Advanced topics in ML and AGT **Fall Semester, 2018/19**

Lecture 11: December 24, 2018

Lecturer: Yishay Mansour Scribe: Gal Cohen, Rom Gendler and Daniel Rosen

1 Swap Regret[1]

1.1 Definition

In the previous lesson we defined *Swap Regret*:
Let A be the set of possible actions, and let $f : A \to A$ be a function that maps an action to another action, (which can be the same or a different action). Given a sequence of losses $\ell_1, \ldots, \ell_T$ and actions $a_1, \ldots, a_T$ we define the regret with respect to f:

$$SR_f = \sum_{t=1}^{T} \ell_t(a_t) - \ell_t(f(a_t))$$

The swap regret is the maximum over all functions in a set $F = \{f : A \to A\}$:

$$\text{Swap Regret} = \max_{f \in F} SR_f$$

1.2 Reduction from Swap Regret to External Regret

We will now give a black-box reduction showing how any algorithm B achieving good external regret can be used as a subroutine to achieve good swap regret as well. The high-level idea is illustrated in the following diagram:

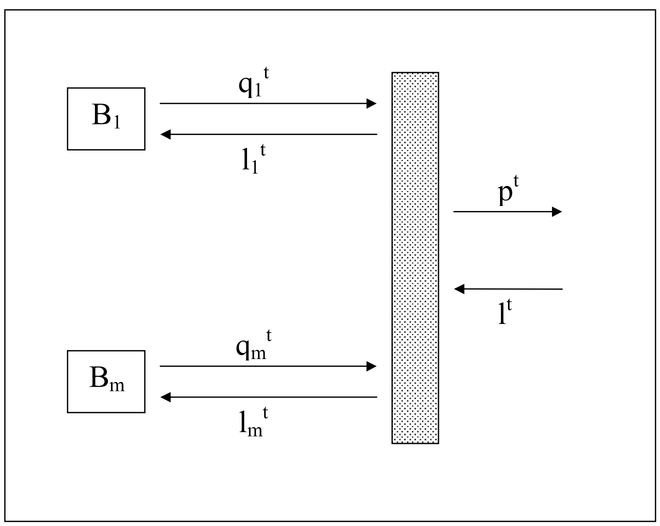

Figure 1: Reduction from Swap Regret to External Regret

Let $B_1, \ldots, B_m$ be algorithms that minimize the external regret, where m is the number of actions. At each time step t:

1. Algorithm B_i will generate a probability vector $q_i^t \in \Delta(A)$.
2. Combine these probabilities in a particular way to produce our own probability vector $p^t \in \Delta(A)$.
3. Receive a loss vector ℓ^t.

[1]Based on scribe notes by Peretz Yafin, Ben Kantor and Roy Mor from 2017/18

4. Partition it among the m algorithms, giving algorithm B_i it's part of the loss ℓ_i^t:

$$\ell_i^t = p^t(i)\ell^t.$$

Notice that $p^t(i)$ is a scalar and ℓ^t is a vector, the multiplication is done element-wise. Algorithm B_i's loss of action j at time t is therefore:

$$\ell_i^t(j) = p^t(i)\ell^t(j) \ .$$

We choose p^t in the following way. At each time step t, algorithm B_i outputs a distribution q_i^t. We define the matrix M^t to be the stochastic matrix whose rows are q_i^t:

$$M^t = \begin{pmatrix} —q_1^t— \\ \vdots \\ —q_m^t— \end{pmatrix}$$

We choose p^t to be a distribution s.t. $(p^t)^T = (p^t)^T M^t$ (we can view p^t as a stationary distribution of the Markov Process defined by M^t, it is well known that such a p^t exists). For intuition into this choice of p^t, notice that it implies we can consider action selection in two equivalent ways:

1. Directly sample an action $a \sim p^t$.
2. First sample an algorithm $i \sim p^t$ and then use algorithm B_i to select an action $a \sim q_i^t$.

From our choice of p^t it follows that:

$$\forall a : \sum_{i=1}^{m} p^t(i) q_i^t(a) = p^t(a) \ .$$

so the two sampling ways give the same distribution.

