COS 511: Theoretical Machine Learning

Lecturer: Rob Schapire — Lecture #24
Scribe: Jad Bechara — May 2, 2018

1 Review of Game Theory

The games we will discuss are two-player games that can be modeled by a game matrix M with all elements in $[0, 1]$. Simultaneously, Mindy (row player) chooses a row i and Max (column player) chooses a column j. From Mindy's point of view, her loss will be: $M(i, j)$ (which she is trying to minimize, and which Max is trying to maximize). Choosing a single row or a single column corresponds to playing deterministically, i.e. pure strategies.

Alternatively, we can allow the players to play in a randomized way (known as mixed strategies): simultaneously, Mindy chooses a distribution P over rows (the strategy she ends up playing is $i \sim P$) and Max chooses a distribution Q over columns (the strategy he ends up playing is $j \sim Q$). In this case, Mindy's expected loss (which we sometimes will refer to simply as her loss) is:

$$\sum_{i,j} P(i)M(i,j)Q(j) = P^\top MQ := M(P,Q) \tag{1}$$

where $P(i)$ is the probability that Mindy plays strategy i, and $Q(j)$ is the probability that Max plays strategy j.[1]

2 Minimax Theorem

Suppose that the game is played sequentially. In other words, Mindy first picks a distribution P (over the rows), then, knowing P and in order to maximize his expected gain, Max picks a distribution $Q = \arg\max_Q M(P, Q)$.[2] Now, knowing Max will play in this fashion, Mindy will pick her mixed strategy P to minimize her expected loss, which thus will be: $\min_P \max_Q M(P, Q)$ (where $\min_P$ is over all possible mixed row strategies, and $\max_Q$ is over all possible mixed column strategies). Similarly, if Max plays first, then Mindy's expected loss is : $\max_Q \min_P M(P, Q)$. As we have seen when studying Support Vector Machines, in general, the second player is always at an advantage, i.e.

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

However, in this case, John von Neumann's Minimax theorem shows that for two player zero-sum games (when one player's loss is the other player's gain):

$$\max_Q \min_P M(P,Q) = \min_P \max_Q M(P,Q) := v \tag{3}$$

[1] Note that we denote by $M(i, Q)$ Mindy's loss for playing some row i when Max plays some distribution over columns Q, and vice-versa for $M(P, j)$.

[2] For a fixed distribution P, this function is linear in Q, so the optimal strategy is a pure strategy: $\max_Q M(P, Q) = \max_j M(P, j)$.

where v denotes the value of the game. In what follows, we will prove this theorem using techniques from online learning.

The Minimax theorem implies that $\exists\ P^* : \forall\ Q,\ M(P^*, Q) \leq v$ (regardless of what Max plays, Mindy can ensure a loss of at most v) and that $\exists\ Q^* : \forall\ P,\ M(P, Q^*) \geq v$ (regardless of what Mindy plays, Max can ensure a gain of at least v). In this case, we say that P^* is a minmax strategy and Q^* a maxmin strategy.[3]

In classical game theory, having knowledge of M, one can compute P^* and Q^* immediately (using linear programming). However, things are not always so simple. In some cases, we might not have knowledge of M, in others, M could be very large, and more notably, (P^*, Q^*) assumes both players are "rational" (optimal and adversarial), which is not always the case. Thus, a natural extension of the situation that will allow for online learning, is to allow the game to be played repeatedly. Consider the following online algorithm:[4][5]

Algorithm 1 Learning Protocol

```
INPUT: Game Matrix $M$, Number of rows $n$, Number of iterations $T$.
  for $t = 1, ..., T$ do
      Learner (Mindy) chooses $P_t$
      Environment (Max) chooses $Q_t$ (knowing $P_t$)
      Learner suffers loss $M(P_t, Q_t)$
      Learner observes $M(i, Q_t)\ \forall\ i$
OUTCOME: Learner's total loss is $\sum_{t=1}^{T} M(P_t, Q_t)$.
```

We want to compare this total loss to the loss the learner would have gotten by fixing an optimal strategy P for all T rounds. In other words, we want to prove:

$$\frac{1}{T}\sum_{t=1}^{T} M(P_t, Q_t) \leq \min_P \frac{1}{T}\sum_{t=1}^{T} M(P, Q_t) + \text{small regret}. \tag{4}$$

Note that the first term on the right-hand side of the inequality is at most v, the value of the game (it can be smaller when the other player is sub-optimal).

