Advanced Topics in ML and AGT October 23, 2017

Lecture 1

Lecturer: Yishay Mansour *Scribes: Alon Resler, Eliran Shabbat, Ran Levy*

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 we we receive some input x_t.
- Then we choose a response p_t (action, prediction or classification).
- Then we receive feedback y_t and experience some 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.

2 Course Outline

Main topics in the course will include:

- Convex learning optimization, regret minimization
  - At time t, for some input x_t and loss $f_t(x_t)$.
  - We will have online absolute loss $\sum_{t=1}^{T} f_t(x_t)$.
  - And best static point with minimal loss $\min_{x^* \in \mathcal{X}} \sum_{t=1}^{T} f_t(x_t^*)$.
  - We define the regret as the difference between the online loss and the loss of a constant minimal loss action x^*:

$$Regret = \sum_{t=1}^{T} f_t(x_t) - \min_{x^* \in \mathcal{X}} \sum_{t=1}^{T} f_t(x_t^*)$$

- Stocastic models
  - Actions with stochastic loss for each action.
  - Rule out actions.
  - Better lower bounds using Information Theory.
- Game Theory and Equilibrium
  - Zero-sum game.
  - Correlated equilibrium.
  - Multiple participants with different algorithms.
- Bandits
  - Contextual Bandit.
  - Linear Bandits.

3 Online Learning Model

The Model

We have discrete time steps. At time t the following occur:

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

Note that the online model is adversarial, i.e., we assume that the provided input is selected in the worst possible way for the learning algorithm. Namely, we do not assume stochastic settings regarding how the inputs are generated.
A naive goal in this setting is to minimize our loss for a given time steps horizon T:

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

However, this loss might be high due to the input sequence or due to bad prediction algorithm. This is why we will prefer the regret which compares the online loss to the best offline loss.

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 there will be rain tomorrow, i.e. $\mathcal{Y} = \{0, 1\}$. We will want to allow the forecaster to express its uncertainty by taking $y_t \in [0, 1] = \mathcal{D}$ (i.e. the probability for rain tomorrow). Here we take the same loss function
$$l(p_t, y_t) = |p_t - y_t|$$
Just now it "measure" our error in the probability p_t.

Discussion

Our goal in the online setting will be to handle all pairs (x_t, p_t) that the adversary gives us. In the model as we described it we have no chance to succeed. For example, in the first example above, the adversary can always choose $y_t = 1 - p_t$ and make us experience maximal loss in each round. To overcome this we have two options: to add some restriction on the adversary or to choose a different success criteria.

1. **Restriction on the adversary** - we will assume that there exist a hypotheses class $\mathcal{H}$ where 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 all 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. This setting is called the ***Realizable Setting***

2. **Different success criteria** - Here we do not make any assumption 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 define by

$$Regret = \sum_{t=1}^{T} l(p_t, y_t) - \min_{h^* \in \mathcal{H}} \sum_{t=1}^{T} l(h^*(x_t), y_t)$$

   Another way to think about the regret is the *"price"* we pay for not having all the data points in advance. This setting is called ***The Regret Minimization Setting***

Example for Regret Problems

Online linear regression:

- Let $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \mathcal{D} = \mathbb{R}$.
- Hypothesis define by vectors $w \in \mathbb{R}^d$:

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

- Each round loss is $l(p, y) = (p - y)^2$
- The regret of the algorithm is:

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

Prediction with Expert Advise:

- Here the input is a set of d experts $D = \{1, 2, ..., d\}$.
- The learning algorithm choose at each time step t an expert $p_t \in D$.

