MIT 6.7220/15.084 — Nonlinear Optimization Tue, Apr 30th 2024

Lecture 17
Online optimization and regret

Instructor: Prof. Gabriele Farina (✉ gfarina@mit.edu)*

Online optimization has been a central topic in machine learning and optimization for the past two decades. The main idea is to design algorithms that can make decisions in an online fashion, in a *nonstationary* environment.

1 Online optimization

Online optimization is an abstraction for repeated decision-making. In particular, at each time step t, we want to select a point $x_t \in \Omega$, where Ω is a specified feasible set. Based on the decision x_t, the algorithm then receives a loss function $f_t : \Omega \to \mathbb{R}$ that reveals the quality of the decision. Both the decision-maker and the environment have memory: the decision-maker can take into account the past losses $f_1, ..., f_{t-1}$ when deciding the next decision $x_t \in \Omega$, and the environment can generate—possibly adversarially—losses f_t based on all decisions $x_1, ..., x_t$ observed so far.

Depending on the specific assumptions regarding how adversarial the environment is, and how much is revealed regarding f_t to the online optimizer (that is, the decision-maker), people have special names for some settings. For example:

- The environment is said to be *stochastic* if the f_t are drawn from an unknown fixed distribution.
- The problem is said to be *bandit* if the decision-maker only observes the loss $f_t(x_t)$ of the decision x_t made at time t, instead of the function f_t itself.
- Sometimes people like to refer to the setting where the decision-maker has full knowledge of the loss functions f_t as *full-information* online optimization.

Overall, we will focus in the next couple of lectures on the full-information, adversarial case. Most of the other settings can often be reduced to this one, including the challenging adversarial bandit setting.

The performance of an online optimization algorithm is typically measured by its regret.

Definition 1.1 (Regret). The regret incurred by a sequence of t decisions $x_1, x_2, ..., x_t \in \Omega$ given the revealed loss functions $f_1, f_2, ..., f_t$ is defined as

$$\mathrm{Reg}_T := \sum_{t=1}^{T} f_t(x_t) - \min_{x \in \Omega} \sum_{t=1}^{T} f_t(x),$$

that is, the difference between the loss accumulated and the minimum loss that could have been accumulated in hindsight by the *best, fixed* decision in Ω.

*These notes are class material that has not undergone formal peer review. The TAs and I are grateful for any reports of typos.

The goal of an online optimization algorithm is to guarantee that the regret is as small as possible. In particular, we are interested in algorithms that have *sublinear* regret as a function of the number of iterations T of the algorithm.

Remark 1.1. Regret can be negative! This is because we are trading off the ability to make *time-dependent* decisions against an unknown loss function, versus making the best *fixed* decision with hindsight.

Other performance metrics have been proposed, such as *dynamic* regret, *switching* regret, *swap* regret, and more. However, the standard notion of regret defined above is (i) the most common, (ii) already nontrivial to keep sublinear, and (iii) already extremely useful in applications.[1] More specifically, as we will see next, the notion of regret defined above is related to equilibrium approximation in game theory.

Today, it is known how to construct online optimization algorithms that achieve sublinear regret in a wide range of settings. Specifically, when Ω is closed and convex and the losses f_t are convex and Lipschitz-continuous, it is well-understood that sublinear regret can be achieved via an *online* generalization of the mirror descent algorithm that we will discuss in the next lecture.

2 Connections to decision-making and game theory

Online optimization has important applications to decision-making and game theory. Techniques based on online optimization are used in a wide range of applications, including reinforcement learning, online advertising, online auctions, and game solving. In particular, landmark AI results such as superhuman performance in poker have been achieved using online optimization techniques.

In this section, we will see how online optimization can be useful in solving nonsequential, simultaneous-action games. In a later class, we will see how to use online optimization to solve imperfect-information sequential games like poker.

2.1 Normal-form games

A normal-form game is a game in which players simultaneously choose actions and receive payoffs. The prototypical example of a normal-form game is rock-paper-scissors.

■ **Actions and payoffs.** In a normal-form game, each player i has a set of actions A_i, and each player receives a payoff that depends on the actions chosen by all players. For this class, we will focus on two-player games. Letting the actions of Player 1 on the rows, and the actions of Player 2 on the columns, we can arrange the payoff obtained by each player $i \in \{1, 2\}$ in a matrix $U_i \in \mathbb{R}^{A_1 \times A_2}$.

Example 2.1. In rock-paper-scissors, we have

$$U_1 := \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}, \qquad U_2 := -U_1.$$

When the payoffs are opposite for the players, that is, $U_2 = -U_1$, the game is aptly called *zero-sum.*

[1]To distinguish the notion of regret defined in Definition 1.1 from these other notions, sometimes people like to refer to the notion in Definition 1.1 using the terms *static* regret (common in the online/reinforcement learning theory literature) or *external* regret (common in the game theory literature).

