Lecture 05: Φ-Regret Minimization

Lecturer: John Lazarsfeld

September 29, 2025

Abstract

Beyond external regret: swap regret and the Φ-regret framework. Introduction and analysis of the Blum-Mansour and Gordon-Greenwald-Marks algorithms.

Recap from Lecture 04. In the previous lectures, we have introduced general families of online learning algorithms and have established analysis frameworks for proving optimal, sublinear regret bounds in the full or gradient-feedback models, as well as the extension of these algorithms to the bandit-feedback model. However, these algorithms all minimize *external regret*. In this lecture, we focus on more refined comparator classes that correspond to the stronger notion of Φ-regret as well as new algorithms for Φ-regret minimization.

1 Beyond External Regret

Comparator Classes as Transformations of Actions. The external regret of an online algorithm $\mathcal{A}$ measures the difference between its cumulative incurred cost from playing iterates $\{x_t\}$, and the cumulative cost of an optimal *fixed action*:

$$\mathrm{Reg}_{\mathcal{A}}(T) = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} f_t(x) .$$

As introduced in Lecture 01, we view the right hand term (which does not depend on the algorithm's output) as a minimization over a set of *comparators*, which in this case is the set of all actions in $\mathcal{X}$.

We could equivalently view this minimization as over the set of *constant* maps $\Phi_{\text{constant}} = \{\phi_x\}$ where for any $x \in \mathcal{X}$, we have $\phi_x(y) = x$ for all $y \in \mathcal{X}$. This leads to equivalence:

$$\mathrm{Reg}_{\mathcal{A}}(T) = \sum_{t=1}^{T} f_t(x_t) - \min_{\phi \in \Phi_{\text{constant}}} \sum_{t=1}^{T} f_t(\phi(x)) .$$

This leads to the following interpretation of the cumulative cost of the comparator in the definition of regret: we measure the total cost of the strategy that, at time t, instead of playing action x_t, plays action $\phi(x_t)$.

This viewpoint of comparators as transformations of actions extends far beyond the set of constant transformations. For example, suppose Φ is some general set of maps of the form $\phi : \mathcal{X} \to \mathcal{X}$. Minimizing over such transformations yields a natural generalization of comparators and regret. In the following section, we first introduce the special case of linear transformations and swap regret before introducing the full generalization.[1]

2 Swap-Regret and the Blum-Mansour Algorithm

In this section we start by considering a special (and natural) class of *swap* transformations, which leads to the notion of *swap-regret*.

[1]The content of these notes roughly follows those of Farina (2024).

Swap Transformations as Stochastic Matrices. In the experts setting with $\mathcal{X} = \Delta_n$ and $f_t(x) = \langle x, \ell_t \rangle$ for all t, a *swap* transformation is a linear operator $P : \Delta_n \to \Delta_n$. In particular, we represent P as a column-stochastic matrix $P = (p^{(1)}, \ldots, p^{(n)}) \in \mathbb{R}^{n \times n}$ where each $p^{(i)} \in \Delta_n$. Let Φ_{swap} be the set

$$\Phi_{\text{swap}} = \left\{ P = (p^{(1)}, \ldots, p^{(n)}) \in \mathbb{R}^{n \times n} : \text{ each } p^{(i)} \in \Delta_n \right\}$$

that contains all such column-stochastic matrices. Observe for fixed n that Φ_{swap} is a convex, compact set. Using the set Φ_{swap} as a comparator class then leads to the following notion of *swap regret*:

Definition 1. *Let $\{x_t\}$ be the iterates of an online learning algorithm $\mathcal{A}$ be an online learning algorithm on loss functions $\{f_t\}$. Fix a family of transformations Φ such that $\phi : \mathcal{X} \to \mathcal{X}$ for each $\phi \in \Phi$. Then the swap regret of algorithm $\mathcal{A}$ after T rounds is*

$$\text{SwapReg}_{\mathcal{A}}(T) = \sum_{t=1}^{T} f_t(x_t) - \min_{P \in \Phi_{\text{swap}}} \sum_{t=1}^{T} f_t(P x_t) \ .$$