Regret Analysis The loss that algorithm B_i "sees" at time t is:

$$\ell_{B_i}^t = (\ell_i^t)^T q_i^t = (p^t(i)\ell^t)^T q_i^t = p^t(i)((\ell^t)^T q_i^t)$$

Recall that algorithm B_i minimizes the external regret and let R_i be the regret bound for B_i, then:

$$\forall j \quad L_{B_i}^T = \sum_{t=1}^{T} \ell_{B_i}^t = \sum_{t=1}^{T} p^t(i)((\ell^t)^T q_i^t) \leqslant \sum_{t=1}^{T} \ell_i^t(j) + R_i = \sum_{t=1}^{T} p^t(i)\ell^t(j) + R_i \ .$$

In words, B_i ensures low regret for a function f that maps an action i to an action j, i.e. $f(i) = j$.
If we sum the losses of the m algorithms at time t, we get:

$$\sum_{i=1}^{m} p^t(i)((q_i^t)^T \ell^t) = (p^t)^T M^t \ell^t = (p^t)^T \ell^t = \ell_{\text{on}}^t \tag{1}$$

Therefore, sum of the perceived losses of the m algorithms is equal to the actual loss.
Summing Equation 1 over all time steps, we get that for all functions $f : A \to A$:

$$L_{\text{on}}^T = \sum_{t=1}^{T} \ell_{\text{on}}^t = \sum_{t=1}^{T}\sum_{i=1}^{m} p^t(i)((q_i^t)^T \ell^t) = \sum_{i=1}^{m} L_{B_i}^T \leqslant \sum_{i=1}^{m}\left(\sum_{t=1}^{T} p^t(i)\ell^t(f(i)) + R_i\right) \leqslant \sum_{i=1}^{m} L_{i\to f(i)}^T + mR = L_{\text{on},f}^T + mR,$$

where R $= \max_i R_i$.
We already saw a regret minimization algorithm that achieves an external regret of $R \leqslant 2\sqrt{T\log m}$.
Therefore, we can derive an algorithm that achieves a swap regret bound of:

$$\text{Swap Regret } \leqslant mR \leqslant 2m\sqrt{T\log m} \ .$$

This bound can be further improved to the following:

$$\text{Swap Regret } \leqslant 2\sqrt{mT \log m}$$

In order to achieve this bound first notice that since

$$T \geqslant L_{on}^T = \sum_{i=1}^{m} L_{B_i}^T$$

and

$$R_i \leqslant \frac{\log m}{\eta} + \eta L_i$$

we get

$$SR \leqslant \sum_{i=1}^{m} R_i \leqslant \frac{m \log m}{\eta} + \eta \sum_{i=1}^{m} L_i \leqslant \frac{m \log m}{\eta} + \eta T$$

Now, for $\eta = \sqrt{\frac{m \log m}{T}}$, we indeed get

$$SR \leqslant 2\sqrt{mT \log m}$$

Giving an improved bound:

$$SR = O\Big(\sqrt{mT \log m}\Big)$$

1.3 Lower Bound

For parameter k, let $m = 2k$, $T = \alpha k$, we will show:

$$SR \geqslant \Omega(\sqrt{Tm}) = \Omega(\sqrt{\alpha} k)$$

Given algorithm with distribution p^t in time step t, let $M_t(i)$ be the number of times action i was chosen up to time t, i.e. $M_t(i) = \sum_{\tau=1}^{t-1} p^\tau(i)$. Partition the actions to pairs $(2j-1, 2j)$, The adversary assigns losses $\ell^t(j)$ as follows:

- If $M_t(2j) + M_t(2j-1) < \frac{\alpha}{2}$, that is, the j-th pair was not chosen many times, then:

$$(\ell^t(2j-1), \ell^t(2j)) = \begin{cases} (1,0) & \text{with probability } \frac{1}{2} \\ (0,1) & \text{with probability } \frac{1}{2} \end{cases}$$

- Otherwise, $\ell^t(2j-1) = \ell^t(2j) = 1$

Since $\sum_{i=1}^{m} M_t(i) = \sum_{i=1}^{m} \sum_{\tau=1}^{t-1} p^\tau(i) = \sum_{\tau=1}^{t-1} \sum_{i=1}^{m} p^\tau(i) = t-1$, the expected loss of any algorithm is at least $k\frac{\alpha}{2}\frac{1}{2} + (T - \frac{k\alpha}{2}) \cdot 1 = \frac{3}{4}k\alpha = \frac{3}{4}T$