Definition 2.1 (Zero-sum game). A two-player game is said to be *zero-sum* if $U_2 = -U_1$.

■ **Mixed strategies.** Players can randomize their choice of actions, that is, play according to probability distribution. A mixed strategy for Player 1 is a vector $x \in \Delta(A_1)$, where $\Delta(A_1)$ is the set of probability distributions over A_1. Similarly, a mixed strategy for Player 2 is a vector $y \in \Delta(A_2)$.

Given a choice of mixed strategies for each player, the expected utility for each player is then given by

$$u_1(x,y) = x^\top U_1 y, \qquad u_2(x,y) = x^\top U_2 y.$$

■ **Nash equilibrium.** The Nash equilibrium is a central concept in game theory. It is a strategy profile in which no player has an incentive to deviate from their strategy.

Definition 2.2 (Nash equilibrium in two-player games). A strategy profile (x, y) is a *Nash equilibrium* in a two-player game if

$$x^\top U_1 y = \max_{\hat{x} \in \Delta(A_1)} \hat{x}^\top U_1 y \qquad \text{(Player 1 is playing optimally against Player 2); and}$$

$$x^\top U_2 y = \max_{\hat{y} \in \Delta(A_2)} x^\top U_2 \hat{y} \qquad \text{(Player 2 is playing optimally against Player 1).}$$

A strategy profile (x, y) is an *ε-approximate Nash equilibrium* (or simply *ε-Nash equilibrium*) if

$$\max_{\hat{x} \in \Delta(A_1)} \{\hat{x}^\top U_1 y\} - x^\top U_1 y \le \varepsilon \qquad \text{and} \qquad \max_{\hat{y} \in \Delta(A_2)} \{x^\top U_2 \hat{y}\} - x^\top U_2 y \le \varepsilon.$$

In two-player *zero-sum* games, where by definition $U_2 = -U_1$,

$$x^\top U_2 y = \max_{\hat{y} \in \Delta(A_2)} x^\top U_2 \hat{y} \qquad \text{is equivalent to} \qquad x^\top U_1 y = \min_{\hat{y} \in \Delta(A_2)} x^\top U_1 \hat{y}.$$

Hence, Player 2 is selecting any strategy that minimizes the expected utility of Player 1. No matter the strategy x selected by Player 1, Player 1's expected utility will therefore be $\min_{y \in \Delta(A_2)} x^\top A y$. Since Player 1 is playing optimally, then their strategy should be a solution to

$$\max_{x \in \Delta(A_1)} \left(\min_{y \in \Delta(A_2)} x^\top U_1 y \right).$$

Hence, a Nash equilibrium in a two-player zero-sum game is the best possible strategy x against a maximally strong adversary. In this light, it is perhaps not surprising that in two-player games, Nash equilibrium is the preferred solution concept for designing AI agents playing against top professionals in go, chess, poker, *et cetera*.

2.2 Online optimization in games

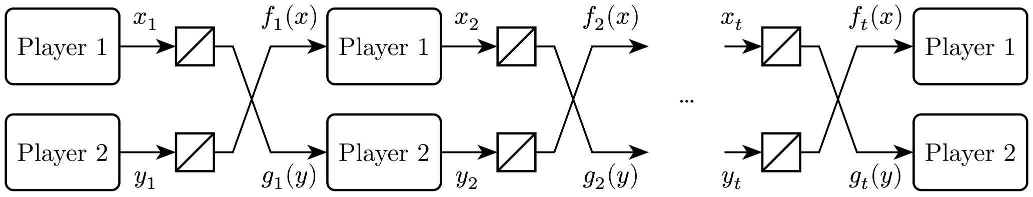
Conceptually, players in games would like to compute randomized strategies that maximize their expected utility given what the other players are playing. However, the learning of all agents makes the problem inherently nonstationary. This is where *online* optimization saves the day.

The fundamental idea is to just apply online optimization from the point of each player. At each time t, each player selects a randomized strategy $x_t \in \Delta(A_1)$, $y_t \in \Delta(A_2)$. Based on the selected strategies, the loss functions that the players receive are then set to the opposite of their expected utility:

$$f_t : \Delta(A_1) \to \mathbb{R}, \qquad f_t(x) := -x^\top U_1 y_t = (-U_1 y_t)^\top x$$

$$g_t : \Delta(A_2) \to \mathbb{R}, \qquad g_t(y) := -x_t^\top U_2 y = (-U_2^\top x_t)^\top y$$

This process, called *self-play*, is repeated for a number of iterations.



As it turns out, there is a deep connection between the regret accumulated by the players in self-play and equilibrium approximation in the game. For two-player zero-sum games in particular, the following strong result holds.

