DS 801: Advanced Optimization for Data Science **KAIST, Fall 2024**
Lecture #20: Online Convex Optimization May 20, 2024
Lecturer: Dabeen Lee

1 Outline

In this lecture, we cover

- introduction to online learning,
- online convex optimization,
- online mirror descent.

2 Online Convex Optimization

Online convex optimization (OCO) is an online learning problem, that is to make a sequence of predictions based on the history of past decisions and their results. The framework of OCO is closely related to game theory, statistical learning theory, and stochastic modeling as well as convex optimization. The contents of this section are based on the text of Hazan [Haz16].

Let us provide some applications of the OCO framework.

- (Online spam filtering) We receive emails repeatedly, for each of which we apply an existing spam-filtering system. A spam-filtering system has a list of words and expressions, based on which it can predict whether an email is spam or valid. When an email that is classified as valid by the existing filter turns out to be spam, we have to update the filter so that we can filter similar spam emails later.
- (Online advertisement selection) A web browser selects a collection of online advertisements for its ad slots. The web browser posts a catalog of online ads and observes their popularity from users by the click-through rates. Later, the browser can change its ad selection based on its prediction about user demands.

2.1 Online Binary Classification

Let us consider a mathematical model to establish an email spam filtering system. Recall that we used the support vector machine (SVM) for binary classification. Just to remind you what it was, we find a pair of a coefficient vector w and a right-hand side value b to use the hyperplane $w^\top x = b$ to classify data points. Given a feature vector x, we assign it label $\text{sign}(w^\top x - b)$ where $\text{sign}(c)$ has value 1 if $c \geq 0$ and value -1 if $c < 0$. When a training set of multiple data is available, we can find such a classifier (w, b) by solving a convex optimization problem whose objective is to minimize the hinge loss.

However, in some scenarios, data points dynamically arrive so that we gradually accumulate the data. In such cases, we may adjust our model over time, and the learning process continues. To be more specific, let us consider the online binary classification problem described as follows. An email is represented by its feature vector $x \in \mathbb{R}^d$ and label $y \in \{-1, 1\}$. The feature vector can encode words and expressions written in it, while the label indicates whether the email is spam or not. Let's say that $y = 1$ indicates spam and $y = -1$ indicates valid. For each time slot t, we repeat the following procedure.

- The spam filtering system prepares a classifier (w_t, b_t) based on the past emails represented by $(x_1, y_1), \dots, (x_{t-1}, y_{t-1}) \in \mathbb{R}^d \times \{-1, 1\}$.
- New email with feature vector x_t arrives.
- The spam filter predicts that its label is $\mathrm{sign}(w_t^\top x_t - b)$, while the true label of the email is y_t.
- The spam filter incurs a loss of $\max\{0, 1 - y_t(w_t^\top x_t - b)\}$.

After T emails, the cumulative loss is given by

$$\sum_{t=1}^{T} \max\{0, 1 - y_t(w_t^\top x_t - b)\}.$$

Compared to a best classifier, we incur

$$\sum_{t=1}^{T} \max\{0, 1 - y_t(w_t^\top x_t - b)\} - \min_{(w,b)\in\mathbb{R}^d\times\mathbb{R}} \sum_{t=1}^{T} \max\{0, 1 - y_t(w^\top x_t - b)\}$$

more loss. Denoting the loss function at each time t as

$$f_t(w, b) = \max\{0, 1 - y_t(w^\top x_t - b)\},$$

the excess cumulative loss is rewritten as

$$\sum_{t=1}^{T} f_t(w_t, b_t) - \min_{(w,b)\in\mathbb{R}^d\times\mathbb{R}} \sum_{t=1}^{T} f_t(w, b).$$

Therefore, the online binary classification problem is an instance of online convex optimization where the best fixed decision corresponds to the best spam classifier.

2.2 Adversarial Multi-Armed Bandits

Suppose that we have d slot machines (or bandits). Then, at each t, the player chooses which slot machine to play. Here, let $i_t \in \{1, \dots, d\}$ be the machine that the player chooses at time t. Then the reward of playing machine $i \in \{1, \dots, d\}$ at time t is given by $r_{i,t}$, which is revealed only after a play. Then we may compare the player's cumulative reward against the total reward of the best slot machine as follows.

$$\max_{i\in\{1,\dots,d\}} \sum_{t=1}^{T} r_{t,i} - \sum_{t=1}^{T} r_{t,i_t}.$$

3 Regret Minimization for Online Convex Optimization

Online advertisement selection, email spam filter, and multi-armed bandits involve sequential decision-making that depends on interactions between decisions made by the decision-maker and data provided by the environment. As mentioned earlier, this process is called online learning or online optimization in the sense that the learning process and the optimization task proceed based on the history of information accumulated so far. As opposed to online learning and online optimization, offline learning and offline optimization assume that complete information is available.

