COS 511: Theoretical Machine Learning

Lecturer: Rob Schapire
Scribe: Eugene Brevdo

Lecture #23
April 30, 2008

1 Review of Zero-Sum Games

Last time we introduced a mathematical model for two player zero-sum games. Any such game can in principle be formulated via its Game Matrix M. We saw two possible models for such games: deterministic play and randomized play. In deterministic play, players make decisions according to "pure" strategies, and in randomized play, players act according to "mixed" (random) strategies. At the end of each game we have an outcome. For pure strategy games, the outcome is a scalar. For mixed strategy games, we can calculate the expectation of the outcome (the *expected outcome*).

Deterministic Game:

- Game matrix M
- Row player (min) chooses row i (the min outcome)
- Column player (max) chooses column j (the max outcome)
- outcome $= M(i, j)$

Randomized Game:

- Game matrix M
- Row chosen according to distribution P over the rows
- Column chosen according to distribution Q over the columns
- (expected) outcome $= \sum_{i,j} P(i)M(i, j)Q(j) = P^T MQ$

where for randomized games, we use the notation $M(P, Q) = P^T MQ$ to denote the expected outcome as a function of the distributions chosen by the players.

There are many connections between game theory and machine learning, and in today's class we will try to bring together topics from both of these fields and unify them.

1.1 A Basic Inequality for Randomized Games

Last time we also introduced a basic analysis for the game of Rock-Paper-Scissors (R,P,S). In this game, both sides play simultaneously. Suppose the game is modified slightly, and instead play is sequential. Mindy (the row player) goes first, and Max (the column player) can decide what to do. In a deterministic setting this game is not very interesting. Let's consider randomized play:

- min chooses P first

- max chooses Q knowing P

Since Max knows that Mindy has chosen P and knows that for a given Q, the outcome is $M(P,Q)$, then Max will want to choose Q to maximize $M(P,Q)$, giving outcome $\max_Q M(P,Q)$. Since Mindy knows what Max will do and will know the given outcome, she chooses to minimize this outcome. Thus, the final outcome will be

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

where again, in this case Mindy has gone first. The minimizing P above is called the *min max strategy*.

Example: Consider the game where Mindy prefers rock and has chosen $P = (1/2, 1/4, 1/4)$:

$$R \text{ with probability } 1/2$$
$$P \text{ with probability } 1/4$$
$$S \text{ with probability } 1/4$$

then Max can always choose paper and $Q = (0,1,0)$. The game matrix M for this game is:

	R	P	S
R	1/2	1	0
P	0	1/2	1
S	1	0	1/2

and thus we have $M(P,Q) = 5/8$. In fact, if Mindy chooses a P which is not uniform, the payoff will always be greater than $1/2$.

Now, if Max chooses the column distribution Q first, and Mindy chooses the P to minimize the outcome, we would have an out come of

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

where the selected Q is also known as the *max min strategy*. We have discussed this in other contexts, and note that the player going second has the advantage. More specifically, it is straightforward to show

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

2 Fundamental Theorem of Zero-Sum Games

It turns out that for all zero-sum games with finite moves, the above inequality can be turned into an equality. Thus, regardless of who goes first, in a game of optimal players, the expected outcome is always the same. We denote the outcome value v, this is the Value of the game. So, $v = \min_P \max_Q M(P,Q) = \max_Q \min_P M(P,Q)$. This theorem was proved by von Neumann (who was at IAS in Princeton), and is called the von Neumann min max theorem. We will aim to prove this theorem via an online learning algorithm and arguments we have seen previously.

Theorem 2.1 (von Neumann min max theorem) *For randomized zero-sum games of two players:*

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

The theorem implies that even if Max knows Mindy's strategy, Max cannot get a better outcome than v; v is the best possible value: $\exists$ min max strategy P^* such that $\forall Q : M(P^*,Q) \leq v$. It also implies that no matter what strategy Mindy uses, the outcome is at worst v: $\exists$ max min strategy Q^* such that $\forall P : M(P,Q^*) \geq v$. P^* and Q^* are the optimal strategies, assuming that the opponent is also using an optimal strategy. Thus for a two person zero sum game against a good opponent, your best bet is to find the min max strategy and to always play it.

This is not the end of the story. There are many other issues to consider in such games, usually due to limited information about the game or the opponent. Here are some examples:

1. We don't always know M.
2. M can be very large (and computing P^* computationally difficult).
3. The strategy P^* is only the best when playing against an *optimal* opponent; we might do better against a non-optimal opponent (i.e., dumb, not mean, etc...).

   Bart Simpson, for example, always plays Rock instead of choosing the uniform distribution. Thus you can play Paper and always beat Bart, instead of only about 1/3 of the time.

