**Algorithmic Excursions: Topics in Computer Science II** **Spring 2016**

# Week 11: Zero-Sum Games, Learning Theory and Boosting

*Lecturer: Kasturi Varadarajan* *Scribe: Sikder Huq*

## 11.1 Solving zero-sum games approximately

We show how the general algorithm presented in the last lecture can be used to approximately solve zero-sum games. Let $A$ be a payoff matrix of a finite 2-player zero-sum game, with $n$ rows. When the row player plays strategy $i$ and the column player plays strategy $j$, then the payoff to the column player is $A(i,j)$. We assume $A(i,j) \in [0,1]$. If the row player chooses strategy $i$ from a distribution $\mathbf{p}$ over the rows, then the expected payoff to the column player for choosing a strategy $j$ is

$$A(\mathbf{p}, j) = \mathop{\mathrm{E}}_{i \sim \mathbf{p}}[A(i,j)]$$

Thus, the best response for the row player is the strategy $i$ which minimizes this payoff. John von Neumann's min-max theorem says that if each of the players chooses a distribution over their strategies to optimize their worst case payoff, then the value they obtain is:

$$\lambda^* := \min_P \max_j A(\mathbf{p}, j)$$

Our goal is to find a distribution $\widetilde{\mathbf{p}}$ such that

$$\max_j A(\widetilde{\mathbf{p}}, j) \leq \lambda^* + \epsilon$$

In each round, given a distribution $\mathbf{p}^{(t)}$ on the rows, we choose $j^{(t)}$ to be the best response strategy to $\mathbf{p}^{(t)}$ for the column player. This follows:

$$j^{(t)} = \text{avg } \max_j A(\mathbf{p}, j)$$

Thus, the loss vector $\mathbf{m}^{(t)} = \text{column}_j^{(t)}$.

Generate $\sum_{t=1}^{T} \mathbf{m}^{(t)}\mathbf{p}^{(t)} \leq \sum_{t=1}^{T} \mathbf{m}^{(t)}\mathbf{p} + \eta \sum_{t=1}^{T} |\mathbf{m}^{(t)}|\mathbf{p} + \frac{\ln n}{\eta}$, for any $\mathbf{p}$.

Now since $\mathbf{m}^{(t)}\mathbf{p}^{(t)} = A(\mathbf{p}^{(t)}, j^{(t)})$ and all $A(i,j) \in [0,1]$. We get,

$$\lambda^* T \leq \sum_{t=1}^{T} A(\mathbf{p}^{(t)}, j^{(t)}) \leq A(\mathbf{p}, j^{(t)}) + \eta \sum_{t=1}^{T} 1 + \frac{\ln n}{\eta}$$

Dividing by $T$,

$$\lambda^* \leq \frac{1}{T} \sum_{t=1}^{T} A(\mathbf{p}^{(t)}, j^{(t)}) \leq \frac{1}{T} A(\mathbf{p}, j^{(t)}) + \eta + \frac{\ln n}{\eta T}$$

By definition, $\lambda^* = A(\mathbf{p}^{(t)}, j^{(t)})$. We set $\mathbf{p}$ to be the best strategy of the row player, so $A(\mathbf{p}, j^{(t)}) \leq \lambda^*$, for any $j$. Therefore we get,

$$\frac{1}{T}\sum_{t=1}^{T} A(\mathbf{p}^{(t)}, j^{(t)}) \leq \lambda^* + \eta + \frac{\ln n}{\eta T}$$

Our goal is to come up with a probability distribution that is almost as good as $\lambda^*$. Let $\widetilde{t}$ minimizes $A(\mathbf{p}^{(t)}, j^{(t)})$. Therefore,

$$A(\mathbf{p}^{(\widetilde{t})}, j^{(\widetilde{t})}) \leq \lambda^* + \eta + \frac{\ln n}{\eta T}$$

Pick $\eta = \frac{\epsilon}{2}$ and $T = \left\lceil \frac{\ln n}{(\epsilon/2)^2} \right\rceil$; we get

$$A(\mathbf{p}^{(\widetilde{t})}, j^{(\widetilde{t})}) \leq \lambda^* + \epsilon$$

Therefore,

$$\lambda^* \leq \frac{1}{T}\sum_{t=1}^{T} A(\mathbf{p}, j^{(t)}) + \epsilon, \text{ for any } \mathbf{p}$$

Let $\mathbf{p}$ corresponds to playing row $i$ with probability 1.

$$\lambda^* - \epsilon \leq \frac{1}{T}\sum_{t=1}^{T} A(i, j^{(t)})$$

Let $\mathbf{q}^*$ be the probability distribution on columns that assigns to column $j$ the probability

$$\frac{|\{t : j^{(t)} = j\}|}{T}$$

So,

$$\lambda^* - \epsilon \leq \frac{1}{T}\sum_{t=1}^{T} A(i, j^{(t)}) \leq A(i, \mathbf{q}^*)$$

for any row $i$.

For example, consider the following table:

| $t$ | $j^{(t)}$ |
|---|---|
| 1 | $\alpha$ |
| 2 | $\beta$ |
| 3 | $\alpha$ |
| 4 | $\gamma$ |
| 5 | $\alpha$ |

The probability distribution for this example is $\frac{1}{5}(A(i,\alpha), A(i,\beta), A(i,\alpha), A(i,\gamma), A(i,\alpha))$. We get, $\mathbf{q}^*\alpha = 3/5$, $\mathbf{q}^*\beta = 1/5$ and $\mathbf{q}^*\gamma = 1/5$.