Swap Regret as an Upper Bound on External Regret. Suppose that $\mathcal{X} = \Delta_n$ and each loss function f_t is linear. Then observe that for every $x \in \mathcal{X}$, the stochastic matrix $P_{e_i} = e_i \mathbf{1}^\top \in \Phi_{\text{swap}}$ maps every $x \in \Delta_n$ to the vertex e_i. It follows that

$$\min_{x \in \Delta_n} \sum_{t=1}^{T} f_t(x) = \min_{e_i : i \in [n]} \sum_{t=1}^{T} f_t(e_i) = \min_{P_{e_i} : i \in [n]} \sum_{t=1}^{T} f_t(P_{e_i} x_t) \geq \min_{P \in \Phi_{\text{swap}}} \sum_{t=1}^{T} f_t(P x_t)$$

where the inequality follows from the fact that $\{P_{e_i} : i \in [n]\} \subset \Phi_{\text{swap}}$. Thus for any online learning algorithm $\mathcal{A}$:

$$\text{Reg}_{\mathcal{A}}(T) = \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \Delta_n} \sum_{t=1}^{T} f_t(x) \leq \sum_{t=1}^{T} f_t(x_t) - \min_{P \in \Phi_{\text{swap}}} \sum_{t=1}^{T} f_t(P x_t) = \text{SwapReg}_{\mathcal{A}}(T) \ . \tag{1}$$

The Blum-Mansour Algorithm for Swap-Regret Minimization. We now present an algorithm of Blum and Mansour (2007) (which we will refer to as BM) for *minimizing swap-regret*.

Algorithm 1 Blum-Mansour Algorithm (BM) for Experts Setting

Input: Initial $x_1 \in \Delta_n$; for $i \in [n]$, online learning algorithm $\mathcal{A}^{(i)}$ with initialization $p_t^{(i)} = e_i$.
for $t = 1, \ldots, T$ **do**:

1. Play action $x_t \in \Delta_n$, and incur cost $f_t(x_t) = \langle x_t, \ell_t \rangle$. Observe loss vector $\ell_t \in \mathbb{R}^n$.
2. For each $i \in [n]$:
   (i) Give to algorithm $\mathcal{A}^{(i)}$ the loss function $g_t^{(i)}(x) = \langle x, x_t^{(i)} \cdot \ell_t \rangle$.
   (ii) Receive from algorithm $\mathcal{A}^{(i)}$ the output $p_{t+1}^{(i)} \in \Delta_n$.
3. Construct the column-stochastic matrix $P_{t+1} = (p_{t+1}^{(1)}, \ldots, p_{t+1}^{(n)}) \in \Phi_{\text{swap}}$.
4. Select $x_{t+1} \in \Delta_n$ satisfying the fixed point equation

$$x_{t+1} = P_{t+1} x_{t+1} \ . \tag{2}$$

end for

Remark 2. We make several remarks about the BM algorithm:

- **Overall intuition:** Just as in the usual (full-feedback) experts setting framework, at every time step, the algorithm plays an action $x_t \in \Delta_n$, incurs a loss $f_t(x_t)$, observes the function f_t, and uses this information to choose its next action x_{t+1}. However, to choose its actions $\{x_t\}$, the BM algorithm uses as subroutines n distinct online learning algorithms $\mathcal{A}^{(i)}$.

  In particular, we assume each $\mathcal{A}^{(i)}$ also operates in the experts setting: it plays actions $p_t^{(i)} \in \Delta_n$ at each timestep and receives linear loss functions $g_t^{(i)}$ (in this case, these loss functions $g_t^{(i)}$ will be constructed and given as input to each $\mathcal{A}^{(i)}$ by the outer BM algorithm).

- **Fixed-point computation**: Observe in expression (2) that the algorithm must output x_{t+1} satisfying the fixed point equation $x_{t+1} = P_{t+1}x_{t+1}$. As P_{t+1} is a stochastic matrix, observe that this is equivalent to finding the *stationary distribution* of the finite-state Markov Chain represented by P_{t+1}. Note that such a stationary distribution is guaranteed to exist by Brouwer's fixed point theorem or by an eigenvector computation.

