

Modern Algorithmic Game Theory

Martin Schmid

Department of Applied Mathematics
Charles University in Prague

November 5, 2025

The background image shows a wide, frozen body of water, likely a lake or reservoir, covered in a thick layer of white snow. Dark, winding paths of ice or asphalt cut through the snow, forming a network of roads or lanes. The perspective is from an aerial view, looking down at the intricate patterns created by the different routes.

Regret Minimization

Introduction to the Concept of Regret

We consider a problem of repeatedly making decisions in an uncertain environment.

We assume the following:

- A set of N actions \mathcal{A} , e.g. {rock, paper, scissors}
- An algorithm \mathcal{H} that produces a probabilistic policy π at each timestep
- An adversarial environment that selects a loss vector x in response to the policy π

Examples:

- Choosing a road to take to work; the environment responds with traffic on roads
- Choosing an action in a game; the environment responds with an action

Introduction to the Concept of Regret

- We would like to study worst-case performance guarantees of our algorithm \mathcal{H} , even if the loss function is not known in advance and can be chosen arbitrarily by the environment
- We will use the notion of **regret** which provides strong guarantees in such settings
- Regret tells us how much we could have improved in **retrospect** if we had used an alternative policy π' ; it is defined as the difference between the loss of our policy π and the alternative policy π'
- The alternative policy comes from a **comparison class** \mathcal{G}
- A comparison class consisting of individual actions \mathcal{A} leads to the notion of **external regret**; we compare our performance to the best single action in retrospect
- There are also **internal** and **swap** regrets which allow richer comparison classes

Model Formalization

- We have a set of actions $\mathcal{A} = \{1, \dots, n\}$
- At each time step t , the algorithm \mathcal{H} chooses a policy $\pi^t \in \Delta(\mathcal{A})$
- The algorithm then receives a loss vector $l^t \in \mathbb{R}^{|\mathcal{A}|}$;
- The algorithm's loss at time t is the weighted sum $L^t = \sum_{a \in \mathcal{A}} \pi_a^t l_a^t$
- The cumulative loss of action a is $L_a^T = \sum_{t=1}^T l_a^t$
- The algorithm's cumulative loss after T timesteps is simply $L^T = \sum_{t=1}^T L^t$
- The external regret R_a^T of action a compares the cumulative loss that **would have been received** if we had played action a at each timestep t to the cumulative loss we have received. Mathematically, $R_a^T = L^T - L_a^T$
- Finally, the external regret of π is $R_\pi = \max_{a \in \mathcal{A}} R_a^T$

Regret with Respect to the Optimal Sequence

Theorem

Let \mathcal{G}_{all} be a comparison class consisting of all functions mapping times $\{1, \dots, T\}$ to actions $\mathcal{A} = \{1, \dots, n\}$. Then, for any online algorithm \mathcal{H} , there exists a sequence of losses $l^1 \dots l^T$, such that the regret $R_{\mathcal{G}_{all}}$ is at least $T(1 - \frac{1}{N})$.

Proof: At each timestep t , the action a with the lowest probability π_a^t gets loss 0 and all the remaining actions get a loss of 1. Since $\min_{a \in \mathcal{A}} \pi_a^t \leq \frac{1}{N}$, the cumulative loss L^T of \mathcal{H} will be at least $T(1 - \frac{1}{N})$. On the other hand, there exists an algorithm $g \in \mathcal{G}_{all}$ that achieves a total loss of 0.

- It is not possible to guarantee low regret with respect to the overall optimal sequence of decisions!

Deterministic Algorithm \mathcal{H}

- Consider a deterministic algorithm \mathcal{H} that places all of its probability mass on a single action a^t at each timestep t
- Now consider an algorithm that at each timestep t chooses an action a^t with the lowest cumulative loss $a^t = \arg \min_{a \in \mathcal{A}} L_a^{t-1}$
- The algorithm is also known as **Follow the Leader** (FTL) in the online optimization literature
- If each player in a repeated game uses FTL, we get the **Fictitious play** algorithm
- Unfortunately, we can't guarantee a low total regret of any deterministic algorithm

Bound on the Loss of a Deterministic Algorithm

Theorem

For any deterministic algorithm \mathcal{H} , there exists a sequence of loss vectors for which $L^T = T$ while the best possible action in hindsight achieves $L_{min}^T = \frac{T}{N}$.

- Thus the total regret of any deterministic algorithm grows linearly, i.e. $\mathcal{O}(T)$
- Our goal is to find an algorithm that we call a **regret minimizer** that guarantees a **sub-linear** growth of the total regret, e.g. $\mathcal{O}(\sqrt{T})$ or $\mathcal{O}(\log T)$
- This leads to the average regret approaching 0 as the number of timesteps $T \rightarrow \infty$
- Algorithms with guaranteed sub-linear total regret are sometimes called **no-regret** or **Hannan consistent** algorithms
- Can stochastic algorithms achieve sub-linear total regret?

Lower Bounds for Arbitrary Stochastic Algorithm

Theorem

Consider $T < \log_2 N$. There exists a stochastic generation of losses such that, for any online algorithm \mathcal{H} , we have $\mathbb{E}[L^T] = T/2$ and $L_{min}^T = 0$.

- In other words, the theorem says that we cannot hope to achieve sub-linear regret when the number of timesteps T is small compared to $\log_2 N$

Theorem

Let $N = 2$. There exists a stochastic generation of losses such that for any online algorithm \mathcal{H} we have $\mathbb{E}[L^T - L_{min}^T] = \Omega(\sqrt{T})$.

