8803 Connections between Learning, Game Theory, and Optimization

Maria-Florina Balcan — Lecture 4: September 2nd, 2010

1 From External Regret to Swap Regret

1.1 Model and Preliminaries

We assume an adversarial online model where there are N available actions $X = \{1, \dots, N\}$. At each time step t, an online algorithm H selects a distribution p^t over the N actions. After that, the adversary selects a loss vector $\ell^t \in [0,1]^N$, where $\ell_i^t \in [0,1]$ is the loss of the i-th action at time t. In the *full information model*, the online algorithm H receives the loss vector ℓ^t and experiences a loss $\ell_H^t = \sum_{i=1}^N p_i^t \ell_i^t$. (This can be viewed as an expected loss when the online algorithm selects action $i \in X$ with probability p_i^t.) In the *partial information model*, the online algorithm receives $(\ell_{k^t}^t, k^t)$, where k^t is distributed according to p^t, and $\ell_H^t = \ell_{k^t}^t$ is its loss. The loss of the i-th action during the first T time steps is $L_i^T = \sum_{t=1}^T \ell_i^t$, and the loss of H is $L_H^T = \sum_{t=1}^T \ell_H^t$.

The aim for the *external regret* setting is to design an online algorithm that will be able to approach the performance of the best algorithm from a given class of algorithms $\mathcal{G}$; namely, to have a loss close to $L_{\mathcal{G},min}^T = \min_{g \in \mathcal{G}} L_g^T$. Formally we would like to minimize the external regret $R_{\mathcal{G}} = L_H^T - L_{\mathcal{G},min}^T$, and $\mathcal{G}$ is called the *comparison class*. The most studied comparison class $\mathcal{G}$ is the one that consists of all the single actions, i.e., $\mathcal{G} = X$. In this context, we want the online algorithm's loss to be close to $L_{min}^T = \min_i L_i^T$, and the external regret is defined as $R = L_H^T - L_{min}^T$.

External regret uses a fixed comparison class $\mathcal{G}$, but one can also envision a comparison class that depends on the online algorithm's actions. We can consider modification rules that modify the actions selected by the online algorithm, producing an alternative strategy which we will want to compete against. A *modification rule* F has as input the history and the current action selected by the online procedure and outputs a (possibly different) action. (We denote by F^t the function F at time t, including any dependency on the history.) Given a sequence of probability distributions p^t used by an online algorithm H, and a modification rule F, we define a new sequence of probability distributions $f^t = F^t(p^t)$, where $f_i^t = \sum_{j:F^t(j)=i} p_j^t$. The loss of the modified sequence is $L_{H,F} = \sum_t \sum_i f_i^t \ell_i^t$. Note that at time t the modification rule F shifts the probability that H assigned to action j to action $F^t(j)$. This implies that the modification rule F generates a different distribution, as a function of the online algorithm's distribution p^t.

We will focus on the case of a finite set $\mathcal{F}$ of memoryless modification rules (they do not depend on history). Given a sequence of loss vectors, the regret of an online algorithm H

with respect to the modification rules $\mathcal{F}$ is

$$R_{\mathcal{F}} = \max_{F \in \mathcal{F}} \{L_H^T - L_{H,F}^T\}.$$

Note that the external regret setting is equivalent to having a set $\mathcal{F}^{ex}$ of N modification rules F_i, where F_i always outputs action i. For *internal regret*, the set $\mathcal{F}^{in}$ consists of $N(N-1)$ modification rules $F_{i,j}$, where $F_{i,j}(i) = j$ and $F_{i,j}(i') = i'$ for $i' \neq i$. That is, the internal regret of H is

$$\max_{F \in \mathcal{F}^{in}} \{L_H^T - L_{H,F}^T\} \;=\; \max_{i,j \in X} \left\{ \sum_{t=1}^{T} p_i^t (\ell_i^t - \ell_j^t) \right\}.$$

A more general class of memoryless modification rules is *swap regret* defined by the class $\mathcal{F}^{sw}$, which includes all N^N functions $F : \{1,\ldots,N\} \to \{1,\ldots,N\}$, where the function F swaps the current online action i with $F(i)$ (which can be the same or a different action). That is, the swap regret of H is

$$\max_{F \in \mathcal{F}^{sw}} \{L_H^T - L_{H,F}^T\} \;=\; \sum_{i=1}^{N} \max_{j \in X} \left\{ \sum_{t=1}^{T} p_i^t (\ell_i^t - \ell_j^t) \right\}.$$

Note that since $\mathcal{F}^{ex} \subseteq \mathcal{F}^{sw}$ and $\mathcal{F}^{in} \subseteq \mathcal{F}^{sw}$, both external and internal regret are upper-bounded by swap regret.

1.2 Generic conversion from external to swap regret

In this section we give an elegant and very general *black-box conversion* showing how any procedure A achieving good external regret can be used as a subroutine to achieve good swap regret as well.

