Advanced Topics in ML and AGT October 15, 2018

Lecture 1

Lecturer: Yishay Mansour *Scribes: Uri Shaham, Omer Koren, Eitan-Hai Mashiah*[1]

Course requirements

- Scribe notes (30% of the final grade).
- Problem sets (30% of the final grade).
- Final project (40% of the final grade).

1 Introduction

The course will focus on the online learning model. The motivation for the online model comes from the need that often raises in the real world, to make decision (prediction or classification) under uncertainty conditions (on the future).

The general setting of the model is as followed:

At time step t:

1. We receive some input x_t.
2. We choose a response p_t (action, prediction or classification).
3. We receive a feedback y_t.
4. We suffer a loss $l(p_t, y_t)$.

Our goal will be to minimize the accumulated loss:

$$\sum_{t=1}^{T} l(p_t, y_t)$$

The course will focus on problems around this basic structure.

[1] *Based on the scribe of Alon Resler, Eliran Shabbat, Ran Levy, 2017.*

2 Course Outline

Main topics in the course will include:

Part 1: Adversarial model - no assumptions about input generation process.

- Convex learning optimization, regret minimization:
  1. At time t:
     (a) We choose some data point p_t.
     (b) We observe a convex loss function $f_t(\cdot)$.
     (c) We suffer a loss $f_t(p_t)$.
  2. The total loss is $\sum_{t=1}^{T} f_t(p_t)$.

  We will compare the online loss to the minimal loss relative to a single static point and define the regret as the difference between these two values:

$$Regret = \underbrace{\sum_{t=1}^{T} f_t(p_t)}_{\text{online loss}} - \underbrace{\min_{x^* \in \mathcal{X}} \sum_{t=1}^{T} f_t(x^*)}_{\text{benchmark}}$$

  Our goal is to show that the *Regret* is sub-linear in T, i.e:

$$\lim_{T\to\infty} \frac{Regret}{T} = 0$$

- Learning context:
  1. Classification problems (recall, Perceptron and Winnow):
     Goal: Find a hypothesis that performs as close as possible to the best one in the hypothesis class $\mathcal{H}$.
  2. Partial information:
     We can only observe the value of $f_t(p_t)$, but not the function $f_t(\cdot)$, which means we can only observe the loss caused by actions we actually played.
     Goal: Minimize the *Regret*.

Part 2: Stochastic models

- Each action has an unknown gain distribution.
- There is a best action - the one that maximizes the gain expectation.
- The *Regret* is the difference between the average gain of the online algorithm and the gain of the best action.

- Models:
  - Basic model: Gain for each action
  - Gain according to the context: Gain for action is influenced by both the action itself and the context.
  - Linear model: Each action has a corresponding vector which is multiplied by the context vector.
  - Markov Models

Part 3: Game Theory and Equilibrium

- Zero-sum game
- Correlated equilibrium
- Inferior actions
- Calibration

3 Online Learning Model

The Model

We have discrete time steps $\{1, ..., T\}$. At time t:

1. The algorithm receives an unlabeled example $x_t \in \mathcal{X}$.
2. The algorithm chooses $p_t \in \mathcal{D}$, a *prediction*, *classification* or *a distribution over labels*.
3. The algorithm receives a *feedback* $y_t \in \mathcal{Y}$.
4. The algorithm suffers a loss $l(p_t, y_t)$.

Note that the online model is adversarial, i.e., the algorithm must handle any series of inputs. Namely, we do not assume stochastic settings regarding how the inputs are generated. In addition, we usually take $\mathcal{D} = \mathcal{Y}$ or $\mathcal{D} = \Delta(\mathcal{Y})$ (were $\Delta(\mathcal{Y})$ is the set of all distributions over the final set $\mathcal{Y}$).