Analysis of Blum-Mansour: Reducing Swap to External Regret. We state and prove the following guarantee for the BM algorithm that uses the set of subroutines $\{\mathcal{A}^{(i)}\}$.

Theorem 3. *Let $\{x_t\}$ be the iterates of Blum-Mansour (Algorithm 1) on the sequence of linear loss function $\{f_t(x) = \langle x, \ell_t\rangle\}$ using online learning algorithms $\{\mathcal{A}^{(i)}\}_{i\in[n]}$ as subroutines. Then*

$$\mathrm{SwapReg}_{BM}(T) \le \sum_{i=1}^{n} \mathrm{Reg}_{\mathcal{A}^{(i)}}(T) .$$

Remark 4. We make several remarks about this swap regret guarantee:

- **Instantiation using MWU as Subroutines:** If all subroutines $\mathcal{A}^{(i)}$ are instantiated to be the MWU algorithm, then recall from Lecture 02 that $\mathrm{Reg}_{MWU}(T) \le 2\sqrt{T\log n}$ with an appropriately tuned stepsize. This leads to the following concrete corollary:

  Corollary 5. *Assume the setting of Theorem 3 and suppose each $\mathcal{A}^{(i)}$ is set to MWU with stepsize $\eta := \sqrt{\frac{\log n}{T}}$. Then $\mathrm{Reg}_{\mathcal{A}^{(i)}}(T) \le 2\sqrt{T\log n}$ for each $i \in [n]$, and*

  $$\mathrm{SwapReg}_{BM}(T) \le \sum_{i=1}^{n} \mathrm{Reg}_{MWU}(T) \le 2n \cdot \sqrt{T\log n} .$$

  The proof of the corollary follows as a direct consequence of the regret bound for MWU from Lecture 02 and the swap-regret bound of BM stated in Theorem 3.

- **Polynomial dimension-dependence:** Observe in Corollary 5 that the swap regret guarantee scales polynomially in n. An active line of research in the past several years has been devoted to designing different algorithms for swap regret minimization with improved dependence on the dimension n of the action space (see, e.g., Dagan et al. (2023) and Peng and Rubinstein (2024))

Proof. The proof of the theorem follows from (i), using the fixed-point property in Step 4 of the algorithm, and (ii) using the linearity of the loss functions f_t to reduce the swap-regret of BM to the external regret of the individual subroutines $\mathcal{A}^{(i)}$.

For this, fix a comparator $P = (p^{(1)}, \dots, p^{(n)}) \in \Phi_{\text{swap}}$, and observe that

$$\sum_{t=1}^{T} f_t(x_t) - f_t(Px_t) = \sum_{t=1}^{T} f_t(P_t x_t) - f_t(Px_t) \tag{3}$$

$$= \sum_{t=1}^{T} \langle \ell_t, P_t x_t \rangle - \langle \ell_t, P x_t \rangle \tag{4}$$

$$= \sum_{t=1}^{T} \left(\sum_{i=1}^{n} x_t^{(i)} \cdot \langle \ell_t, p_t^{(i)} \rangle \right) - \left(\sum_{i=1}^{n} x_t^{(i)} \cdot \langle \ell_t, p^{(i)} \rangle \right) . \tag{5}$$

Here, the expression (3) follows from the fixed-point update rule in (2) (which also holds at time $t = 1$ given the initialization of each $p_1^{(i)}$ in the algorithm description), and expressions (4) and (5) follow by the definition and linearity of the loss functions f_t.