- The feedback is vector $y_t \in \mathcal{Y} = [0,1]^d$ where the i-th coordinate, $y_t[i]$ is the loss of expert i at time step t.
- We can think on each expert as an hypothesis, thus we have the hypothesis class $\mathcal{H} = \{h_1, ..., h_d\}$ and the regret is:

$$regret = \sum_{t=1}^{T} l(p_t, y_t) - \underbrace{\min_{h_i \in \mathcal{H}} \sum_{t=1}^{T} y_t[i]}_{\text{ERM solution}}$$

Online Ranking:

- At time step t the input is query and set of k relevant pages.
- The algorithm predict a permutation p_t on the relevant paper.
- The adversary output the correct permutation y_t.

4 Realizable Setting

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

Definition 1 (Mistake bound) *Algorithm $\mathcal{A}$ has 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 ever made by $\mathcal{A}$ is bounded by M.*

We shall assume that the class $\mathcal{H}$ is finite. We will want to express M with $|\mathcal{H}|$.

4.1 Consistent algorithm

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

Algorithm 1 Consistent algorithm

```
procedure CON(H):
    V_1 ← H
    for t = 1, ..., T do
        Observe x_t
        Choose arbitrary h_t ∈ V_t and predict p_t = h_t(x_t)
        Observe y_t = h*(x_t) and update V_{t+1} ← {h ∈ V_t : h(x_t) = y_t}
```

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

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 $\forall t \ \ |V_t| \geq 1$.
- *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 this 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 potentialy 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 step 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 $\lceil log_2|\mathcal{H}| \rceil$ mistakes.*

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

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

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

■

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

Proof: To show a lower bound we need to show that for any algorithm, there exist a class of hypotheses $\mathcal{H}$, an hypothesis $h^\star \in \mathcal{H}$, sequences of examples $x_1, x_2, ..., x_T$ and observations $y_1 = h^\star(x_1), y_2 = h^\star(x_2), ..., y_T = h^\star(x_T)$, such that for any sequence of

predictions $p_1, p_2, ..., p_T$, $M \geq \frac{1}{2} log_2 |\mathcal{H}|$.

Let us define $\mathcal{H}$ as the group of all possible classifications over $x_1, x_2, ..., x_T$, namely $\mathcal{H} = \{0,1\}^T$, and $T = log_2|\mathcal{H}|$. The observation y_t is chosen randomly from $\{0,1\}$.

Note that since $\mathcal{H}$ contains all possible observation vectors, $\{y_i\}_{i=1}^T$ is a valid hypothesis. Now, since at any time t, the observation is chosen randomly from $\{0,1\}$, the probability of mistake is 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 step t and 0 otherwise, then:

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

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

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.

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

Proof: Since all we need to show is the existance of a sequence of observations, we can make the algorithm perform a mistake at any time step by picking an observation different then the prediction made by the algorithm. This choise of observations vector is valid because $\mathcal{H}$ contains all the possible vectors. On that case, the algorithm will have $T = log_2|\mathcal{H}|$ mistakes. ■

5 External 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. 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 at time step 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 aim for the external regret setting is to design an online algorithm that will be able to approach the performance of the best hypothesis. Namely, to have a loss close to

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

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

- The algorithm will have weight $w_h^t \geq 0$ for each hypothesis h.
- The algorithm will set probability distribution over the hypotheses according to those weights by $p_h^t = \frac{w_h^t}{W^t}$ where $W^t = \sum_h w_h^t$ and then draw a hypothesis according to the distribution.
- The algorithm will observe a loss vector l^t (the losses of all hypotheses in $\mathcal{H}$).

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

$$l_A^t = \sum_h \frac{w_h^t}{W^t} l_h^t = \sum_h p_h^t l_h^t$$

and the algorithm loss by

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

Note the the loss is define as expected loss with respect to the distribution defined by the weights of each step. In addition this settings also capture deterministic setting because we can describe deterministic choice of h by assigning weights 0 to all $h' \neq h$ and $w_h = 1$

Now, using our notations we can define the regret:

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

Our goal will be to minimize the *Regret*.

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 then one it just pick the one with the minimal index).