This setting can model many learning problems. For example:

1. **Classification problems:** Here $\mathcal{X} = \mathbb{R}^d$ and $\mathcal{Y} = \mathcal{D} = \{0, 1\}$. We can use zero-one loss function:
$$l(p_t, y_t) = |p_t - y_t| = \begin{cases} 1 & p_t \neq y_t \\ 0 & p_t = y_t \end{cases}$$

2. **Weather Prediction:** As before $\mathcal{X} = \mathbb{R}^d$ (think of $x \in \mathcal{X}$ as a vector of today's weather features like wind, pressure etc.) and our goal is to forecast if it will rain tomorrow, i.e. $\mathcal{Y} = \{0, 1\}$. We will want to allow the forecaster to express its uncertainty by taking $p_t \in [0, 1] = \mathcal{D}$ (i.e. the probability for rain tomorrow). Here we take the same loss function
$$l(p_t, y_t) = |p_t - y_t|$$

Discussion

Our goal in the online setting will be to handle all pairs (x_t, y_t) that the adversary gives us. In the model as we described it we have no chance to succeed. For instance, in the classification example above, the adversary can always choose $y_t \neq p_t$ and therefore

$$\sum_{t=1}^{T} l_{01}(p_t, y_t) = T$$

To overcome this we have two options:

1. **Restriction on the adversary:** We will assume that there exists a hypotheses class $\mathcal{H}$ and every hypothesis $h \in \mathcal{H}$ is a function $f : \mathcal{X} \to \mathcal{Y}$. The adversary will choose $h^* \in \mathcal{H}$ and then for every x_t the feedback will be $y_t = h^*(x_t)$. We will try to minimize $M_A(\mathcal{H})$ - the number of mistakes algorithm A makes with respect to the hypothesis class $\mathcal{H}$. This setting is called the ***Realizable Setting***, since there exists a hypothesis h^* consistent with all labels.

2. **Different success criteria:** We do not make any assumptions on (x_t, y_t) but we will compare our loss to the loss of the best hypothesis from given hypotheses class $\mathcal{H}$. The measure we use, the *Regret*, is defined by

$$Regret_T(\{(x_t, y_t)\}_{t=1}^{T}) = \sum_{t=1}^{T} l(p_t, y_t) - \sum_{t=1}^{T} l(h^*(x_t), y_t)$$

$$h^* = \operatorname*{argmin}_{h \in \mathcal{H}} \sum_{t=1}^{T} l(h(x_t), y_t)$$

   This setting is called ***The Regret Minimization Setting***.

Example for Regret Problems

1. Online linear regression:
   - Let $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \mathcal{D} = \mathbb{R}$.
   - Hypothesis defined by vectors $w \in \mathbb{R}^d$:

$$p = h_w(x) = \sum_{i=1}^{d} w_i x_i = w^T x$$

- The loss is $l(p, y) = (p - y)^2$
- The *Regret* of the algorithm is:

$$\underbrace{\sum_{t=1}^{T} (w_t^\top x_t - y_t)^2}_{\text{online loss}} - \underbrace{\min_{w \in \mathbb{R}^d} \sum_{t=1}^{T} (w^\top x_t - y_t)^2}_{\text{regression loss}}$$

2. Prediction with Expert Advise:

   - Input is a set of d experts $D = \{1, ..., d\}$.
   - The learning algorithm chooses at each time step t an expert $p_t \in D$.
   - The feedback is a vector $y_t \in \mathcal{Y} = [0, 1]^d$, where the p_t-th coordinate, $y_t[p_t]$, is the loss of expert p_t at time step t.
   - We can think of each expert as an hypothesis, thus we have the hypothesis class $\mathcal{H} = \{h_1, ..., h_d\}$. Therefore, the *Regret* will be:

$$Regret = \sum_{t=1}^{T} l(p_t, y_t) - \min_{h_i \in \mathcal{H}} \sum_{t=1}^{T} y_t[i]$$

4 Realizable Setting

In a realizable setting, there exists a hypothesis $h^\star \in \mathcal{H}$, where $\mathcal{H}$ is the hypotheses class, such that $y_t = h^\star(x_t)$ at each time t. The goal is to minimize the number of mistakes the algorithm makes.

Definition 1 (Mistake bound) *Algorithm $\mathcal{A}$ has a mistake bound M if for any hypothesis $h^\star \in \mathcal{H}$ and for any sequence of examples $x_1, x_2, ..., x_T$, the total number of mistakes made by $\mathcal{A}$ is bounded by M.*

We assume that the class $\mathcal{H}$ is finite, and express the mistake bound M as a function of $|\mathcal{H}|$.