Moreover, observe for each $i \in [n]$ that for any $x \in \Delta_n$, the quantity $x_t^{(i)} \cdot \langle \ell_t, p_t^{(i)} \rangle$ is the function value $g_t^{(i)}(x)$ as specified in step (2.i) of the algorithm. Thus we can further write:

$$\sum_{t=1}^{T} f_t(x_t) - f_t(Px_t) = \sum_{t=1}^{T} \sum_{i=1}^{n} \left(g_t^{(i)}(p_t^{(i)}) - g_t^{(i)}(p^{(i)}) \right) \tag{6}$$

$$= \sum_{i=1}^{n} \sum_{t=1}^{T} \left(g_t^{(i)}(p_t^{(i)}) - g_t^{(i)}(p^{(i)}) \right) \tag{7}$$

$$\leq \sum_{i=1}^{n} \left(\sum_{t=1}^{T} g_t^{(i)}(p_t^{(i)}) - \min_{p^\star \in \Delta_n} \sum_{t=1}^{T} g_t^{(i)}(p^\star) \right) = \sum_{i=1}^{n} \mathrm{Reg}_{\mathcal{A}^{(i)}}(T) , \tag{8}$$

where the final inequality is due to the definition of *external* regret for the algorithm $\mathcal{A}^{(i)}$.

Minimizing the inequality over all $P \in \Phi_{\text{swap}}$ then yields

$$\mathrm{SwapReg}_{BM}(T) = \sum_{t=1}^{T} f_t(x_t) - \min_{P \in \Phi_{\text{swap}}} \sum_{t=1}^{T} f_t(Px_t) \leq \sum_{i=1}^{n} \mathrm{Reg}_{\mathcal{A}^{(i)}}(T) . \qquad \square$$

3 Φ-Regret and the Gordon-Greenwald-Marks Framework

The Blum-Mansour algorithm for swap-regret minimization can be seen as a special case of the more general Φ-regret-minimization framework introduced by Gordon et al. (2008).

Formalizing Φ-Regret. As introduced earlier, we now consider a general family of comparators/transformations Φ, where $\phi : \mathcal{X} \to \mathcal{X}$ for each $\phi \in \Phi$. This leads to the general definition of Φ-regret:

Definition 6. *Let $\{x_t\}$ be the iterates of an online learning algorithm $\mathcal{A}$ be an online learning algorithm on loss functions $\{f_t\}$. Fix a family of transformations Φ. Then the Φ-Regret of algorithm $\mathcal{A}$ after T rounds is*

$$\Phi\text{-}\mathrm{Reg}_{\mathcal{A}}(T) = \sum_{t=1}^{T} f_t(x_t) - \inf_{\phi \in \Phi} \sum_{t=1}^{T} f_t(\phi(x_t)) .$$

Remark 7. We make several remarks about the Φ-regret definition:

- **Assumptions on Φ**: In Definition 6 we have made no assumptions about the structure of Φ. However, for simplicity we will generally assume that Φ is a convex and compact subset of a d-dimensional reproducing-kernel Hilbert space (RKHS). See the discussion in Gordon et al. (2008, Section 2.1).

Special cases: As mentioned, external regret and swap regret both arise as special cases of Φ-regret: we recover external regret when Φ is the set of *constant* transformations $\{\phi_x(y) : y \mapsto x \mid x \in \mathcal{X}\}$, and we recover swap regret using the set Φ_{swap}.

Gordon-Greenwald-Marks Framework. For a fixed set Φ, the framework of Gordon et al. (2008) gives a simple method for reducing Φ-regret minimization to external-regret minimization. We state the steps of this in the following algorithm for the case when $\mathcal{X} = \Delta_n$.

Algorithm 2 Gordon-Greenwald-Marks (GGM) Algorithm for Experts Setting

Input: Initial $x_1 \in \Delta_n$; online learning algorithm $\mathcal{A}_\Phi$ over Φ with ϕ_1 such that $x_1 = \phi_1(x_1)$.
for $t = 1, \ldots, T$ **do**:

1. Play action $x_t \in \Delta_n$ and incur cost $f_t(x_t) = \langle x_t, \ell_t \rangle$. Observe loss vector $\ell_t \in \mathbb{R}^n$.
2. For the subroutine $\mathcal{A}_\Phi$:
   (i) Construct loss function $G_t : \Phi \to \mathbb{R}$ given by $G_t(\phi) := \langle \phi(x_t), \ell_t \rangle$ for $\phi \in \Phi$.
   (ii) Give to $\mathcal{A}_\Phi$ the loss function G_t .
   (iii) Receive as output from $\mathcal{A}_\Phi$ the transformtion $\phi_{t+1} \in \Phi$.
3. Select $x_{t+1} \in \Delta_n$ satisfying the fixed point expression

$$x_{t+1} = \phi_{t+1}(x_{t+1}) \,. \tag{9}$$

end for

Remark 8. We make several remarks about the GGM algorithm:

- **Blum-Mansour as a Special Case of Gordon-Greenwald-Marks**: The Blum-Mansour algorithm (Algorithm 1) is in some sense a special case of the Gordon-Greenwald-Marks framework: the matrix P_{t+1} constructed using the outputs of the n subroutines $\mathcal{A}^{(i)}$ in BM is effectively the output of $\mathcal{A}_\Phi$ in the GGM framework. The key characteristic of the GGGM framework is that it does not specify how to encode the set of transformations Φ into a representation that is conducive to external regret minimization. For the set of stochastic matrices Φ_{swap}, the BM algorithm sidesteps this possible issue by parameterizing the set of stochastic matrices by n distinct column vectors, and running external regret minimization over each of the columns. If Φ can be represented as a simpmlex, then the external regret minimization over Φ is a certain instance of the experts setting.
- **Computational Aspects of Fixed-Point Calculation**: A second important consideration is that the GGM framework again requires a fixed point computation (expression (9)) at each timestep. For some sets of transformations Φ, such computations could be straightforward (e.g., for the case of Φ_{swap}, which can be done via an eigenvector computation, or using the power method), but in general this subproblem can be computationally hard.

Disregarding any computational challenges, the following theorem shows that the GGM algorithm's Φ-regret guarantee reduces to the *external* regret guarantee of the subroutine $\mathcal{A}_\Phi$:

Theorem 9. *Fix Φ, and let $\{x_t\}$ be the iterates of GGM (Algorithm 2) on loss functions $\{f_t\}$ using subroutine $\mathcal{A}_\Phi$. Let $\mathrm{Reg}_{\mathcal{A}_\Phi}(T)$ denote the external regret of $\mathcal{A}_\Phi$ over the loss functions $\{G_t\}$ as defined in step (2.i) of the algorithm. Then*

$$\Phi\text{-Reg}_{GGM}(T) = \mathrm{Reg}_{\mathcal{A}_\Phi}(T) \,.$$

Proof. First, observe from Step 2 of the GGM algorithm that the subroutine $\mathcal{A}_\Phi$ has outputs $\{\phi_t\}$ against the sequence of losses $\{G_t\}$. Thus using the definition of $G_t(\phi) = \langle \phi(x_t), \ell_t \rangle$ and

$\mathrm{Reg}_{\mathcal{A}_\Phi}(T)$ we have

$$\mathrm{Reg}_{\mathcal{A}_\Phi}(T) = \sum_{t=1}^{T} \langle \phi_t(x_t), \ell_t \rangle - \min_{\phi^\star \in \Phi} \sum_{t=1}^{T} \langle \phi^\star(x_t), \ell_t \rangle \tag{10}$$

$$= \sum_{t=1}^{T} \langle x_t, \ell_t \rangle - \min_{\phi^\star \in \Phi} \sum_{t=1}^{T} \langle \phi^\star(x_t), \ell_t \rangle = \Phi\text{-}\mathrm{Reg}_{GGM}(T) \; . \tag{11}$$

Here, the second line follows by the fixed point property of the GGM iterates from (9). □

References

Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8(6), 2007.

Yuval Dagan, Constantinos Daskalakis, Maxwell Fishelson, and Noah Golowich. From external to swap regret 2.0: An efficient reduction and oblivious adversary for large action spaces. *arXiv preprint arXiv:2310.19786*, 2023.

Gabriele Farina. ϕ-regret minimization, 2024. URL `https://www.mit.edu/~gfarina/2024/6S890f24_L08_phi/`.

Geoffrey J Gordon, Amy Greenwald, and Casey Marks. No-regret learning in convex games. In *Proceedings of the 25th international conference on Machine learning*, pages 360–367, 2008.

Binghui Peng and Aviad Rubinstein. Fast swap regret minimization and applications to approximate correlated equilibria. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1223–1234, 2024.