2.1 Multiplicative Updates

Since the above looks like an online learning problem, we will construct an algorithm similar to the Weighted Majority algorithm (i.e. with multiplicative weights). Knowing that we need to compute P_t on every update, consider the following update rules:

Algorithm 2 Multiplicative Weights (MW)

```
  $\forall\ i:\ P_1(i) = \frac{1}{n}$
  $\forall\ i:\ P_{t+1}(i) = \frac{P_t(i) \cdot \beta^{M(i,Q_t)}}{\text{Normalization}}$ for some fixed $\beta \in [0, 1)$
```

This algorithm is called "Multiplicative Weights" or "MW". Notice that the greater the loss the learner would have suffered if row i had been played, the smaller the probability of choosing that row gets.

[3] Also, (P^*, Q^*) forms a Nash equilibrium.

[4] Everything is viewed from Mindy's perspective.

[5] On each iteration, the learner observes the loss they would have gotten for every row chosen.

Theorem 1. *If we set $\beta = \frac{1}{1+\sqrt{\frac{2\ln n}{T}}}$, then:*

$$\frac{1}{T}\sum_{t=1}^{T} M(P_t, Q_t) \leq \min_P \frac{1}{T}\sum_{t=1}^{T} M(P, Q_t) + \Delta_T \tag{5}$$

where $\Delta_T = \mathcal{O}\left(\sqrt{\frac{\ln n}{T}}\right)$, so that the regret goes to 0 as $T \to \infty$.

Proof. The proof involves an analysis similar to the one done for the Weighted Majority algorithm (potential function argument) and will be omitted. □

2.2 Minimax Theorem Proof

We will next prove the Minimax Theorem using the Multiplicative Weights algorithm and its anaylsis. Suppose that Mindy and Max behave according to the following procedure:

Algorithm 3

```
for $t = 1, ..., T$ do
    Mindy picks $P_t$ using Multiplicative Weights (Algorithm 2)
    Max picks $Q_t = \arg\max_Q M(P_t, Q)$
```

Define $\bar{P} = \frac{1}{T}\sum_{t=1}^{T} P_t,\ \bar{Q} = \frac{1}{T}\sum_{t=1}^{T} Q_t$. Now, note the following chain of inequalities:

$$\begin{aligned}
\min_P \max_Q P^\top M Q &\leq \max_Q \bar{P}^\top M Q & (6)\\
&= \max_Q \frac{1}{T}\sum_{t=1}^{T} P_t^\top M Q & (7)\\
&\leq \frac{1}{T}\sum_{t=1}^{T} \max_Q P_t^\top M Q & (8)\\
&= \frac{1}{T}\sum_{t=1}^{T} P_t^\top M Q_t & (9)\\
&\leq \min_P \frac{1}{T}\sum_{t=1}^{T} P^\top M Q_t + \Delta_T & (10)\\
&= \min_P P^\top M \bar{Q} + \Delta_T & (11)\\
&\leq \max_Q \min_P P^\top M Q + \Delta_T & (12)
\end{aligned}$$

where (6) is due to picking a particular value of P, (7) is by definition of $\bar{P}$, (8) is due to convexity (also, taking the maximum term-by-term can only give a greater or equal quantity), (9) is by definition of Q_t, (10) is by the MW algorithm and Theorem 1, (11) is by definition of $\bar{Q}$, and (12) is by definition of max. Since $\Delta_T = \mathcal{O}\left(\sqrt{\frac{\ln n}{T}}\right)$, taking $T \to \infty$ proves the result.

Notice that by taking the right-hand term of (6) and the inequality in (12) we also get: $\max_Q \bar{P}^\top MQ \leq \max_Q \min_P P^\top MQ + \Delta_T$. Or, in other words:

$$\max_Q M(\bar{P}, Q) \leq v + \Delta_T \tag{13}$$

This implies that we can approximate the value of the game, v, using Algorithm 3 (i.e. $\bar{P}$ is an approximate min max strategy).

Let us now observe how Online Learning and Boosting an be seen as special cases of the above result.

3 Relation to Online Learning

Consider the following version of Online Learning:

```
Algorithm 4
INPUT: Finite hypothesis space $\mathcal{H}$, Finite domain space $\mathcal{X}$, Number of iterations $T$.
  for $t = 1, ..., T$ do
      Get $x_t \in \mathcal{X}$
      Predict $\hat{y}_t \in \{0, 1\}$
      Observe true label $c(x_t)$ (mistake if $\hat{y}_t \neq c(x_t)$)
```

We want to show: $\#(\text{mistakes}) \leq \min_{h \in \mathcal{H}} \#(\text{mistakes by } h) + \text{small regret}$. In order to do so, let's construct a game matrix M, where the rows are indexed by $h \in \mathcal{H}$, and the columns are indexed by $x \in \mathcal{X}$. Define $M(h, x) = \mathbb{1}[h(x) \neq c(x)]$.