If there is a pair with $M_T(2j) + M_T(2j-1) < \frac{\alpha}{2}$, then either action $2j-1$ or action $2j$ has a loss of at most $\frac{T}{2}$. This implies the lower bound: regret $\geqslant \frac{3}{4}T - \frac{T}{2} = \frac{T}{4}$. Otherwise, for any pair with $M_T(2j) + M_T(2j-1) \geqslant \frac{\alpha}{2}$ swapping the actions of the pair to the lower loss appropriately, gives an expected improvement of $\Omega(\sqrt{\alpha})$ for each pair. With k such pairs, the lower bound $\Omega(\sqrt{\alpha}k) = \Omega(\sqrt{mT})$ follows.

2 Wide Range Regret

2.1 Definition

We want to extend our model in two ways:

- Modification Rules: function f which given history and current action, chooses another action. In contrast to Swap Regret in which the functions were not history dependent.

- Time Selection: Function which sets which times are counted. $I(t) \in [0,1]$, can be thought as weighting of the steps which we want to be good at. Example for set of functions: the set of all of the intervals, its size is $O(T^2)$.

In time t, given history until time t for a modification rule $f : A \to A$, which is history dependent, Let M a matrix such that

$$M_f^t(i,j) = \begin{cases} 1 & j = f(i) \\ 0 & \text{otherwise} \end{cases}.$$

For an algorithm H with relation to (I, f) the regret is

$$\text{Regret}(H, I, f) = \sum_{t=1}^{T} I(t) \left(\sum_{a \in A} \underbrace{p^t(a)}_{\text{probabilities of } H} (\ell_t(a) - \ell_t(f(a))) \right)$$

$$= \sum_{t=1}^{T} I(t)(p_t^T \ell_t - (p_t^T M_f)^T \ell_t)$$

Given a set of pairs of modifications rule and time selection functions $S \subseteq I \times F$, the regret is,

$$\text{Regret}(H, S) = \max_{(I,f) \in S} \text{Regret}(H, I, f)$$

We want to derive a regret bound as a function of $|S|$.

2.2 Reduction from Wide Range Regret to External Regret

Theorem 1. *Given H_{ext} with external regret $R(T,K)$ and losses in $[-1,1]$. It is possible to build an algorithm H for losses in $[0,1]$ such that Regret$(H,S) = R(T,|S|)$, which runs in polynomial time in T, K and $|S|$.*

The reduction:

- Initialize H_{ext} with arms set S.
- A time t algorithm H_{ext} returns distribution q_t over S.
- Choose pair (I, f) with probability $I(t)q_t(I, f)$.
- Take a distribution p_t which is equal before and after (formal definition will be given in the proof).
- Feed H_{ext} loss

$$\bar{\ell}_t(I,f) = I(t) \sum_{a \in A} p_t(a)(\ell_t(f(a)) - \ell_t(a))$$

Intuitively, the loss passed to the black box H_{ext} for (I, f) is such that $q_t(I, f)$ measures the negative of the regret with respect to time selection function I and having modified the output of H_{ext} using function f.

Proof. : For computing p_t algorithm H_{ext} gets $I(t)$ and $q_t(I, f)$. If $\sum_{(I,f) \in S} I(t)q_t(I, f) = 0$ we return an arbitrary p_t. Otherwise, we compute p_t such that

$$p_t^T = p_t^T \left(\sum_{(I,f) \in S} \underbrace{\frac{I(t)q_t(I,f)}{\sum_{(J,g) \in S} J(t)q_t(J,g)}}_{\text{weights are summed to 1}} M_f^t \right)$$

Such p_t exists since every row of the matrix is summed to 1, that is, 1 is eigenvalue of the matrix.