The following gives the list of main components.

1. (A sequence of convex loss functions) We are given convex loss functions $f_1, \dots, f_T$ where T is the length of time horizon. The functions are revealed one at a time sequentially.
2. (Sequential decisions) At each time step t, we get to choose a decision/prediction x_t before the function f_t for the time step is revealed. In other words, the function f_t is unknown to the decision maker when making a decision.
3. (Bounded domain) The set of available decisions (the feasible set), denoted C, is bounded and convex.

Then we compute the accumulated losses incurred over the T time steps.

$$\sum_{t=1}^{T} f_t(x_t).$$

This is indeed an online learning problem because, to make a new decision x_{t+1}, we may use the history of the past decisions and their corresponding losses

$$x_1,\ f_1(x_1),\ x_2,\ f_2(x_2),\ \dots,\ x_t,\ f_t(x_t)$$

although the loss function f_{t+1} for time step $t+1$ is not yet given.

3.1 Performance Metric: the Notion of Regret

Let $\mathcal{A}$ be an algoriothm for online convex optimization, and let $x_1^{\mathcal{A}}, \dots, x_T^{\mathcal{A}}$ denote the decisions made by algorithm $\mathcal{A}$. We have defined the cumulative loss, minimizing which is our goal basically. At the same time, to measure how close algorithm $\mathcal{A}$ is to being optimal, we compare the cumulative loss of algorithm $\mathcal{A}$ against the cumulative loss of the best fixed decision. To be more precise, we consider the following notion of *regret*.

$$\mathrm{Regret}_T(\mathcal{A}) = \sum_{t=1}^{T} f_t(x_t^{\mathcal{A}}) - \min_{x\in C} \sum_{t=1}^{T} f_t(x).$$

Here, setting the benchmark as a single best decision is motivated by email spam filter for which we need to find the most effective spam filtering system and multi-armed bandits in which the goal is to find the most profitable slot machine.

We focus on developing algorithms that minimize the regret. By taking a sequence of actions to minimize the regret, we learn and get close to the action of the best decision maker.

Our goal is to design an algorithm $\mathcal{A}$ whose regret is sublinear in T, which means that $\mathrm{Regret}_T(\mathcal{A}) = o(T)$. What does this indicate? We look at the time averaged regret.

$$\frac{1}{T}\sum_{t=1}^{T} f_t(x_t^{\mathcal{A}}) - \min_{x\in C} \frac{1}{T}\sum_{t=1}^{T} f_t(x) = \frac{\mathrm{Regret}_T(\mathcal{A})}{T} = o(1).$$

In particular, in the offine setting where $f_1 = \cdots = f_T = f$, the statement is equivalent to

$$\frac{1}{T}\sum_{t=1}^{T} f(x_t^{\mathcal{A}}) - \min_{x\in C} f(x) = \frac{\mathrm{Regret}_T(\mathcal{A})}{T} = o(1).$$

Hence, a sublinear regret means that the time averaged optimality gap goes to 0 as T increases.

3.2 Online Gradient Descent

There is a simple algorithm for online convex optimization that minimizes regret. In fact, a modification of gradient descent works for the online setting, and it is called online gradient descent.

Algorithm 1 Online Gradient Descent (OGD)

```
  Initialize $x_1 \in C$.
  for $t = 1, \dots, T$ do
      Observe $f_t(x_t)$ and obtain $g_t \in \partial f_t(x_t)$.
      Obtain $x_{t+1} = \mathrm{proj}_C \{x_t - \eta_t g_t\}$ for a step size $\eta_t > 0$.
  end for
```

The only distinction compared to the subgradient method for the offline setting is that we obtain a subgradient from the subdifferentials $\partial f_t(x_t)$ of functions f_t that are sequentially revealed. This simple algorithm does achieve an aymptotically optimal regret.

Theorem 20.1. *Let $f_1, \dots, f_T$ be an arbitrary sequence of convex loss functions satisfying $\|g_t\|_2 \leq L$ for any $g_t \in \partial f_t(x)$ for every $x \in \mathbb{R}^d$ and $t \geq 1$. Then online gradient descent given by Algorithm 1 with step sizes $\eta_t = R/(L\sqrt{t})$ where $R^2 = \sup_{x,y \in C} \|x - y\|_2^2$ satisfies*

$$\sum_{t=1}^{T} f_t(x_t) - \min_{x \in C} \sum_{t=1}^{T} f_t(x) \leq \frac{3}{2} LR\sqrt{T}.$$