Now, run the Multiplicative Weights algorithm on the matrix M. On each round t, MW will use the distribution P_t to pick a hypothesis $h \sim P_t$ and predict $\hat{y}_t = h(x_t)$. Then, let Q_t be concentrated on x_t (i.e. probability 1 on column x_t and 0 everywhere else). From the bound on the MW algorithm, we get:

$$\sum_{t=1}^{T} M(P_t, x_t) \leq \min_{h \in \mathcal{H}} \sum_{t=1}^{T} M(h, x_t) + \mathcal{O}\left(\sqrt{T \ln |\mathcal{H}|}\right) \tag{14}$$

Note that it is enough to compute the right-hand side over all pure strategies only. Also, notice that $\min_{h \in \mathcal{H}} \sum_{t=1}^{T} M(h, x_t)$ is the number of mistakes made by the best hypothesis in the class, and that:

$$\sum_{t=1}^{T} M(P_t, x_t) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{h \sim P_t}[\mathbb{1}[h(x_t) \neq c(x_t)]] = \sum_{t=1}^{T} \Pr_{\hat{y}_t}[\hat{y}_t \neq c(x_t)] = \mathbb{E}[\#(\text{mistakes})]. \tag{15}$$

Plugging this into (13) yields the desired result.

4 Relation to Boosting

Now, let's look at a simplified version of Boosting (with known edge $\gamma > 0$):

```
Algorithm 5
INPUT: Weak hypothesis space $\mathcal{H}$, Training set $\mathcal{X}$, Number of iterations $T$.
  for $t = 1, ..., T$ do
      Booster chooses distribution $D_t$ over $\mathcal{X}$
      Weak learner chooses $h_t \in \mathcal{H}$ such that $\Pr_{x \sim D_t}[h_t(x) \neq c(x)] \leq \frac{1}{2} - \gamma$
OUTPUT: $H = \text{MAJORITY}(h_1, ..., h_t)$.
```

Since the distributions D_t are over the examples (not hypotheses), using the matrix M defined as in the previous analysis, construct $M' = 1 - M^\top$, so that

$$M'(x, h) = \mathbb{1}[h(x) = c(x)].$$

In fact, M and M' actually represent exactly the same game but with the roles of the row and column players reversed: transposing switches the examples with the hypotheses; negating switches min and max; and adding 1 to every entry simply translates the entries to $[0, 1]$ while having no impact on the game.

Now, run the Multiplicative Weights algorithm on the matrix M'. On each round t, MW will find a distribution P_t on the rows of M'. So, let $D_t = P_t$, get $h_t \in \mathcal{H}$ and let Q_t be concentrated on h_t. From the bound on the MW algorithm, we get:

$$\frac{1}{T}\sum_{t=1}^{T} M'(P_t, h_t) \leq \min_{x \in \mathcal{X}} \frac{1}{T}\sum_{t=1}^{T} M'(x, h_t) + \Delta_T \tag{16}$$

Note that $M'(P_t, h_t)$ is the probability of choosing a row (according to $D_t = P_t$) that is correctly classified by h_t, so $M'(P_t, h_t) \geq \frac{1}{2} + \gamma$. Combining this with the above, and rearranging terms yields:

$$\forall\ x \in \mathcal{X} :\ \frac{1}{T}\sum_{t=1}^{T} M'(x, h_t) \geq \frac{1}{2} + \gamma - \Delta_T > \frac{1}{2} \tag{17}$$

since Δ_T goes to 0 as $T \to \infty$. Notice that $\frac{1}{T}\sum_{t=1}^{T} M'(x, h_t)$ is the fraction of hypotheses that are correct on example x, so when we take the majority vote H, we immediately get that $H(x) = c(x)$. Therefore, the training error goes to 0.

Finally, the two previous analyses show that (versions of) the Weighted Majority algorithm and AdaBoost can both be viewed as special cases of a more general algorithm for playing repeated games. Furthermore, the games that are used for online learning and boosting are in fact duals of each other, in the sense that they represent the exact same game, but with the row and column players reversed.