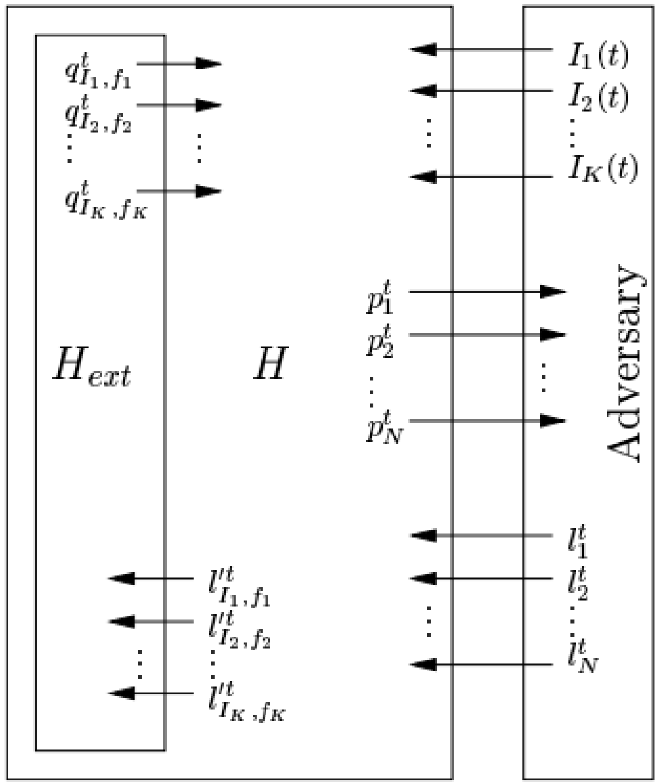

Figure 2: Reduction from External Regret to Wide Range Regret. Note that we denoted h_{ext} losses with $\bar{l}$

We defined the losses as:

$$\bar{\ell}_t(I,f) = I(t)\sum_{a\in A} p_t(a)(\ell_t(f(a)) - \ell_t(a)) = I(t)p_t^T(M_f^t - \mathbb{I})\ell_t$$

Observe that

$$\sum_{t=1}^{T} \bar{\ell}_t(I,f) = \sum_{t=1}^{T} I(t)p_t^T(M_f^t - \mathbb{I})\ell_t = -\text{Regret}(H,I,f)$$

Note that $\bar{\ell}_t(I,f) \in [-1,1]$ since

$$|\bar{\ell}_t(I,f)| = \left| I(t)\sum_{a\in A} p^t(a)(\ell_t(f(a)) - \ell_t(a)) \right| \leqslant |I(t)| \sum_{a\in A} |p_t(a)|\,|(\ell_t(f(a) - \ell_t(a))| \leqslant I(t)\sum_{a\in A} p_t(a)\cdot 1 \leqslant \sum_{a\in A} p_t(a) \leqslant 1$$

From the external regret of H_{ext}

$$\forall (I,f)\in S \;\; \sum_{t=1}^{T} \underbrace{\sum_{(J,g)\in S} q_t(J,g)\bar{\ell}_t(J,g)}_{H_{ext}\text{ loss in time } t} \leqslant \sum_t \bar{\ell}_t(I,f) + R(T,|S|) = -\text{Regret}(H,I,f) + R(T,|S|)$$

Claim which we will prove later is that the expected loss of H_{ext} at time $1 \leqslant t \leqslant T$ is 0, i.e.

$$\sum_{(J,g)\in S} q_t(J,g)\bar{\ell}_t(J,g) = 0$$

Given that, the claim with the inequality above yields:

$$-\text{Regret}(H,I,f) + R(T,|S|) \geqslant \sum_{t=1}^{T} 0 = 0$$

Therefore,

$$\text{Regret}(H, I, f) \leqslant R(T, |S|).$$

as required.

Proof for the claim:

$$\begin{aligned}
\sum_{(J,g)\in S} q_t(J,g)\bar{\ell}_t(J,g) &= \sum_{(J,g)\in S} q_t(J,g)J(t)p_t^T(M_g^t - \mathbb{I})\ell_t = \\
&= \sum_{(J,g)\in S} q_t(J,g)J(t)p_t^T M_g^t \ell_t - \sum_{(J,g)\in S} q_t(J,g)J(t)p_t^T \ell_t = \\
&= p_t^T\left(\sum_{(J,g)\in S} q_t(J,g)J(t)M_g^t\right)\ell_t - \left(\sum_{(J,g)\in S} q_t(J,g)J(t)\right)(p_t^T\ell_t) = \star
\end{aligned}$$