4.1 Consistent algorithm

Let $\mathcal{H}$ be a finite hypothesis class. We start with the simple *Consistent* algorithm that at time t chooses a hypothesis which is consistent with all the examples seen so far.

Theorem 2 *For any finite hypothesis class $\mathcal{H}$, the consistent algorithm makes at most $|\mathcal{H}| - 1$ mistakes.*

Algorithm 1 Consistent algorithm

```
1: procedure CON($\mathcal{H}$):
2:     $V_1 \leftarrow \mathcal{H}$
3:     for $t = 1, ..., T$ do
4:         Observe $x_t$
5:         Choose arbitrary $h_t \in V_t$
6:         Predict $p_t = h_t(x_t)$
7:         Observe $y_t = h^\star(x_t)$
8:         Update $V_{t+1} \leftarrow \{h \in V_t : h(x_t) = y_t\}$
```

Proof:

- *No Regression* - Any hypothesis which is not consistent at time t, is not consistent at time $t+1$. Thus, $V_{t+1} \subseteq V_t$.
- *Termination* - Since $\mathcal{H}$ is realizable, i.e. $h^\star \in \mathcal{H} = V_1$ we get that $\forall t.\ h^\star \in V_t$, thus at any time t: $|V_t| \geq 1$.
- *Progress* - If the algorithm mistakes at time t, then $h_t \notin V_{t+1}$. Thus, $|V_{t+1}| \leq |V_t| - 1$.

We start with $V_1 = \mathcal{H}$. At time T, if M is the number of mistakes, then, $1 \leq |V_T| \leq |\mathcal{H}| - M$. Thus $M \leq |\mathcal{H}| - 1$ ■

The problem with the CON algorithm is that $|\mathcal{H}|$ can be very big. Note that the number boolean function on $\{0,1\}^n$ is 2^{2^n}.

4.2 Halving algorithm

The goal of the halving algorithm is to reduce the potentially large number of mistakes made by the consistent algorithm by choosing a good hypothesis at each time step (rather than an arbitrary consistent one). The prediction at time t that is used by the halving algorithm is the prediction made by the majority of the hypothesis in V_t. That is

$$p_t = \underset{b}{argmax} |\{h \in V_t : h(x_t) = b\}|$$

Theorem 3 *For any finite hypothesis class $\mathcal{H}$, the halving algorithm makes at most $\lfloor log_2 |\mathcal{H}| \rfloor$ mistakes.*

Proof: The choice of hypothesis made by the halving algorithm, is a valid choice of the consistent algorithm.

- *No Regression* - As before.
- *Termination* - As Before.
- *Progress* - A mistake of the algorithm means that more than half of the hypotheses were wrong. Thus, if M is the number of mistakes, we get that:

$$1 \leq |V_T| \leq 2^{-M}|V_1| = 2^{-M}|\mathcal{H}| \Rightarrow 0 \leq log_2(2^{-M}|\mathcal{H}|) = -M + log_2|\mathcal{H}|$$
$$\Rightarrow M \leq log_2|\mathcal{H}| \Rightarrow M \leq \lfloor log_2|\mathcal{H}| \rfloor$$

■

Theorem 4 (Lower bound) *There is no algorithm that achieves a mistake bound $M < \frac{1}{2}log_2|\mathcal{H}|$.*

Proof: To show a lower bound we need to show that for any algorithm $\mathcal{A}$, there exists a class of hypotheses $\mathcal{H}$, a hypothesis $h^\star \in \mathcal{H}$ and an input sequences $x_1, x_2, ..., x_T$, such that for any sequence of predictions $p_1, p_2, ..., p_T$, $M \geq \frac{1}{2}log_2|\mathcal{H}|$.

We will show one class of hypotheses $\mathcal{H}$ and a random choice of $h^\star$ such that $E[M] = \frac{1}{2}log_2|\mathcal{H}|$.

We will set $x_1, x_2, ..., x_T$, when $T = log_2|\mathcal{H}|$ and $\mathcal{H} = \{0,1\}^T$. Therefore $|\mathcal{H}| = 2^T$, and there exists a hypothesis in $\mathcal{H}$ for every classification sequence $y_1, ..., y_T$. Each observation y_t is chosen randomly from $\{0,1\}$.

