1+2u_2x_2+{u_1}^2{x_1}^2+{u_2}^2{x_2}^2+2u_1x_1u_2x_2) \\ &= 1+2z_1+2z_2+{z_1}^2+{z_2}^2+2z_1z_2 \end{aligned} \tag{10}$$

Therefore, we need to change the mapping a little bit to $(x_1,x_2)\mapsto(1,\sqrt{2}x_1,\sqrt{2}x_2,{x_1}^2,{x_2}^2,\sqrt{2}x_1x_2)$. That does not change what we can represent at all since we are looking for a linear combination of these terms anyway. We just change the constants to make the math a little bit more convenient.

Now we may write the inner product as

$$\psi(\vec{x})\cdot\psi(\vec{u}) = (1+\vec{x}\cdot\vec{u})^2 \tag{11}$$

When generalized to degree d, just replace the power of 2 with d. Here $\psi(\vec{x})$ is of length n^d and the computational complexity is approximately $O(n + ln(d))$, which means much much faster computation.

$(1 + \vec{x} \cdot \vec{u})^d$ is called a *kernel* $K(x, u)$, in this case, a polynomial kernel. A kernel is a function taking two objects, not necessarily vectors. We can use any such function so long as it satisfies certain properties. We do not even have to know the form of ψ. In particular, kernels K must satisfy Mercer's conditions:

- K must be symmetric, and
- K must be positive definite (thinking of K as a giant matrix)

Now the classifier becomes

$$\begin{aligned} sgn(\vec{w} \cdot \vec{x}) &= sgn(\sum \alpha_i y_i (\vec{x}_i \cdot \vec{x})) \\ &= sgn(\sum \alpha_i y_i K(\vec{x}_i, \vec{x})) \end{aligned} \tag{12}$$

We can plug any K satisfying the Mercer's condition here. By changing the form of the hypothesis, the VC dimension changes.

With $\|\vec{x}_i\|_2 = 1$, we have $VCdim = 1/\delta^2$. If $\|\vec{x}_i\|_2 \leq R$, $x_i \to x_i/R, \delta \to \delta/R$, thus, $VCdim \leq R^2/\delta^2$. $\|\vec{x}\|_2 \leq 1 \Rightarrow \|\psi(\vec{x})\|_2^2 = \psi(\vec{x}) \cdot \psi(\vec{x}) = (1+\vec{x} \cdot \vec{x})^d \leq 2^d$. So R is increasing as we move to higher dimensions, but intuitively, the seap δ is also increasing since it becomes easier to separate the data in higher dimensions. The hope is that δ will increase faster than R.

3 Online Learning Model

Unlike PAC (batch version) learning model, where first we have training examples to train the classifier and then have a test phase, for online learning model, the learner gets one example at a time, and then make prediction. In other words, in online learning model, training and test happen at the same time. The goal is to minimize the number of mistakes. Some possible examples are weather forecast and stock market prediction.

Another difference is, here we do not make any randomization over data and do not make any statistical assumption, and we have an adversary choosing the examples' labels, while in PAC model, examples are random.

Let's start with "learning with expert advice". For example, predict S&P500 based on expert advice. (See Table 1.) On each day, each expert predicts if the S&P500 will go up or down that day. Then the learner combines their predictions into its own prediction. Then at the end of the day, they find out if the market actually went up or down. This is repeated over a sequence of T days.

Assume there is at least one perfect expert who makes correct prediction every time, then the learner is not bad compared to the best expert.

	expert1	expert2	expert3	expert4	learner(master)	outcome
day 1	↑	↑	↓	↑	↑	↑
day 2	↓	↑	↓	↑	↓	↑
⋮						
day T						

Table 1: Learning with expert advice

Goal:

The number of the mistakes of the learner $\leq$ the number of the mistakes of the best expert + small amount (growing slowly compared to the number of days T).

The "best expert" is, looking backward, who makes the least mistakes.

More formally:

N = number of experts

for $t = 1, 2, \ldots, T$

each expert i predicts $\xi_i \in \{0, 1\}$

learner predicts $\hat{y} \in \{0, 1\}$

observe outcome $y \in \{0, 1\}$

Simple case:

One expert makes perfect prediction. Anybody makes a mistake, throw them out.

Halving algorithm:

$\hat{y}$ = majority vote of experts with no mistakes so far,

W = number of experts with no mistakes so far.

Initially, W = N.

Learner makes a mistake $\Rightarrow \geq 1/2$ surviving experts are wrong $\Rightarrow \geq 1/2$ eliminated.

After 1 mistake, $W \leq N/2$;

after 2 mistakes,$W \leq N/4$;

⋮

after m mistakes, $W \leq (\frac{1}{2})^m N$.

So, $1 \leq W \leq (\frac{1}{2})^m N \Rightarrow m \leq \log_2 N$

Related to PAC - Special case:

$H = \{h_1, \ldots, h_N\}$, target concept $c \in H$.

On each round, observe an example x (not have to be random).

$\xi_i = h_i(x)$

predict $\hat{y}$

observe $y = c(x)$

number of mistakes $\leq \lg N = \lg |H|$

Let

$$M_A(H) = max_{c, x_1, \ldots, x_t}(number of mistakes) \tag{13}$$

$$Opt(H) = min_{det. alg's A} M_A(H) \tag{14}$$

Theorem:

$$Opt(H) \leq M_{halving}(H) \leq \lg |H|$$

Question:

How does the VC dimension relate to these quantities?