As most games are played over and over again, there is an opportunity for learning either the game rules M or the opponent's strategy, or both, even without any knowledge of either at the beginning. Let's consider an online version of T iterations of the game:

$n = \#$ rows
for $t = 1, \ldots, T$

- row (learner) chooses P_t
- column (environment) chooses Q_t
- learner observes $M(i, Q_t)$ for each row i
- learner suffers loss $M(P_t, Q_t)$

and we define the total loss $= \sum_{t=1}^{T} M(P_t, Q_t)$.

Note that at each iteration, the learner is able to observe *part* of the game matrix M. Here, $M(i,Q)$ is the outcome of M given that Mindy chose row i and the distribution over the columns is Q. Similarly $M(P,j)$ is the outcome of M given that max chose column i and the distribution over the rows was P. If Q_t is concentrated on a particular column, then the learner sees that entire column. The opposite extreme is when the learner can see only a particular element of the game matrix M. Such cases are more complicated to

analyze.

Now, the learner wishes to minimize the total loss, as compared to the best loss possible, had the learner just chosen the best *fixed* strategy for the T iterations and stuck with it. We want:

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

Note that $\min_P \sum_{t=1}^{T} M(P, Q_t) \leq vT$, with equality if $Q_t = Q^*$. This happens when the opponent is an optimal opponent, leading to $P = P^*$. In general this doesn't have to be the case and we can do significantly better. Thus we can do almost as well as if we had known P ahead of time.

2.1 Multiplicative Updates

Let's consider a simple multiplicative weight update algorithm, which outputs distributions over rows, thus telling us how to play:

$$P_1(i) = \frac{1}{n} \quad \forall i$$
$$P_{t+1}(i) = \frac{P_t(i)\beta^{M(i,Q_t)}}{Z}$$

where $0 < \beta < 1$ and Z is a normalizing constant. Note that $M(i, Q_t)$ is the loss for expert i. We can prove the following bound for this algorithm (which is a direct generalization of the weighted majority vote algorithm we studied earlier):

Theorem 2.2 (Performance of Multiplicative Weight Updates) *Using the algorithm above, we have*

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

where a_β and c_β are functions of β.

We are not going to prove this theorem, but we have seen similar potential-based arguments of this sort of proof before.

Corollary 2.3 *It is possible to set β so that, when dividing both sides by T to calculate the average per-round loss:*

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

where $\Delta_t = O\left(\sqrt{\frac{\ln n}{T}}\right)$.

Note that the small error term falls off to 0 as $T \to \infty$, for n fixed. Thus the average per-round loss approaches the best possible.

In the theorem above, Q_t depends on P_t, but we keep this same Q_t on the right hand side. Thus, we don't get everything we wanted; however it is sufficient to prove the min max theorem. To prove the theorem, first assume Q_t is chosen adversarially to maximize the loss:

$$\begin{aligned} Q_t &= \arg\max_Q M(P_t, Q) \\ &= \arg\max_j M(P_t, j) \end{aligned}$$

and define

$$\begin{aligned} \bar{P} &= \frac{1}{T}\sum_t P_t \\ \bar{Q} &= \frac{1}{T}\sum_t Q_t. \end{aligned}$$

Note that the average of several distributions is also a distribution. Now, we know that $\max\min M(P,Q) \leq \min\max M(P,Q)$. We wish to show that $\max\min M(P,Q) \geq \min\max M(P,Q)$. Consider the following string of inequalities:

$$\begin{aligned} \min_P \max_Q P^T M Q &\leq \max_Q \bar{P}^T M Q && (1) \\ &= \max_Q \frac{1}{T}\sum_t P_t^T M Q && (2) \\ &\leq \frac{1}{T}\sum_t \max_Q P_t^T M Q && (3) \\ &= \frac{1}{T}\sum_t P_t^T M Q_t && (4) \\ &\leq \min_P \frac{1}{T}\sum_{t=1}^{T} P^T M Q_t + \Delta_T && (5) \\ &= \min_P P^T M \bar{Q} + \Delta_T && (6) \\ &\leq \max_Q \min_P P^T M Q + \Delta_T. && (7) \\ & && (8) \end{aligned}$$

Here, (1) is by definition of the minimum, (2) is by definition of $\bar{P}$, (3) is by convexity: max avg $\leq$ avg max, (4) is by definition of the adversary, (5) is by the bound of the online algorithm, (6) is by definition of $\bar{Q}$, (7) is because we maximize over Q. Thus we have that the $\min\max \leq \max\min + \Delta_T$, with $\Delta_T \to 0$ as $T \to \infty$. This proves the von Neumann min max theorem.

We also learn something useful from the algorithm. If we skip the first inequality, then we see that:

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

where $v = \max_Q \min_P M(P,Q)$. Thus, taking the average of the P_t's computed during the algorithm, we get a distribution that is within Δ_T of optimal. If $\Delta_T = 0$, then $\bar{P}$ is optimal.