Theorem 2.1. Let

$$\begin{aligned}\mathrm{Reg}_T^{(1)} &:= \sum_{t=1}^T f_t(x_t) - \min_{x\in\Delta(A_1)} \sum_{t=1}^T f_t(x),\\ \mathrm{Reg}_T^{(2)} &:= \sum_{t=1}^T g_t(y_t) - \min_{y\in\Delta(A_2)} \sum_{t=1}^T g_t(y),\end{aligned}$$

be the regrets accumulated by Player 1 and 2, respectively, after playing T rounds of the game in self-play. Then, the *average strategy profile*

$$\left(\overline{x}_T, \overline{y}_T\right) := \left(\frac{1}{T}\sum_{t=1}^T x_t, \frac{1}{T}\sum_{t=1}^T y_t\right) \in \Delta(A_1)\times\Delta(A_2)$$

is an ε-Nash equilibrium of the game, where

$$\varepsilon := \frac{\mathrm{Reg}_T^{(1)} + \mathrm{Reg}_T^{(2)}}{T}.$$

Proof. By definition of f_t and g_t, we have

$$\begin{aligned}\mathrm{Reg}_T^{(1)} &= \sum_{t=1}^T f_t(x_t) - \min_{x\in\Delta(A_1)} \sum_{t=1}^T f_t(x) = -\sum_{t=1}^T x_t^\top U_1 y_t - \min_{x\in\Delta(A_1)}\left\{-\sum_{t=1}^T x^\top U_1 y_t\right\},\\ \mathrm{Reg}_T^{(2)} &= \sum_{t=1}^T g_t(y_t) - \min_{y\in\Delta(A_2)} \sum_{t=1}^T g_t(y) = +\sum_{t=1}^T x_t^\top U_1 y_t - \min_{y\in\Delta(A_2)}\left\{+\sum_{t=1}^T x_t^\top U_1 y\right\},\end{aligned}$$

Hence, by summing the two expressions and dividing by T, we get

$$\begin{aligned}\frac{\mathrm{Reg}_T^{(1)} + \mathrm{Reg}_T^{(2)}}{T} &= -\min_{x\in\Delta(A_1)}\left\{-\frac{1}{T}\sum_{t=1}^T x^\top U_1 y_t\right\} - \min_{y\in\Delta(A_2)}\left\{\frac{1}{T}\sum_{t=1}^T x_t^\top U_1 y\right\}\\ &= -\min_{x\in\Delta(A_1)}\left\{-x^\top U_1 \overline{y}_T\right\} - \min_{y\in\Delta(A_2)}\left\{\overline{x}_T^\top U_1 y\right\}\\ &= \max_{x\in\Delta(A_1)}\left\{x^\top U_1 \overline{y}_T\right\} - \min_{y\in\Delta(A_2)}\left\{\overline{x}_T^\top U_1 y\right\}\\ &= \left(\max_{x\in\Delta(A_1)}\left\{x^\top U_1 \overline{y}_T\right\} - \overline{x}_T^\top U_1 \overline{y}_T\right) + \left(\overline{x}_T^\top U_1 \overline{y}_T - \min_{y\in\Delta(A_2)}\left\{\overline{x}_T^\top U_1 y\right\}\right).\end{aligned}$$

Hence, we must have

$$\max_{\hat{x}\in\Delta(A_1)} \left\{\hat{x}^\top U_1 \overline{y}_T\right\} - \overline{x}_T^\top U_1 \overline{y}_T \le \frac{\mathrm{Reg}_T^{(1)} + \mathrm{Reg}_T^{(2)}}{T}; \qquad \overline{x}_T^\top U_1 \overline{y}_T - \min_{\hat{y}\in\Delta(A_2)} \{\overline{x}_T^\top U_1 \hat{y}\} \le \frac{\mathrm{Reg}_T^{(1)} + \mathrm{Reg}_T^{(2)}}{T},$$

which is the statement. □

Remark 2.1. The previous theorem relies fundamentally on the game being two-player zero-sum. In fact, from the **PPAD**-completeness of Nash equilibrium in more general settings [CDT09; DGP09], we know that finding Nash equilibria beyond two-player zero-sum games is in general computationally hard.

Beyond two-player zero-sum games, regret is known to be linked to several notions of *correlated equilibrium*, a generalization of Nash equilibrium that allows for coordination between players.

Bibliography

[CDT09] X. Chen, X. Deng, and S.-H. Teng, “Settling the complexity of computing two-player Nash equilibria,” *Journal of the ACM (JACM)*, vol. 56, no. 3, pp. 1–57, 2009.

[DGP09] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou, “The complexity of computing a Nash equilibrium,” *Communications of the ACM*, vol. 52, no. 2, pp. 89–97, 2009.