Proof. The analysis of online gradient descent is quite similar to that of gradient descent. Let $x^* \in \mathrm{argmin}_{x \in C} \sum_{t=1}^{T} f_t(x)$. Note that

$$\begin{aligned} \|x_{t+1} - x^*\|_2^2 &\leq \|x_t - \eta_t g_t - x^*\|_2^2 \\ &= \|x_t - x^*\|_2^2 + \eta_t^2 \|g_t\|_2^2 - 2\eta_t g_t^\top (x_t - x^*) \\ &\leq \|x_t - x^*\|_2^2 + \eta_t^2 L^2 - 2\eta_t (f_t(x_t) - f_t(x^*)) \end{aligned}$$

where the first inequality is due to the contraction property of the projection operator and the second inequality is due to the convexity of f_t. Then it follows that

$$f_t(x_t) - f_t(x^*) \leq \frac{1}{2\eta_t} \left(\|x_t - x^*\|_2^2 - \|x_{t+1} - x^*\|_2^2 \right) + \frac{\eta_t}{2} L^2.$$

Adding up these inequalities for $t = 1, \dots, T$, we obtain

$$\begin{aligned} \sum_{t=1}^{T} f_t(x_t) - \sum_{t=1}^{T} f_t(x^*) &\leq \sum_{t=1}^{T} \frac{1}{2\eta_t} \left(\|x_t - x^*\|_2^2 - \|x_{t+1} - x^*\|_2^2 \right) + \sum_{t=1}^{T} \frac{\eta_t}{2} L^2 \\ &\leq \sum_{t=1}^{T} \|x_t - x^*\|_2^2 \left(\frac{1}{2\eta_t} - \frac{1}{2\eta_{t-1}} \right) + \frac{L^2}{2} \sum_{t=1}^{T} \eta_t \\ &\leq \frac{R^2}{2} \sum_{t=1}^{T} \left(\frac{1}{\eta_t} - \frac{1}{\eta_{t-1}} \right) + \frac{L^2}{2} \sum_{t=1}^{T} \eta_t \\ &= \frac{R^2}{2} \cdot \frac{1}{\eta_T} + \frac{L^2}{2} \sum_{t=1}^{T} \frac{R}{L\sqrt{t}} \\ &\leq \frac{3}{2} RL\sqrt{T} \end{aligned}$$

where we set $1/\eta_0$ to be 0, the second inequality is because $\|x_{t+1} - x^*\|_2^2 \geq 0$, and the last inequality is because $\sum_{t=1}^T 1/\sqrt{t} \leq 2\sqrt{T}$. □

Therefore, for Lipschitz continuous functions, OGD achieves the regret of $O(\sqrt{T})$. Can we do better than this?

Theorem 20.2. *Any algorithm for online convex optimization incurs $\Omega(LR\sqrt{T})$ regret in the worst case. The same statement holds even when the loss functions are i.i.d. with a fixed stationary distribution.*

For strongly convex and Lipschitz continuous functions, we can achieve a logarithmic regret!

Theorem 20.3. *Let $f_1, \ldots, f_T$ be an arbitrary sequence of convex loss functions satisfying $\|g_t\|_2 \leq L$ for any $g_t \in \partial f_t(x)$ for every $x \in \mathbb{R}^d$ and $t \geq 1$. Moreover, $f_1, \ldots, f_T$ are α-strongly convex with respect to the ℓ_2 norm. Then online gradient descent given by Algorithm 1 with step sizes $\eta_t = 1/(\alpha t)$ satisfies*

$$\sum_{t=1}^T f_t(x_t) - \min_{x \in C} \sum_{t=1}^T f_t(x) \leq \frac{L^2}{2\alpha}(1 + \log T).$$

4 Online Mirror Descent

We learned the online gradient descent (OGD) algorithm that proceeds with the update rule

$$x_{t+1} = \mathrm{proj}_C \{x_t - \eta_t g_t\}$$

where g_t is a subgradient of f_t at x_t. Note that

$$\|x - x_t + \eta_t g_t\|_2^2 = 2\eta_t \left(g_t^\top x + \frac{1}{2\eta_t} \|x - x_t\|_2^2 \right) - 2\eta_t g_t^\top x_t + \eta_t^2 \|g_t\|_2^2.$$

This implies that

$$\begin{aligned} x_{t+1} &= \mathrm{proj}_C \{x_t - \eta_t g_t\} \\ &= \mathrm{argmin}_{x \in C} \|x - x_t + \eta_t g_t\|_2^2 \\ &= \mathrm{argmin}_{x \in C} \left\{ g_t^\top x + \frac{1}{2\eta_t} \|x - x_t\|_2^2 \right\}. \end{aligned}$$

Recall that $f_t(x_t) + g_t^\top x$ is a first-order approximation of f_t at x_t. Moreover, the quadratic term $\|x - x_t\|_2^2 / 2\eta_t$ encourages to choose a solution nearby x_t. Hence, the choice of x_{t+1} is given by a tradeoff between minimizing the first-order approximation and choosing a solution nearby x_t. Here, the distance between x and x_t is measured by the ℓ_2 norm. Then the regret upper bound is given by

$$\frac{3}{2} LR\sqrt{T}$$

where L is a global upper bound on the Lipschitz constants of the loss functions and $R^2 = \sup_{x,y \in C} \|x - y\|_2^2$.