Note that $\mathcal{H}$ contains all possible observation vectors, so we guarantee that $\{y_i\}_{i=1}^T$ is a valid hypothesis. Now, since at any time t, the observation is chosen randomly from $\{0,1\}$, the probability of mistake at any time t is 1/2. Thus, if we define Z_t as the random variable that receives 1 if the algorithm makes an error at time t and 0 otherwise, then:

$$E[M] = E\left[\sum_{t=1}^{T} Z_t\right] = \sum_{t=1}^{T} E[Z] = T/2 = \frac{1}{2}log_2|\mathcal{H}|$$

Thus, there is at least one choice of observations that will make at least $E[M] = T/2 = \frac{1}{2}log_2|\mathcal{H}|$ mistakes. ■

Note that in the proof we chose the observations at random. Actually, We can improve the lower bound by choosing a specific vector of observations, assuming that the algorithm is deterministic.

Theorem 5 (Lower bound) *There is no deterministic algorithm that achieves a mistake bound $M < \lfloor log_2|\mathcal{H}| \rfloor$.*

Proof: We will use the same $\mathcal{H}$ as before. For each p_t that the algorithm chooses, we will assign y_t such that $y_t \neq p_t$. This choice of observations vector is valid because $\mathcal{H}$ contains all the possible vectors. On that case, the algorithm will make $T = log_2|\mathcal{H}|$ errors. ■

5 Regret Minimization

We assume an adversarial online model with finite hypotheses class $\mathcal{H}$. At each time step t, the algorithm chooses a hypothesis $h_t \in \mathcal{H}$ (or a distribution q^t over $\mathcal{H}$). After that, the adversary selects a loss vector $l^t \in [0,1]^{|\mathcal{H}|}$ where $l_h^t \in [0,1]$ is the loss of hypothesis $h \in \mathcal{H}$ at time t. We define the loss of a distribution q^t at time t by

$$\sum_{h \in \mathcal{H}} q_h^t l_h^t$$

Note, if the algorithm chooses a hypothesis $h_t \in \mathcal{H}$, then the above loss is actually $l_{h_t}^t$. We define $L_h^T = \sum_{t=1}^T l_h^t$ as the loss of choosing the same hypothesis h every time. The goal for the external regret setting is to design an online algorithm that will be able to preform as close to the performance of the best hypothesis as possible. Namely, to have a loss close to

$$L_{best}^T = \min_{h \in \mathcal{H}} L_h^T$$

Our algorithmic model will allow random decision making by maintaining weights over hypotheses. Formally, at time t:

1. The algorithm will have a weight $w_h^t \geq 0$ for each hypothesis $h \in \mathcal{H}$.
2. The algorithm will set a distribution q^t over the hypotheses according to those weights by $q_h^t = \frac{w_h^t}{W^t}$ where $W^t = \sum_{h \in \mathcal{H}} w_h^t$. Then, draw a hypothesis according to that distribution.
3. The algorithm will observe a loss vector l^t (the losses of all hypotheses in $\mathcal{H}$).

We will define the loss of an algorithm A at time step t by

$$l_A^t = \sum_{h \in \mathcal{H}} \frac{w_h^t}{W^t} l_h^t = \sum_{h \in \mathcal{H}} q_h^t l_h^t$$

and the total loss by

$$L_A^T = \sum_{t=1}^T l_A^t$$

Note that the loss is defined as expected loss with respect to the distribution defined by the weights of each step.

Now, using our notations we can define the *Regret*:

$$Regret = L_A^T - L_{best}^T$$

Our goal will be to minimize the *Regret*.

Algorithm 2 Greedy algorithm

```
1: procedure GREEDY($\mathcal{H}$)
2:     for $t = 1, ..., T$ do
3:         $h_t \leftarrow \text{argmin}_{h\in\mathcal{H}} L_h^{t-1}$
```

5.1 Deterministic Greedy Algorithm

Our first attempt to develop a regret minimization algorithm will be to consider the greedy approach. At each time step t, the Greedy Algorithm simply selects a hypothesis with minimal loss seen so far (if there is more than one it just picks the one with the minimal index).