- The theorem essentially says that even in the simplest scenario when there are only two actions, we cannot hope for regret $o(\sqrt{T})$

Regret Matching

- We would like to have an algorithm whose regret is close to these lower bounds
- There are many stochastic algorithms that achieve sub-linear regret, e.g. Polynomial Weights or Hedge
- One particularly simple and non-parametric algorithm is **Regret Matching**
- The algorithm considers only actions with positive regrets; therefore, we define $R_a^{t,+} = \max(R_a^t, 0)$
- The algorithm chooses all actions with non-zero $R_a^{t,+}$ with probability proportional to their value $R_a^{t,+}$
- $\pi_a^t = \frac{R_a^{t,+}}{\sum_{a'} R_{a'}^{t,+}}$ if $\sum_{a \in \mathcal{A}} R_a^{t,+} > 0$ and $\frac{1}{N}$ otherwise.
- Regret Matching is guaranteed to have regret bounded by $\mathcal{O}(\sqrt{NT})^1$

¹See <http://www.cs.cmu.edu/~ggordon/ggordon.CMU-CALD-05-112.no-regret.pdf> for the proof.

The background image shows a wide, frozen lake or river system from an aerial perspective. The ice is mostly white and light gray, with several dark, winding paths or roads carved through it, likely made by vehicles. These dark paths form a network across the expanse of white ice.

Regret Minimization in Game Theory

Regret Minimization in Normal-Form Games

- Consider repeatedly playing a two-player normal-form game
- Instead of loss vectors, we will now switch to reward vectors
- Each player can employ a regret minimizer and consider its opponent as an adversarial environment
- The reward vector r^t can be computed using the opponent's policy π_{-i}^{t-1} and player i selects its strategy π_i^t using its regret minimizer
- We define the average regret of player i after T timesteps as

$$R_i^T = \frac{1}{T} \max_{a_i \in \mathcal{A}_i} \sum_{t=1}^T (u_i(a_i, \pi_{-i}^t) - u_i(\pi^t))$$

Convergence in General-Sum Games

- In general-sum games, if all players use a no (external) regret algorithm, the empirical distribution of actions converges to a **coarse correlated equilibrium**
- Coarse correlated equilibria are a generalization of correlated equilibria
- A coarse correlated equilibrium is a probability distribution over strategy profiles $a \in \mathcal{A}$, such that if for all players $i \in \mathcal{N}$ and all unilateral deviations $a'_i \in \mathcal{A}_i$, it holds

$$\sum_{a \in \mathcal{A}} p(a) u_i(a) \geq \sum_{a \in \mathcal{A}} p(a) u_i(a'_i, a_{-i})$$

- If all players use a no-internal-regret algorithm, the empirical distribution of actions converges to a **correlated equilibrium**

Convergence in Zero-Sum Games

Theorem

Consider T iterations of no-regret algorithm in a zero-sum game. If both player's average regret is less than ϵ then the **average** strategy profile $\bar{\pi}$ is a 2ϵ -Nash equilibrium.

Proof of Convergence in Zero-Sum Games

- Let π'_1 be an arbitrary strategy of Player 1. Since both players have their average regret lower than ϵ , we have:

$$\frac{1}{T} \sum_t u_1(\pi'_1, \pi_2) \leq \frac{1}{T} \sum_t u_1(\bar{\pi}) + \epsilon$$

- Taking expectation over t gives us

$$u_1(\pi'_1, \bar{\pi}_2) \leq u_1(\bar{\pi}) + \epsilon \tag{1}$$

- Similarly, for Player 2, we have

$$u_2(\bar{\pi}_1, \pi'_2) \leq u_2(\bar{\pi}) + \epsilon$$

Proof of Convergence in Zero-Sum Games

- Now, we will use our assumption that the game is zero-sum. We have $u_2 = -u_1$ which leads to

$$u_1(\bar{\pi}_1, \pi'_2) \geq u_1(\bar{\pi}) - \epsilon \quad (2)$$

- Chaining the two inequalities in Equations 1 and 2, we get

$$u_1(\pi'_1, \bar{\pi}_2) - \epsilon \leq u_1(\bar{\pi}) \leq u_1(\bar{\pi}_1, \pi'_2) + \epsilon$$

Solving Games with Regret Minimization

- We now have another iterative algorithm for solving normal-form games
- We can choose arbitrary regret minimization algorithm, such as regret matching, and let both players play according to the algorithm
- If we choose regret matching, the asymptotic average regret for each player after T iterations is $\mathcal{O}(\frac{1}{\sqrt{T}})$
- When we then take the average strategy for each player, we have $\mathcal{O}(\frac{1}{\sqrt{T}})$ -Nash equilibrium
- For a fixed ϵ , we need $\mathcal{O}(\frac{1}{\epsilon^2})$ iterations of regret minimization

Properties of the Algorithm

- Very easy to implement.
- Each player only needs to remember their cumulative regrets and their average strategy
- Players do not even have to know the payoff matrix – it does not have to be represented in the memory
- If the opponent does not play according to a regret minimization algorithm, we are guaranteed to earn as much as the best response to the opponent's average strategy in the limit

Convergence Notes

- We mentioned that the convergence rate of average regret of $\mathcal{O}(\frac{1}{\sqrt{T}})$ is optimal in the general setting
- However, in zero-sum games both players can cooperate to solve the game; therefore, they can achieve smaller regret and faster convergence
- There is a known algorithm with $\mathcal{O}(\frac{\ln T}{T})$ regret for both players, with the assumption that they don't know the payoff matrix at the beginning² but the iterations are too slow for practical use

²<http://dl.acm.org/citation.cfm?id=2133057>

Week 5 Homework

You can find more detailed descriptions of homework tasks in the GitHub repository.

1. Regret minimization