COS 511: Foundations of Machine Learning

Rob Schapire
Scribe: Yingying Fan

Lecture #23
5 May 2006

1 Game Theory

Let us first consider the game that every child knows, called Paper-Scissors-Rock. To refresh our memory, this is a two-person game in which at the count of three each player declares either Paper, Scissors, or Rock. If both players declares the same object, then the round is a draw. But Paper loses to Scissors (since scissors can cut a piece of paper), Scissors loses to Rock (since a rock can dull scissors), and finally Rock loses to Paper (since a piece of paper can cover up a rock). For convenience, we are going to refer the first player as the row player and second player as the column player. We say that the loss of the row player is 1 if he loses, 1/2 is it is a draw and 0 if he wins. Clearly, for this game if we enumerate the actions of declaring Rock, Paper, or Scissors as 1,2,3, respectively, then the loss matrix of the row player is

$$\begin{pmatrix} 1/2 & 1 & 0 \\ 0 & 1/2 & 1 \\ 1 & 0 & 1/2 \end{pmatrix}.$$

With this matrix, neither player has a deterministic (pure) winning strategy. If the row player were always to declare Paper, then the column player could counter by always declaring Scissors and guaranteeing himself a win in every round. In fact, if the row player were to stick to any specific declaration, then the column player would eventually get wise to it and respond appropriately and guarantee that he wins. Of course, the same logic applies to the column player. Hence, we should use a mixed strategy, which is also called a randomized strategy. More generally, in a two-person, zero-sum game, we use the following notation:

- i: pure row strategy
- j: pure column strategy
- P: mixed row strategy
- Q: mixed column strategy

In each round, row player chooses strategy i with probability $P(i)$ and column player choose strategy j with probability $Q(j)$. So the expected loss of the row player in this round is

$$\sum_{i,j} M(i,j)P(i)Q(j) = P'MQ = M(P,Q),$$

where P' above means the transpose of P.

2 The Minimax Theorem

Suppose row player plays first and he adopts strategy P. Then the column player's best defense is to use the strategy Q which achieves the following maximum:

$$\max_Q M(P,Q).$$

Since for any given P the column will adopt the above Q, it follows that the column player should employ a strategy P^* that attains the following minimum:

$$\min_P \max_Q M(P,Q). \tag{1}$$

In other words, this is the loss of row player if two players play sequentially with row player plays first. If instead, column player plays first, then the loss of the row player is

$$\max_Q \min_P M(P,Q). \tag{2}$$

It is clear that going second is better than going first. So

$$\max_Q \min_P M(P,Q) \leq \min_P \max_Q M(P,Q). \tag{3}$$

But in fact we have a stronger result, which says the above two losses (1) and (2) are actually equal to each other.

Theorem 1 *(Von Neumann's Minimax theorem)*

$$\max_Q \min_P M(P,Q) = \min_P \max_Q M(P,Q).$$

Before we see the proof, let's look at what this means intuitively. Define $v \equiv \max_Q \min_P M(P,Q)$ $= \min_P \max_Q M(P,Q)$. Then v is often called the "value" of the game. Theorem 1 tells us that

- there exists P^* such that for every Q, the expected loss $M(P^*,Q) \leq v$; thus, v is the highest loss the column player can force, even knowing that the row player is playing P^*;

- furthermore, this is the best possible since there exists Q^* such that for every P (including P^*), the expected loss $M(P,Q^*) \geq v$.

Now we are going to use a machine learning algorithm to play repeated games and also to prove theorem 1. Consider the following learning scenario for repeated play of a matrix game M:

```
For $t = 1, \cdots, T$
    row (learner) chooses $P_t$
    column (environment) chooses $Q_t$ (knowing $P_t$)
    learner suffers loss $M(P_t, Q_t)$.
```

The total loss of row player is $\sum_{t=1}^{T} M(P_t, Q_t)$. But if we fix P in each round above, then the total loss is $\sum_{t=1}^{T} M(P, Q_t)$. So our goal is to make

$$\sum_t M(P_t, Q_t) \leq \min_P \sum_t M(P, Q_t) + \text{small amount}.$$

To this end, we introduce a simple algorithm for the set up above (n is the number of choices of row player)

```
Initially $P_1(i) = 1/n$ for $i = 1, \cdots, n$
On each round

    row (learner) chooses $P_{t+1}(i) = P_t(i)\beta^{M(i,Q_t)}/\text{normalization}$
    column (environment) chooses $Q_t$ (knowing $P_t$)
    learner suffers loss $M(P_t, Q_t)$.
```

This is just a direct generalization of earlier algorithms we have seen for learning with expert advice. By using the similar methods as before, we can prove that the total loss of the above algorithm satisfies

$$\sum_t M(P_t, Q_t) \leq a_\beta \min_P \sum_t M(P, Q_t) + c_\beta \ln n,$$

where $a_\beta = \frac{\ln 1/\beta}{1-\beta}$ and $c_\beta = \frac{1}{1-\beta}$. By choosing proper β, we can change the above inequality to

$$\frac{1}{T} \sum_t M(P_t, Q_t) \leq \min_P \frac{1}{T} \sum_t M(P, Q_t) + \Delta_T, \tag{4}$$

where $\Delta_T = O(\sqrt{\ln n/T})$.