Theorem 6 *The Greedy algorithm, for any sequence of losses, has*

$$L_{Greedy}^T \le |H| L_{best}^T + |H| - 1$$

Proof: We split the run into time blocks such that the value of L_{best}^t is constant. Define the block B_k as the set of the times t in which $L_{best}^t = k$. We next concentrate at block B_k. At the beginning of the block, there are at most $|H|$ hypotheses with $L_h^t = k$. At each mistake of *Greedy*, the number of hypotheses with $L_h^t = k$ is reduced by at least 1. This can occur at most $|H|$ times before L_{best}^t increases by 1.

Therefore, the greedy algorithm makes at most $|H|$ mistakes between successive increments in L_{best}^t . Thus, we have

$$L_{Greedy}^T \le \sum_{k=1}^{L_{best}^T} \underbrace{\sum_{t\in B_k} l_{h_t}^t}_{\le|\mathcal{H}|} + |\mathcal{H}| - 1 \le |\mathcal{H}| L_{best}^T + |\mathcal{H}| - 1$$

■

The above theorem shows a quite weak result since the loss is bounded only by factor of $|\mathcal{H}|$ comparing to the best action. The following theorem shows that this weakness is shared by any deterministic online algorithm.

Theorem 7 *For any deterministic algorithm $\mathcal{A}$ there exists a loss sequence for which:*

$$L_{\mathcal{A}}^T \ge |\mathcal{H}| L_{best}^T + T \mod |\mathcal{H}|$$

Proof: Set a deterministic online algorithm $\mathcal{A}$ and let h_t be the hypothesis at time t. The adversary can force an error at each time step. At time t, $l_{h_t}^t = 1$ and for any $h' \ne h_t$ we have $l_{h'}^t = 0$. This ensures that $\mathcal{A}$ incurs loss of 1 at each time step, so $L_{\mathcal{A}}^T = T$.

Since there are $|\mathcal{H}|$ different hypotheses, there is a hypothesis that $\mathcal{A}$ has selected at most $\lfloor T/|\mathcal{H}|\rfloor$ times. By construction, only the hypotheses selected by $\mathcal{A}$ ever have a loss, thus $L_{best}^T \le \lfloor T/|\mathcal{H}|\rfloor$. Therefore:

$$L_{\mathcal{A}}^T = T = |\mathcal{H}| \cdot \lfloor T/|\mathcal{H}|\rfloor + T \mod |\mathcal{H}| \ge |\mathcal{H}| L_{best}^T + T \mod |\mathcal{H}|$$

■

5.2 Randomized Greedy Algorithm

Now, instead of choosing an arbitrary hypothesis among the hypotheses with minimal loss seen so far, we choose an hypothesis among this set according to a uniform distribution.

Algorithm 3 Randomized Greedy algorithm

```
1: procedure RandomizedGreedy($\mathcal{H}$)
2:     for $t = 1, ..., T$ do
3:         $V_t \leftarrow \{h : L_h^{t-1} = L_{best}^{t-1}\}$
4:         $h_t \overset{r}{\leftarrow} V_t$
```

The following theorem states that the Randomized Greedy algorithm gets a multiplicative factor of $\mathcal{O}(\ln|\mathcal{H}|)$ instead $\mathcal{O}(|\mathcal{H}|)$ achieved by the Greedy algorithm.

Theorem 8 *The Randomized Greedy algorithm, for any loss sequence, has:*

$$L_{RG}^T \leq (\ln|\mathcal{H}| + 1)L_{best}^T + \ln|\mathcal{H}|$$

Proof: As in Theorem 6, define the block B_k as the set of the times t in which $L_{best}^t = k$. Consider the block B_k. Let t be the beginning time of the block and set $m = |V^t|$.

