COS 511: Theoretical Machine Learning

Lecturer: Rob Schapire
Scribe: Jordan Ash

Lecture #24
May 1, 2014

1 Review of Game Theory:

Let M be a matrix with all elements in $[0, 1]$. Mindy (called the row player) chooses row i while Max (called the column player) chooses column j. In this case, from Mindy's expected loss is:

$$Loss = M(i, j)$$

Alternatively, Mindy could select a move randomly from a distribution P over the rows and Max could select a move randomly from a distribution Q over the columns. Here, the expected loss for Mindy is:

$$Loss = \sum_{i,j} P(i)M(i, j)Q(j) = P^T MQ = M(P, Q)$$

P and Q are called "mixed strategies," while i and j are called "pure strategies."

2 Minimax Theorem:

In some games, such as Rock, Paper, Scissors, players move at exactly the same time. In this way, both players have the same information available to them at the time of moving. Now we suppose that Mindy plays first, followed by Max. Max knows the P that Mindy chose, and further knows $M(P, Q)$ for any Q he chooses. Consequently, he chooses a Q that maximizes $M(P, Q)$. Because Mindy knows that Max will choose $Q = \arg\max_Q M(P, Q)$ for any P she chooses, she selects a P that minimizes $\max_Q M(P, Q)$. Thus, if Mindy goes first, she could expect to suffer a loss of $\min_P \max_Q M(P, Q)$. Overall, it may initially seem like the player to go second has an advantage because she has more information available to her. From Mindy's perspective again, this leads to:

$$\max_Q \min_P M(P, Q) \leq \min_P \max_Q M(P, Q)$$

So Mindy playing after Max seems to be better than if the two play in reverse order. However, John von Neumann showed that the expected outcome of a game is always the same, regardless of the order in which players move.

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

Here, v denotes the value of the game. This may seem counterintuitive, because the player that goes second has more information available to her at the time of choosing a move. We will prove the above statement using an online learning algorithm. Let:

$$P^* = \arg\min_P \max_Q M(P, Q)$$
$$Q^* = \arg\max_Q \min_P M(P, Q)$$

Then,

$$\forall Q : M(P^*, Q) \leq v \tag{1}$$
$$\forall P : M(P, Q^*) \geq v \tag{2}$$

In other words, for some optimal P^*, the maximum loss that Max could cause is bounded by v and Mindy's loss is at least v, regardless of the particular strategies they choose.

If we had knowledge of M, we might be able to find P^* by employing techniques from linear programming. However, we don't necessarily have this knowledge, and even if we did, M could be massively large. Further, P^* applies here only for opponents that are perfectly adversarial, so it doesn't account for an opponent that might make mistakes. Thus, it makes sense to try to learn M and Q iteratively.

We do this with the following formulation:

for $t = 1, \ldots, T$
    Mindy chooses P_t
    Max chooses Q_t (with knowledge of P_t)
    Mindy observes $M(i, Q_t) \forall i$
    Loss $= M(P_t, Q_t)$
end

Clearly, the total loss of this algorithm is simply $\sum_{t=1}^{T} M(P_t, Q_t)$. We want to be able to compare this loss to the best possible loss that could have been achieved by fixing any single strategy for all T iterations. In other words, we want to show:

$\sum_{t=1}^{T} M(P_t, Q_t) \leq \min_P \sum_{t=1}^{T} M(P, Q_t) +$ [Small Regret Term]

2.1 Multiplicative Updates

Suppose we use a multiplicative weight algorithm that updates weights in the following way, where n is the number of rows in matrix M:

$$\beta \in [0, 1) \tag{3}$$

$$P_1(i) = \frac{1}{n} \forall i \tag{4}$$

$$P_{t+1}(i) = \frac{P_t(i)\beta^{M(i,Q_t)}}{\text{Normalizing Constant}} \tag{5}$$

Our algorithm is similar to the weighted majority algorithm. The idea is decrease the probability of choosing a particular row proportionally to the loss suffered by selecting that row. After making an argument using potentials, we could use this algorithm to obtain the following bound:

$$\sum_{t=1}^{T} M(P_t, Q_t) \leq \alpha_\beta \min_P \sum_{t=1}^{T} M(P, Q_t) + c_\beta \ln(n) \tag{6}$$

where $\alpha_\beta = \frac{ln(\frac{1}{\beta})}{1-\beta}$ and $c_\beta = \frac{1}{1-\beta}$.

2.2 Corollary

We can choose β such that:

$$\frac{1}{T} \sum_{t=1}^{T} M(P_T, Q_T) \leq \min_P \frac{1}{T} \sum_{t=1}^{T} M(P, Q_t) + \Delta_T \tag{7}$$

where $\Delta_T = O(\sqrt{\frac{ln(n)}{T}})$, which goes to zero for large T. In other words, the loss suffered by Mindy per round approaches the optimal average loss per round. We'll use this result to prove the Minimax theorem.

2.3 Proof

Suppose that Mindy uses the above algorithm to choose P_t, and that Max chooses Q_t such that $Q_t = \arg\max_Q M(P_T, Q)$, maximizing Mindy's loss. Also, let:

$$\bar{P} = \frac{1}{T} \sum_{t=1}^{T} P_t \tag{8}$$

$$\bar{Q} = \frac{1}{T} \sum_{t=1}^{T} Q_t \tag{9}$$

We also know intuitively, as mentioned before, that $\max_Q \min_P M(P, Q) \leq \min_P \max_Q M(P, Q)$, because as stated earlier, the player that goes second has more information available to her.