The high-level idea is as follows (see also Fig. 1). We will instantiate N copies $A_1, \ldots, A_N$ of the external-regret procedure. At each time step, these procedures will each give us a probability vector, which we will combine in a particular way to produce our own probability vector p. When we receive a loss vector ℓ, we will partition it among the N procedures, giving procedure A_i a fraction p_i (p_i is our probability mass on action i), so that A_i's belief about the loss of action j is $\sum_t p_i^t \ell_j^t$, and matches the cost we would incur putting i's probability mass on j. In the proof, procedure A_i will in some sense be responsible for ensuring low regret of the $i \to j$ variety.

The key to making this work is that we will be able to define the p's so that the sum of the losses of the procedures A_i on their own loss vectors matches our overall true loss.

To be specific, let us formalize what we mean by an external regret procedure.

Definition 1 *An R external regret procedure A guarantees that for any sequence of T losses ℓ^t and for any action $j \in \{1,\ldots,N\}$, we have*

$$L_A^T = \sum_{t=1}^{T} \ell_A^t \leq \sum_{t=1}^{T} \ell_j^t + R = L_j^T + R.$$

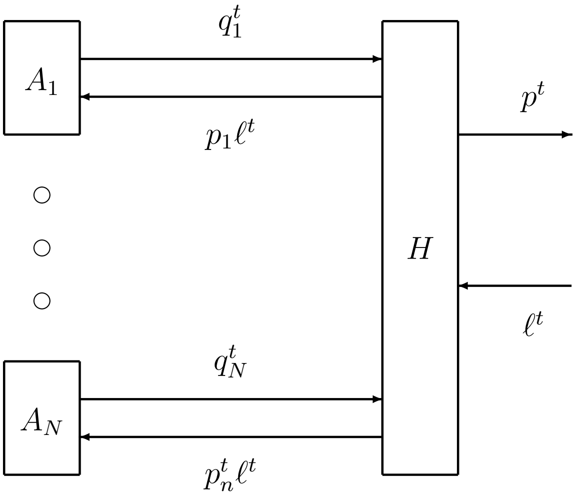

Figure 1: The structure of the swap regret reduction.

Theorem 1 *Given an R external regret procedure, the master online procedure H has the following guarantee. For every function $F : \{1,\ldots,N\} \to \{1,\ldots,N\}$,*

$$L_H \leq L_{H,F} + NR\,,$$

i.e., the swap regret of H is at most NR.

Proof: We assume we have N copies $A_1,\ldots,A_N$ of an R external regret procedure. We combine the N procedures to one master procedure H as follows. At each time step t, each procedure A_i outputs a distribution q_i^t, where $q_{i,j}^t$ is the fraction it assigns action j. We compute a single distribution p^t such that

$$p_j^t = \sum_i p_i^t q_{i,j}^t.$$

That is, $p^t = p^t Q^t$, where p^t is our distribution and Q^t is the matrix of $q_{i,j}^t$. (We can view p^t as a stationary distribution of the Markov Process defined by Q^t, and it is well known such a p^t exists and is efficiently computable.) For intuition into this choice of p^t, notice that it implies we can consider action selection in two equivalent ways. The first is simply using the distribution p^t to select action j with probability p_j^t. The second is to select procedure A_i with probability p_i^t and then to use A_i to select the action (which produces distribution $p^t Q^t$).

When the adversary returns the loss vector ℓ^t, we return to each A_i the loss vector $p_i \ell^t$. So, procedure A_i experiences loss $(p_i^t \ell^t) \cdot q_i^t = p_i^t (q_i^t \cdot \ell^t)$.

Since A_i is an R external regret procedure, for any action j, we have,

$$\sum_{t=1}^{T} p_i^t (q_i^t \cdot \ell^t) \quad \leq \quad \sum_{t=1}^{T} p_i^t \ell_j^t + R \tag{1}$$

If we sum the losses of the N procedures at a given time t, we get

$$\sum_i p_i^t (q_i^t \cdot \ell^t) = p^t Q^t \ell^t,$$

where p^t is the row-vector of our distribution, Q^t is the matrix of $q_{i,j}^t$, and ℓ^t is viewed as a column-vector. By design of p^t, we have $p^t Q^t = p^t$. So, the sum of the perceived losses of the N procedures is equal to our actual loss $p^t \ell^t$.

Therefore, summing equation (1) over all N procedures, the left-hand-side sums to L_H^T, where H is our master online procedure. Since the right-hand-side of equation (1) holds for any j, we have that for any function $F : \{1, \ldots, N\} \to \{1, \ldots, N\}$,

$$L_H^T \leq \sum_{i=1}^{N} \sum_{t=1}^{T} p_i^t \ell_{F(i)}^t + NR = L_{H,F}^T + NR,$$

as desired. ■

Using known guarantees about the randomized weighted majority algorithm we can immediately derive the following corollary.

Corollary 1 *There exists an online algorithm H such that for every function $F : \{1, \ldots, N\} \to \{1, \ldots, N\}$, we have that*

$$L_H \leq L_{H,F} + O(N\sqrt{T \log N})\,,$$

i.e., the swap regret of H is at most $O(N\sqrt{T \log N})$.

Acknowledgment: The material in this lecture is from chapter "Learning, Regret Minimization, and Equilibria" by Avrim Blum and Yishay Mansour.