At each mistake of *Randomized Greedy*, the set V^t shrinks by at least one. Assume w.l.o.g that it shrinks exactly by one. Otherwise, if it shrinks by r, then the probability for this mistake drops since:

$$r/m \leq 1/m + 1/(m-1) + \cdots + 1/(m-r+1)$$

Therefore, the expected number of errors in a block B_k is

$$E[L_{RG}^{B_k}] \leq \sum_{i=|\mathcal{H}|}^{1} 1/i \leq \ln|\mathcal{H}| + 1$$

This implies that the loss of *Randomized Greedy* is at most:

$$L_{RG}^T \leq (\ln|\mathcal{H}| + 1)L_{best}^T + \ln|\mathcal{H}|$$

■

5.3 Randomized Weighted Majority Algorithm

In the Randomized Weighted Majority algorithm, we choose a parameter $0 < \eta \leq 1/2$ and update the weights according to:

$$w_h^{t+1} = (1-\eta)^{l_h^t} w_h^t$$

Algorithm 4 Randomized Weighted Majority (RWM)

```
 1: procedure RWM($\mathcal{H}$)
 2: Init:
 3:     for $h \in \mathcal{H}$ do
 4:         $w_h^1 \leftarrow 1$
 5: At time step $t$:
 6:     Define probability $q_h^t = \frac{w_h^t}{W^t}$, where $W^t = \sum_{h\in\mathcal{H}} w_h^t$
 7:     Observe a loss vector $l^t$
 8:     Suffer a loss $\sum_{h\in\mathcal{H}} q_h^t l_h^t$
 9:     for $h \in \mathcal{H}$ do                                ▷ Update weights
10:         $w_h^{t+1} \leftarrow (1-\eta)^{l_h^t} w_h^t$
```

Theorem 9 *The loss of Randomized Weighted Majority on any sequence satisfies:*

$$L_{RWM}^T \le (1+\eta)L_{best}^T + \frac{\ln|\mathcal{H}|}{\eta}$$

and if we choose $\eta = \min\{1/2, \sqrt{\frac{\ln|\mathcal{H}|}{T}}\}$ *then*

$$Regret = L_{RWM}^T - L_{best}^T \le 2\sqrt{\frac{\ln|\mathcal{H}|}{T}}$$

Proof: Assume for simplicity that $l_h^t \in \{0,1\}$. Define F^t as the loss of the algorithm at time t. That is:

$$F^t = \sum_{h:l_h^t=1} \frac{w_h^t}{W^t}$$

Therefore, the loss of *RWM* is

$$L_{RWM}^T = \sum_{t=1}^T F^t$$

Recall that $W^t = \sum_{h\in\mathcal{H}} w_h^t$. We first show:

Claim 10 $W^{T+1} = W^1 \prod_{t=1}^T (1-\eta F^t) = |\mathcal{H}| \prod_{t=1}^T (1-\eta F^t)$

Proof:

$$\begin{aligned} W^{t+1} &= \sum_h w_h^{t+1} = \sum_{h:l_h^t=0} w_h^{t+1} + \sum_{h:l_h^t=1} w_h^{t+1} = \sum_{h:l_h^t=0} w_h^t + \sum_{h:l_h^t=1} (1-\eta)w_h^t \\ &= W^t - \eta \sum_{h:l_h^t=1} w_h^t = W^t - \eta F^t W^t = W^t(1-\eta F^t) \end{aligned}$$

By applying the above equation recursively we have:

$$W^{T+1} = W^1 \prod_{t=1}^T (1-\eta F^t)$$

■

Since the weights are non-negative:

$$\forall h.\ W^{T+1} \geq w_h^{T+1} = (1-\eta)^{L_h^T}$$

Thus,

$$(1-\eta)^{L_{best}^T} \leq W^{T+1}$$

By Claim 10, it follows that

$$(1-\eta)^{L_{best}^T} \leq W^{T+1} = |\mathcal{H}| \prod_{t=1}^{T} (1-\eta F^t)$$

Thus,

$$L_{best}^T \ln(1-\eta) \leq \ln(|\mathcal{H}|) + \sum_{t=1}^{T} \ln(1-\eta F^t)$$

Using the following inequality: for $z \in [0, \frac{1}{2}]$,

$$-z - z^2 \leq \ln(1-z) \leq -z$$

We get,

$$(-\eta - \eta^2) L_{best}^T \leq \ln|\mathcal{H}| - \eta \underbrace{\sum_{t=1}^{T} F^t}_{=L_{RWM}^T}$$

Therefore,

$$\eta L_{RWM}^T \leq (\eta + \eta^2) L_{best}^T + \ln|\mathcal{H}|$$

Dividing both sides by η we conclude

$$L_{RWM}^T \leq (1+\eta) L_{best}^T + \frac{\ln|\mathcal{H}|}{\eta}$$

■