Recall,

$$\lambda^* = \min_{\mathbf{p}} \max_j A(\mathbf{p}, j)$$

$$\geq \max_{\mathbf{q}} \min_i A(i, \mathbf{q})$$

$$\geq \min_i A(i, \mathbf{q}^*)$$

$$\geq \lambda^* - \epsilon$$

Since this is true for any $\epsilon$

$$\max_{\mathbf{q}} \min_i A(i, \mathbf{q}) = \lambda^*$$

Which is the min-max theorem.

## 11.2 Learning Theory and Boosting

Let $\mathcal{X}$ be some domain and suppose we are trying to learn a concept class $\mathcal{C}$ where each element of $\mathcal{C}$ is a function $c : \mathcal{X} = \{0, 1\}$. There is a distribution $\mathcal{D}$ on the domain $\mathcal{X}$. We try to learn the unknown concept class $\mathcal{C}$. For example, $(x_1, c(x_1)), (x_2, c(x_2))$ where $x_1, x_2$ are i.i.d. according to $\mathcal{D}$. Learning algorithm needs to output a hypothesis $h : \mathcal{X} \to \{0, 1\}$. The *error* of the hypothesis is defined to be $\mathbf{E}_{\mathcal{X}\sim\mathcal{D}}[|h(x) - c(x)|] \leq \epsilon$.

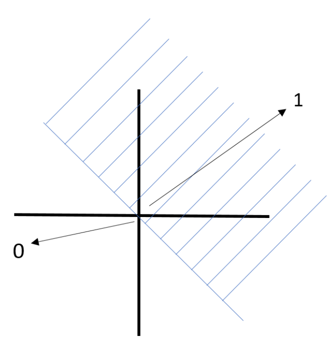


Figure 11.1: Mapping of a concept

**Definition 11.1** ***Weak learner.*** *There exists* $\mathcal{Y} > 0$ *such that for any distribution* $\mathcal{D}$ *on* $\mathcal{X}$*, learner draws samples* $(x_1, c(x_1)), (x_2, c(x_2), \cdots)$ *and outputs hypothesis* $h : \mathcal{X} \to \{0, 1\}$ *such that with probability at least* $1 - \delta$

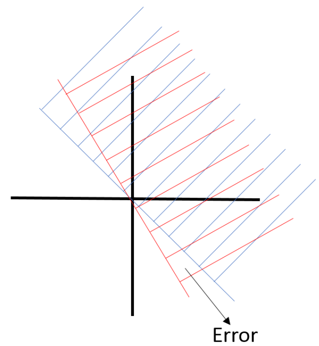


Figure 11.2: Error of a hypothesis

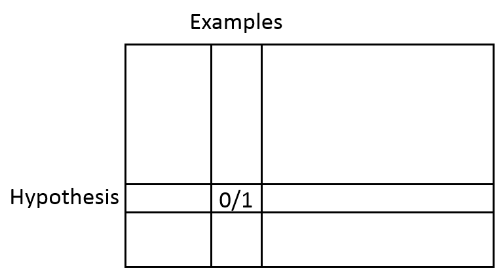


Figure 11.3: Examples vs. hypothesis

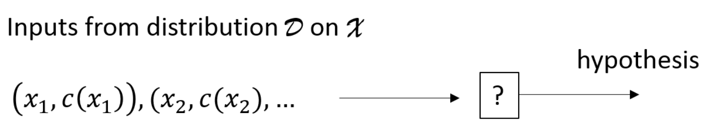


Figure 11.4: Computing hypothesis from stream

$$\underset{\mathcal{X}\sim\mathcal{D}}{\text{Prob}}[h(x) = c(x)] \geq \frac{1}{2} + \lambda$$

$$\underset{\mathcal{X}\sim\mathcal{D}}{\text{E}}[h(x) = c(x)] \leq \frac{1}{2} - \lambda$$

**Definition 11.2** ***Strong learner.*** *For any $\epsilon > 0$, for any distribution $\mathcal{D}$ on $\mathcal{X}$, learner draws samples $(x_1, c(x_1)), (x_2, c(x_2), \cdots)$ and outputs hypothesis $h : \mathcal{X} \to \{0, 1\}$ such that with probability at least $1 - \delta$*

$$\underset{\mathcal{X}\sim\mathcal{D}}{\text{E}}[h(x) - c(x)] \leq \epsilon$$

A weak learner with an assumption that class $\mathcal{H}$ of hypothesis containing weak learner's output has finite

VC implies strong learner.

$S$ of $N$ example: ? (figure 11.4) $\rightarrow$ compute a hypothesis that is incorrect on at most $\epsilon$ fraction of $S$. $S$ is an $\epsilon$-approximation if the error on hypothesis on $\mathcal{D} \leq 2\epsilon$.

| | $X_1$ | $X_2$ | $X_3$ | $\cdots$ | $X_N$ |
|---|---|---|---|---|---|
| $h^{(t)} \rightarrow$ | 0 | 0 | 1 | $\cdots$ | 1 |
| $C \rightarrow$ | 1 | 0 | 1 | $\cdots$ | 0 |
| $\mathbf{m}^{(t)} \rightarrow$ | 0 | 1 | 1 | $\cdots$ | 0 |

At each time $t$, we have a distribution $\mathbf{p}^{(t)}$ on sample. We run weak learners with input $\mathbf{p}^{(t)}$ and obtain a hypothesis $h^{(t)}$ that is good on fraction $\geq \frac{1}{2} + \lambda$.

Loss vector $\mathbf{m}_x^{(t)} = 1 - |h_{(x)}^{(t)} - c(x)|$

Number of steps $T = \frac{2}{\lambda^2} \ln \frac{1}{\epsilon}$, which is independent of $N$. Then we take the majority of hypotheses.

# Reference

[Arora et al., 2012] Arora, S., Hazan, E., and Kale, S. (2012). The multiplicative weights update method: a meta-algorithm and applications. Theory of Computing, 8(6):121-164.