Algorithm 2 Greedy algorithm

```
1: procedure GREEDY(H)
2:     h_1 ← 1
3:     for t = 1, ..., T do
4:         h_t ← arg min_{h∈H} L_h^{t-1}
```

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

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

Proof: Let B_k be the times in which $L_{best}^{t} = k$. At first, there are at most $|H|$ hypotheses with $L_{best}^{t} = k$. At each time t such that the greedy algorithm incurs a loss of 1 and L_{best}^{t} does not increase, at least one hypothesis is removed from H. This can occur at most $|H|$ times before L_{best}^{t} increases by 1.

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

$$L_{Greedy} \leq \sum_{k=1}^{L_{best}} L_{Greedy}^{B_k} + |H| - 1 \leq |H| \cdot L_{best}^{T} + |H| - 1$$

■

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 D there exists a loss sequence for which:*

$$L_{D}^{T} \geq |H| \cdot L_{best}^{T} + T mod |H|$$

Proof: Fix a deterministic D online algorithm and let h_t be the hypothesis it selects at time step t. The adversary will force an error at each time step. At time t, The loss of h_t will be 1 and the loss of any other hypothesis will be 0. This ensures that D incurs loss 1 at each time step, so $L_{D}^{T} = T$.

Since there are $|H|$ different hypotheses, there is some hypothesis that algorithm D has selected at most $\lfloor T/|H| \rfloor$ times. By construction, only the hypotheses selected by D ever have a loss, so this implies that $L_{best}^{T} \leq \lfloor T/|H| \rfloor$. Therefore:

$$L_{D}^{T} = T = |H| \cdot \lfloor T/|H| \rfloor + T mod |H| \geq |H| \cdot L_{best}^{T} + T mod |H|$$

■

Algorithm 3 Randomized Greedy algorithm

1: **procedure** RANDOMIZEDGREEDY($\mathcal{H}$)
2: $V_1 \leftarrow H$
3: **for** $t = 1, ..., T$ **do**
4: $V_t \overset{r}{\leftarrow} \{h : L_h^{t-1} = L_{best}^{t-1}\}$

5.2 Randomized Greedy Algorithm

This algorithm assigns a uniform distribution over the actions with minimum total loss seen so far.

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

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

Proof: Let B_k be the times in which $L_{best}^t = k$. Let $m = |V^T|$.

At time t, the algorithm chooses one hypothesis that has a loss of 1 and gives zero loss to all other hypotheses. If at time t the adversary gives k mistakes, which means the size of V_t shrinks by k, then the loss of the algorithm is k/m. Notice that we can upper bound it using the additive loss we get by having mistakes at k different times:

$$k/m \leq 1/m + 1/(m-1) + ... + 1/(m-k)$$

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

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

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

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

■

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:

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

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

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

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

$$L_{RWM}^T \leq L_{best}^T + 2\sqrt{\frac{\ln|\mathcal{H}|}{T}}$$

Algorithm 4 Randomized Weighted Majority (RWM)

procedure RWM($\mathcal{H}$)
Init:
  for $h \in \mathcal{H}$ **do**
    $w_h^1 \leftarrow 1$
At time step t**:**
  Define probability $p_h^t = \frac{w_h^t}{W^t}$, where $W^t = \sum_{h\in\mathcal{H}} w_h^t$
  Observe a loss vector l^t
  Suffer a loss $\sum_{h\in\mathcal{H}} p_h^t l_h^t$
  for $t = 1, ..., T$ **do** ▷ Update weights
    $w_h^t \leftarrow (1-\eta)^{l_h^{t-1}} w_h^{t-1}$

Proof: Let us assume that $l_h^t \in \{0, 1\}$ (it doesn't affect the correctness of the proof). Let us define F^t as the loss of the algorithm at time t. That is:

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

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,

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

Putting it together, since $W^1 = |\mathcal{H}|$ we have

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

Thus:

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

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

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

We get,

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

Therefore,

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

Dividing both sides by η we conclude

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

■