- If $\sum_{(J,g)\in S} J(t)q_t(J,g) \neq 0$,

$$\begin{aligned}
\star &= p_t^T \frac{\sum_{(J,g)\in S} q_t(J,g)J(t)}{\sum_{(J,g)\in S} q_t(J,g)J(t)}\left(\sum_{(J,g)\in S} q_t(J,g)J(t)M_g^t\right)\ell_t - \left(\sum_{(J,g)\in S} q_t(J,g)J(t)\right)(p_t^T\ell_t) = \\
&= \left(\sum_{(J,g)\in S} q_t(J,g)J(t)\right)\underbrace{p_t^T\left(\sum_{(J,g)\in S}\frac{q_t(J,g)J(t)}{\sum_{(J,g)\in S} q_t(J,g)J(t)}M_g^t\right)}_{p_t^T}\ell_t - \left(\sum_{(J,g)\in S} q_t(J,g)J(t)\right)(p_t^T\ell_t) = \\
&= \left(\sum_{(J,g)\in S} q_t(J,g)J(t)\right)(p_t^T\ell_t) - \left(\sum_{(J,g)\in S} q_t(J,g)J(t)\right)(p_t^T\ell_t) = 0,
\end{aligned}$$

$$\forall (I,f)\in S \ \sum_{t=1}^{T}\underbrace{\sum_{(J,g)\in S} q_t(J,g)\bar{\ell}_t(J,g)}_{H_{ext}\text{ loss in time } t} \leqslant \sum_t \bar{\ell}_t(I,f) + R(T,|S|) = -\text{Regret}(H,I,f) + R(T,|S|)$$

  where the last identity follows from the way we selected p_t.

- Otherwise, if $\sum_{(J,g)} J(t)q_t(J,g) = 0$, since all of the summands are non-negative, for every $(J,g) \in S$, $q_t(J,g)J(t) = 0$ that is either $q_t(J,g) = 0$ or $J(t) = 0$ and obviously $\star = 0$, as required.

□

3 Minimizing Regret with Time Selection Functions

We now present an online algorithm that achieves good external regret bound in presence of time selection functions. The regret bound will depend on $L_{\min}$, the loss of the best action w.r.t the time selection function I, making it a first-order regret bound (zero-order regret bound depends on the global time T). Formally, $L_{\min}$ is defined to be:

$$L_{\min} = \max_I \min_{(I,f)\in S} \sum_{t=1}^{T} I(t)p_t^T M_f^t \ell_t$$

Where M_f^t is defined the same as in the previous section.

Theorem 2. *Regret* $\leqslant O\left(\sqrt{L_{\min}\log|S|}+\log|S|\right)$

Proof. We present an algorithm and analyze its regret. First let's define loss of algorithm H w.r.t I:

$$L_T^{H,I}=\sum_{t=1}^{T} I(t)p_t^T\ell_t$$

and the loss of algorithm H w.r.t (I,f):

$$L_T^{H,I,f}=\sum_{t=1}^{T} I(t)p_t^T M_f^t\ell_t$$

for simplicity we assume that $\forall t\ \exists I\ s.t.\ I(t)\neq 0$ (otherwise all the losses at time t are zero).
Let $\beta\in(0,1)$ be a parameter we will fix later. We define the *Reduced Regret* to be:

$$\tilde{R}_t^{H,I,f}=\beta L_t^{H,I}-L_t^{H,I,f}$$

Note that if $\beta=1$, this is the swap regret. We will use the reduced regret for the regret analysis, but we still want to minimize the regular regret.

Our algorithm will assign a weight $w_t(I,f)$ to every pair of time-selection function I and modification rule f:

$$w_t(I,f)=\beta^{-\tilde{R}_t^{H,I,f}},\ \text{and}\ w_0(I,f)=1$$

and we normalize the weights to a distribution,

$$q_{t+1}(I,f)=\frac{w_t(I,f)}{W_t},\ \text{where}\ W_t=\sum_{(I,f)\in S} w_t(I,f)$$

Intuition for the choice of weights If we have high regret from a pair (I,f) we would like to increase their weight, that way we will play them more and the regret from this pair won't be high anymore. Note that because $\beta\in(0,1)$, $w_t(I,f)=\beta^{-\tilde{R}_t^{H,I,f}}$ increases with the regret.