To apply this algorithm to the proof of the minimax theorem, we are going to choose $Q_t = \arg\max_Q M(P_t, Q)$. Moreover, we define

$$\bar{P} = \frac{1}{T} \sum_t P_t, \ \bar{Q} = \frac{1}{T} \sum_t Q_t.$$

We need to prove

$$\max_Q \min_P M(P, Q) \geq \min_P \max_Q M(P, Q) \tag{5}$$

This together with (3) leads to the minimax theorem. Notice that

$$\begin{aligned}
\min_P \max_Q P'MQ &\leq \max_Q \bar{P}'MQ \\
&= \max_Q \frac{1}{T}\sum_t P_t'MQ \\
&\leq \frac{1}{T}\sum_t \max_Q P_t'MQ \\
&= \frac{1}{T}\sum_t P_t'MQ_t \\
&\leq \frac{1}{T}\min_P \sum_t P'MQ_t + \Delta_T \text{ (by (4))} \\
&= \min_P P'M\bar{Q} + \Delta_T \\
&\leq \max_Q \min_P P'MQ + \Delta_T.
\end{aligned}$$

Since both $\min_P \max_Q P'MQ$ and $\max_Q \min_P P'MQ$ are independent of T and $\Delta_T \to 0$ as $T \to \infty$, we get the inequality (5) by setting $T \to \infty$. This completes the proof.

3 On-line Learning Model

The algorithm for on-line learning model is

```
For $t = 1, \cdots, T$
    Observe $x_t \in X$
    predict $\hat{y}_t \in \{0, 1\}$
    observe $c(x_t)$
    mistake if $\hat{y}_t \neq c(x_t)$.
```

We have the following result for this algorithm

$$\#\text{ Mistakes} \leq \min_{h \in \mathcal{H}}(\#\text{ mistakes using } h) + \text{ small amount.}$$

This result can be obtained by applying the algorithm defined before. We index the rows of M by using the hypotheses h and the columns of M by using the samples x, and define M as

$$M(h, x) = \begin{cases} 1, & \text{if } h(x) \neq c(x) \\ 0, & \text{else.} \end{cases}$$

Then the above error bound can be easily obtained.

4 Boosting Setting

The algorithm for Boosting is

```
For $t = 1, \cdots, T$
    construct $D_t$ over $X$
    get $h_t \in \mathcal{H}$
    $Pr_{x \sim D_t}(h_t(x) \neq c(x)) \leq \frac{1}{2} - \gamma$.
```

We define the matrix M the same as that in the on-line learning model, but change the roles of row and column players (since boosting chooses distribution over rows). Define

$$\tilde{M} \triangleq 1 - M'.$$

Then

$$\tilde{M}(x,h) = \begin{cases} 1, & \text{if } h(x) \neq c(x) \\ 0, & \text{else.} \end{cases}$$

So with this new matrix, we go back to the boosting setting.

```
On round $t$
    compute $P_t$
    use $D_t = P_t$
    Let $Q_t$ be pure strategy concentrated on $h_t$.
```

Then

$$\begin{aligned} \tilde{M}(P_t, Q_t) = \tilde{M}(P_t, h_t) &= \sum_x P_t(x)\tilde{M}(x, h_t) \\ &= Pr_{x \sim P_t}[h_t(x) = c(x)] \geq \frac{1}{2} + \gamma. \end{aligned}$$

Hence,

$$\frac{1}{2} + \gamma \leq \frac{1}{T}\sum_t \tilde{M}(P_t, h_t) \leq \min_P \frac{1}{T}\sum_t \tilde{M}(P, h_t) + \Delta_T = \min_x \frac{1}{T}\sum_t \tilde{M}(x, h_t) + \Delta_T.$$

Therefore, for any x, if T is large enough

$$\frac{1}{T}\sum_t \tilde{M}(x, h_t) \geq \frac{1}{2} + \gamma - \Delta_T > \frac{1}{2}.$$

This implies that majority voting of $h_1, \cdots, h_T$ has training error 0.