Thus we get as close to the maximum as we wish by running the game for more steps. $\bar{P}$ is called the approximate min max strategy. Similarly we can skip the last inequality and show that $\bar{Q}$ is the approximate max min strategy.

As an aside, games that are not zero-sum are much harder to analyze. In general, one wishes to find a strategy for finding the Nash equilibrium. The strategies we are considering are related to a class of strategies that find another equilibrium point called the correlated equilibrium.

3 Relation to Online Learning

Let us consider the following learning problem:
for $t = 1, \ldots, T$

- observe $x_t \in \mathcal{X}$
- predict $\hat{y}_t \in \{0, 1\}$
- observe the true label $c(x_t)$ (we made a mistake if $\hat{y}_t \neq c(x_t)$)

Suppose that we associate experts with each hypothesis $h \in \mathcal{H}$, and we wish to do almost as well as the best hypothesis. We consider $\mathcal{X}$ and $\mathcal{H}$ to be finite sets. And we would like, similarly to what we've seen in the past:

$$\#\text{ mistakes} \leq \#\text{ mistakes of best } h + \text{small}.$$

For the given learning problem, we can set up an equivalent game by formulating a game matrix M:

$$M = \begin{array}{c|ccc} & x_1 & x_2 & \cdots \\ \hline h_1 & & & \\ h_2 & & & \\ \vdots & & & \end{array}$$

where $\{x_i\}$, $i = 1, \ldots, |\mathcal{X}|$ are simply indices into the set of possible instances (not the observations that we see in any given round of learning). Similarly we have $\{h_j\}$, $j = 1, \ldots, |\mathcal{H}|$. We fill in this *mistake matrix*:

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

and apply the game playing algorithm to the game matrix. Given a particular observation x_t, the algorithm chooses a distribution P_t from the rows. This is a distribution on the hypotheses. The algorithm chooses a random hypothesis according to this distribution and applies it to the example given. For the analysis, we have:

$$\underbrace{\sum_t M(P_t, X_t)}_{E[\#\text{ mistakes}]} \leq \min_P \sum_t M(P, x_t) + \text{small} = \underbrace{\min_h \sum_t M(h, x_t)}_{\#\text{ mistakes of best h}} + \text{small}$$

which, after all the definitions are properly plugged in, is the same bound as we achieved in the online learning model.

4 Relation to Boosting

We can consider the basic Boosting learning problem as a game between two players: the Boosting algorithm and the weak learning algorithm.

$$\mathcal{H} = \{\text{weak hypotheses}\}$$
$$\mathcal{X} = \text{training examples}$$

for $t = 1, \ldots, T$

- The booster chooses a distribution D_t over $\mathcal{X}$
- The WL algorithm selects $h_t \in \mathcal{H}$ such that:

$$Pr_{x \sim D_t}[h_t(x) \neq c(x)] \leq \frac{1}{2} - \gamma.$$

We cannot turn this procedure into a game readily using the M we derived in the last section, because the hypothesis space is no longer over the examples. However, by flipping the game board, converting the row player into the min player, and renormalizing so that the resulting M values are in $[0, 1]$, we can apply the game playing algorithm to the Boosting problem. Thus, we use the game matrix

$$M' = 1 - M^T$$

Now

$$M'(x, h) = \begin{cases} 1 & \text{if } h(x) = c(x) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Our reduction becomes, for boosting, to let $D_t = P_t$ and Q_t is the distribution concentrated on the h_t given to us: all the weight is on one particular column. By applying the game playing algorithm, we have

$$M'(P_t, Q_t) = M'(P_t, h_t) = Pr_{x \sim D_t}[h_t(x) = c(x)] \geq \frac{1}{2} + \gamma. \tag{10}$$

And in three steps we have the guarantees of a Boosting algorithm:

1. Using the bound given in corollary (2.3):

$$\frac{1}{2} + \gamma \leq \frac{1}{T} \sum_t M'(P_t, h_t) \leq \min_P \frac{1}{T} \sum_{t=1}^{T} M'(x, h_t) + \Delta_T \tag{11}$$

   where the first inequality is due to (10) applied to the individual iterations.

2. Since the second inequality in (11) applies to the minimum over all P (all rows), it applies to all rows:

$$\forall x : \frac{1}{T} \sum_t M'(x, h_t) \geq \frac{1}{2} + \gamma - \Delta_T > 1/2$$

   where the second inequality is true provided T is large enough (so that $\Delta_T < \gamma$). Note that $\frac{1}{T} \sum_t M'(x, h_t)$ is the fraction of weak hypotheses that correctly classify x.

3. Because for any x, the fraction of hypotheses that correctly classify x approaches a value strictly greater than $1/2$ we have that:

$$MAJ(h_1(x), \cdots, h_T(x)) = c(x)$$

on all x.

Thus the game formulation $M'(x, h)$ as given in (9) solves the problem of choosing sample weights in the simple, non-adaptive case of Boosting.