We are now ready to describe the algorithm. For each time t, our algorithm H will do the following:

1. Compute $q_t(I,f)$.
2. Choose p_t just like in the previous section:

$$p_t^T=p_t^T\left(\sum_{(I,f)\in S}\frac{I(t)q_t(I,f)}{\sum_{(J,g)\in S}J(t)q_t(J,g)}M_f^t\right)$$

(Recall that we assumed that $\forall t\exists I\ s.t.\ I(t)\neq 0$, therefore the denominator is not zero)

Claim 1. $\forall t\ \sum_{(I,f)\in S} w_t(I,f)\leqslant\sum_{(I,f)\in S} w_{t-1}(I,f)$

The proof of the claim will be given later this section. We now complete the proof of the theorem.

From Claim 1:

$$\beta^{-(\beta L_T^{H,I}-L_T^{H,I,f})}=\beta^{-\tilde{R}_T^{H,I,f}}=w_t(I,f)\leqslant\sum_{(I,f)\in S} w_0(I,f)=|S|$$

Taking logarithm from both sides we get:

$$\left(\beta L_T^{H,I}-L_T^{H,I,f}\right)\log\left(\frac{1}{\beta}\right)\leqslant\log|S|$$

Therefore:

$$L_T^{H,I} \leqslant \frac{1}{\beta}\left(L_T^{H,I,f} + \frac{\log|S|}{\log\frac{1}{\beta}}\right)$$

Note that the inequality above is true for all pairs $(I,f) \in S$, specifically it is also true for the (I,f) pair that give $L_{\min}$, so:

$$L_T^{H,I} \leqslant \frac{1}{\beta}\left(L_{\min} + \frac{\log|S|}{\log\frac{1}{\beta}}\right)$$

If we choose β s.t:

$$\frac{1}{\beta} = 1 + \min\left\{\frac{1}{2}, \sqrt{\frac{\log|S|}{L_{\min}}}\right\}$$

we get:

$$Regret \leqslant O\left(\sqrt{L_{min}\log|S|} + \log|S|\right)$$

□

Notice that β is dependent on $L_{\min}$, so it can not be chosen beforehand. This can be solved using the *doubling trick*: At each time step calculate $L_{\min}$ and every time $L_{\min}$ doubles itself, restart the algorithm. This will multiply the regret bound by a constant factor.

We now prove Claim 1.

Proof. Note that for any $\beta \in (0,1)$ and $x \in [0,1]$ we have $\beta^x \leqslant 1-(1-\beta)x$ and $\beta^{-x} \leqslant 1+(1-\beta)\frac{x}{\beta}$. Therefore,

$$\begin{aligned}
\sum_{(I,f)\in S} w_t(I,f) &= \sum_{(I,f)\in S} w_{t-1}(I,f)\beta^{I(t)(p_t^T M_f^t \ell_t - \beta p_t^T \ell_t)} \\
&\leqslant \sum_{(I,f)\in S} w_{t-1}(I,f)\left[1-(1-\beta)I(t)(p_t)^T M_f^t \ell_t\right]\cdot\left[1+(1-\beta)I(t)p_t^T \ell_t\right] \\
&\leqslant \sum_{(I,f)\in S} w_{t-1}(I,f) - (1-\beta)W_{t-1}\sum_{(I,f)\in S} q_t(I,f)I(t)p_t^T M_f^t \ell_t + (1-\beta)W_{t-1}\sum_{(I,f)} q_t(I,f)I(t)p_t^T \ell_t \\
&= \sum_{(I,f)\in S} w_{t-1}(I,f) - (1-\beta)W_{t-1}p_t^T\left(\sum_{(I,f)\in S} q_t(I,f)I(t)M_f^t \ell_t\right) + (1-\beta)W_{t-1}p_t^T\left(\sum_{(I,f)} q_t(I,f)I(t)\right)\ell_t \\
&= \sum_{(I,f)\in S} w_{t-1}(I,f)
\end{aligned}$$

where the last equation is due to our choice of p_t. □