To show equality, which would prove the Minimax theorem stated earlier, it's enough to show that $\max_Q \min_P M(P,Q) \geq \min_P \max_Q M(P,Q)$ also.

$$\min_P \max_Q P^T M Q \leq \max_Q \bar{P}^T M Q$$

By definition of $\bar{P}$:

$$= \max_Q \frac{1}{T} \sum_{t=1}^{T} P_t^T M Q$$

By convexity:

$$\leq \frac{1}{T} \max_Q \sum_{t=1}^{T} P_t^T M Q$$

By definition of Q_t:

$$= \frac{1}{T} \sum_{t=1}^{T} P_t^T M Q_t$$

By corollary 2.2:

$$\leq \min_P \frac{1}{T} \sum_{t=1}^{T} P^T M Q_t + \Delta_T$$

By definition of $\bar{Q}$:

$$= \min_P P^T M \bar{Q} + \Delta_T$$
$$\leq \max_Q \min_P P^T M Q + \Delta_T$$

The proof is finished because Δ_T goes to zero as T gets large. This proof also shows that:

$$\max_Q \bar{P}^T M Q \leq v + \Delta_T$$

where $v = \max_Q \min_P P^T M Q$. If we take the average of the P_t terms computed at each round of the algorithm, we get something within Δ_T of the optimal value. Because Δ_T goes to zero for large values of T, we can get closer to the optimal strategy by simply increasing T. In other words, this strategy becomes more and more optimal as the number of rounds T increases. For this reason, $\bar{P}$ is called an approximate min max strategy. A similar argument could be made to show that $\bar{Q}$ is an approximate max min strategy.

3 Relation to Online Learning

In order to project our analysis into an online learning framework, consider the following problem setting:

for $t = 1, \ldots, T$
    Observe x_t from X
    Predict $\hat{y}_t \in \{0, 1\}$
    Observe true label $c(x_t)$
end

Here we consider each hypothesis h as being an expert from the set of all hypotheses H. We want to show that:

number of mistakes $\leq$ number of mistakes of best h + [Small Regret Term]

We set up a game matrix M where $M(i, j) = M(h, x) = 1$ if $h(x) \neq c(x)$ and 0 otherwise. Thus, the size of this matrix is $|H| \cdot |X|$. Given an x_t, the algorithm must choose some P_t, a distribution used to predict x_t's label. h is chosen according to the distribution P_t, and then $\hat{y}_t$ is chosen as $h(x_t)$. Q_t in this context is the distribution concentrated on x_t (is 1 at x_t and 0 at all other $x \in X$). Consequently:

$$\begin{aligned}
&\sum_{t=1}^{T} M(P_t, x_t) \\
&= \text{E[number of mistakes]} \\
&\leq \min_h \sum_{t=1}^{T} M(h, x_t) + \text{[Small Regret Term]}
\end{aligned}$$

Notice that $\min_h \sum_{t=1}^{T} M(h, x_t)$ is equal to the number of mistakes made by the best hypothesis h. If we substitute $M(P_t, x_t)$ with $\sum_h P_t(h) \cdot 1\{h(x_t) \neq c(x_t)\} = Pr[h(x) \neq c(x)]$ above, we obtain the same bound as was found in the previous section.

4 Relation to Boosting

We could think of boosting as a game between the boosting algorithm and the weak learner it calls. Consider the following problem:

for $t = 1, \ldots, T$
    The boosting algorithm selects a distribution D_t over the training set samples X
    The weak learner chooses a hypothesis h_t
end

Here we assume that all weak learners h_t obey the weak learning assumption, i.e. that $\Pr_{(x,y)\sim D_t}[h_t(x) \neq y)] \leq \frac{1}{2} - \gamma$ and $\gamma > 0$. We could define the game matrix M' in terms of the matrix M used in the last section. However, here we want a distribution over the X samples rather than over the hypotheses, so we need to transpose and normalize M.

$M' = 1 - M^T$

In other words, $M'(i,j) = M'(x,h) = 1$ if $h(x) = c(x)$ and 0 otherwise. Here, $P_t = D_t$, and Q_t is a distribution fully concentrated on the particular h_t chosen by the weak learner. We could apply our same analysis from the multiplicative weights algorithm:

$\frac{1}{T}\sum_{t=1}^{T} M'(P_t, h_t) \leq \min_x \frac{1}{T}\sum_{t=1}^{T} M'(x, h_t) + \Delta_T$

Also,

$M'(P_t, h_t) = \sum_x P_T(x) \cdot 1\{h(x) = c(x)\} = Pr[h_t(x) = c(x)] \geq \frac{1}{2} + \gamma$

Combining these facts:

$$\begin{aligned}
\frac{1}{2} + \gamma &\leq \frac{1}{T}\sum_{t=1}^{T} M'(x, h_t) \\
&\leq \min_x \frac{1}{T}\sum_{t=1}^{T} M'(x, h_t) + \Delta_T
\end{aligned}$$

Rearranging,

$$\forall x : \frac{1}{T}\sum_{t=1} TM'(x, h_t) \geq \frac{1}{2} + \gamma - \Delta_T > \frac{1}{2}$$

which is again true because Δ_T approaches 0 as T gets large. In other words, we have found that over $\frac{1}{2}$ of the weak hypotheses correctly classify any x when T gets sufficiently large. Because the final hypothesis is just a majority vote of these weak learners, we have proven that the boosting algorithm drives training error to zero when enough weak learners are employed.