Let us consider the case where

- loss functions $f_1, \ldots, f_T$ are L-Lipschitz in the ℓ_1 norm,

- the domain C is given by

$$C = \left\{ x \in \mathbb{R}^d_+ : \ x_1 + \cdots + x_d = \frac{R}{2} \right\}.$$

Note that

$$\sup_{x,y \in C} \|x-y\|_1^2 = R^2 \quad \text{and} \quad \sup_{x,y\in C} \|x-y\|_2^2 = \frac{1}{2}R^2.$$

As the loss functions are L-Lipschitz in the ℓ_1 norm, it follows that

$$\|\nabla f_t(x)\|_\infty \leq L$$

for $x \in C$ and $t = 1, \ldots, T$. Here, we have

$$\|\nabla f_t(x)\|_2 \leq \sqrt{d}\|\nabla f_t(x)\|_\infty \leq L\sqrt{d}$$

for $x \in C$ and $t = 1, \ldots, T$. In fact, it can be that the Lipschitz constant of f_t in the ℓ_2 norm can blow up by a factor of $\sqrt{d}$. Then, online gradient descent may incur a regret of order

$$O(LR\sqrt{dT}).$$

Here, we have the additional factor of $\sqrt{d}$. Again, this is due to the observation that the upper bound on the Lipschitz constants of loss functions has become $L\sqrt{d}$, not L. Perhaps, measuring the Lipschitz constants of loss functions in the ℓ_2-norm is not the best idea.

Recall that the update rule of OGD is

$$x_{t+1} = \mathrm{argmin}_{x \in C} \left\{ g_t^\top x + \frac{1}{2\eta_t}\|x - x_t\|_2^2 \right\}.$$

Here, the distance between x and x_t is captured by the ℓ_2 distance between them. Instead, we may consider the notion of **Bregman divergence**. To define it, we take a strongly convex function ψ with respect to a norm $\|\cdot\|$. Then the Bregman divergence of p and q with respect to ψ is given by

$$D_\psi(p,q) = \psi(p) - \psi(q) - \nabla\psi(q)^\top(p-q).$$

Example 20.4. Note that

$$\psi(x) = \frac{1}{2}\|x\|_2^2$$

is strongly convex in the ℓ_2 norm, and the corresponding Bregman divergence is given by

$$D_\psi(p,q) = \frac{1}{2}\|p-q\|_2^2.$$

Example 20.5. Consider

$$\psi(x) = \sum_{i=1}^d x_i \log x_i,$$

which is strongly convex over

$$C = \{x \in \mathbb{R}^d_+ : \ \|x\|_1 = 1\}$$

in the ℓ_1 norm. Moreover, the corresponding Bregman divergence is given by

$$D_\psi(p,q) = \sum_{i=1}^d p_i \log\frac{p_i}{q_i} = KL(p,q)$$

for $p, q \in C$. **Pinsker's inequality** states that

$$KL(p,q) \geq \frac{1}{2}\|p-q\|_1^2.$$

The **Online Mirror Descent (OMD)** algorithm runs with the update rule

$$x_{t+1} = \mathrm{argmin}_{x\in C} \left\{ g_t^\top x + \frac{1}{\eta_t} D_\psi(x, x_t) \right\}.$$

Theorem 20.6. *Let $f_1, \dots, f_T$ be an arbitrary sequence of convex loss functions that are L-Lipschitz in a norm $\|\cdot\|$. Assume that the Bregman divergence D_ψ satisfies*

$$D_\psi(x, y) \geq \frac{1}{2}\|x - y\|^2$$

for any $x, y \in C$. Moreover, $R^2 = \sup_{x,y\in C} D_\psi(x, y)$. Then online mirror descent with step sizes $\eta_t = R/(L\sqrt{t})$ guarantees that

$$\sum_{t=1}^{T} f_t(x_t) - \min_{x\in C} \sum_{t=1}^{T} f_t(x) = O\left(LR\sqrt{T}\right).$$

References

[Haz16] Elad Hazan. Introduction to online convex optimization. *Found. Trends Optim.*, 2(3–4):157–325, aug 2016. 2