Khoury College of Computer Sciences
Northeastern University
CS 7870: Seminar in Theoretical Computer Science

Fall 2023
10 November 2023
Scribe: Shashank Manjunath

**Lecture 18 Outline:**

- Hedge Algorithm
- Convex Optimization
- Gradient Descent

In this lecture, we continue our study of online learning and online convex optimization. We build on the learning with expert advice problem studied in the previous lecture and present the Hedge algorithm for general loss values. We then introduce some definitions and preliminaries for convex optimization, including the gradient descent algorithm. The lecture is largely based on [Haz16, Chapter 2].

# 1 The Hedge Algorithm

We consider the following framework for online learning. We have $N$ experts and express the loss function at time $t$ with $\ell_t$, where $\ell_t(i)$ indicates the loss of expert $i$ at time $t$. For this lecture, we assume that $\ell_t(i) \in [0, 1]$. (This generalizes the $\{0, 1\}$ case studied last class; more general loss values can also be handled.) At each time step, we incur the loss of one selected expert, and measure our performance in the form of regret:

$$\text{Regret}(T) = \max_i \left\{ \sum_{t=1}^{T} \ell_t(A_t) - \sum_{t=1}^{T} \ell_t(i) \right\}$$

This can alternatively be formulated as:

$$\text{Regret}(T) = \sum_{t=1}^{T} \ell_t(A_t) - \min_i \left\{ \sum_{t=1}^{T} \ell_t(i) \right\}$$

We wish to study how Regret varies as a function of $t$ under various algorithms. A trivial upper bound for Regret is $T$ since the loss at each step is bounded by 1. At a minimum, a low-regret algorithm should achieve $o(T)$. The Hedge algorithm achieves an $O(\sqrt{T \log N})$ bound on Regret. The general idea of the Hedge algorithm is to implement a weighted majority of experts. In the following, we use $\exp(x)$ to represent $e^x$.

Note the weight update:

$$w_{t+1}(i) = w_t(i) \exp(-\varepsilon \ell_t(i)) \tag{1}$$

Additionally, note the following useful inequalities:

**Algorithm 1:** Hedge Algorithm

```
Data: Fix $\varepsilon$. Set $w(i) = 1 \, \forall i$
for $t = 1, \cdots, T$ do
    Select expert $i_t = i$ with probability $x_t(i) = \frac{w_t(i)}{\sum_j w_t(j)}$
    Incur loss of $\ell_t(i_t)$
    Update weights: $w_{t+1}(i) = w_t(i) \exp(-\varepsilon \ell_t(i))$
Return weights $w_T$
```

$$\exp(-x) \geq 1 - x \tag{2}$$

$$\exp(-x) \leq 1 - x + \frac{x^2}{2} \tag{3}$$

We now proceed with the analysis of the regret of the Hedge algorithm.

**Theorem 1.** The Regret of the Hedge algorithm is bounded as follows:

$$\text{Regret}(T) \leq \frac{\ln(N)}{\varepsilon} + \frac{\varepsilon}{2} \sum_{t=1}^{T} x_t^\top \ell_t^2 \tag{4}$$

*Proof.* Define $\phi_t = \sum_i w_t(i)$. Note that $\phi_1 = \sum_i 1 = N$. Furthermore, by definition, we have

$$w_t(i) = x_t(i) \sum_i w_t(i) = x_t(i) \phi_t.$$

Thus,

$$\begin{aligned}
\phi_{t+1} &= \phi_t \left( \sum_i x_t(i) \exp(-\varepsilon \ell_t(0)) \right) \\
\phi_{t+1} &\leq \phi_t \left( \sum_i x_t(i) \left( 1 - \varepsilon \ell_t(i) + \frac{\varepsilon^2 \ell_t(i)^2}{2} \right) \right)
\end{aligned}$$

by the expansion of $\exp(-\varepsilon \ell_t(i))$ using Equation (3).

$$\begin{aligned}
\phi_{t+1} &\leq \phi_t \left( \sum_i x_t(i) - \varepsilon \sum_i x_t(i) \ell_t(i) + \frac{\varepsilon^2 \sum x_t(i) \ell_t(i)^2}{2} \right) \\
&\leq \phi_t \left( 1 - \varepsilon x_t^\top \ell_t + \frac{\varepsilon^2}{2} x_t^\top \ell_t^2 \right) \\
&\leq \phi_t \exp\left( -\varepsilon x_t^\top \ell_t + \frac{\varepsilon^2}{2} x_t^\top \ell_t^2 \right).
\end{aligned}$$

(For the second inequality, we use $\sum_i x_t(i) = 1$.) Therefore,

$$\begin{aligned}\phi_T &\leq \phi_1 \exp\left(-\varepsilon \sum_{t=1}^{T} x_t^\top \ell_t + \frac{\varepsilon^2}{2} \sum_{t=1}^{T} x_t^\top \ell_t^2\right) \\ &\leq N \exp\left(-\varepsilon \sum_{t=1}^{T} x_t^\top \ell_t + \frac{\varepsilon^2}{2} \sum_{t=1}^{T} x_t^\top \ell_t^2\right)\end{aligned}$$

Fix any expert $i$. By definition of $\phi_T$ and $w_T(i)$, we have

$$\phi_T \geq \exp\left(-\varepsilon \sum_{t=1}^{T} \ell_t(i)\right).$$

We thus obtain

$$-\varepsilon \sum_{t=1}^{T} \ell_t(i) \leq \ln(N) - \varepsilon \sum_{t=1}^{T} x_t^\top \ell_t + \frac{\varepsilon^2}{2} \sum_{t=1}^{T} x_t^\top \ell_t^2 \therefore \sum_{t=1}^{T} x_t^\top \ell_t(i)$$

Rearranging, we obatin the following upper bound on the expected loss of the Hedge algorithm.

$$\sum_{t=1}^{T} \ell_t(i) + \frac{\varepsilon}{2} \sum_{t=1}^{T} x_t^\top \ell_t^2 + \frac{\ln(N)}{\varepsilon},$$

which yields the desired upper bound on the Regret of the Heade algorithm. □

We want to set $\varepsilon$ so as to minimize regret. If $\ell_t(i) \in [0,1] \, \forall \, i$, then we have:

$$\sum_t x_t^\top \ell_t \leq \sum_t \ell_t(i) + \frac{\varepsilon T}{2} + \frac{\ln(N)}{\varepsilon}$$

To minimize the regret, we need to set $\varepsilon$ so as to balance the two terms $\frac{\varepsilon T}{2}$ and $\frac{\ln(N)}{\varepsilon}$. This is attained by setting

$$\varepsilon = \sqrt{\frac{2\ln(N)}{T}}$$

We thus obtain that the Regret of the Hedge algorithm is $O(\sqrt{T \ln(N)}$.

## 2 Convex Optimization and Gradient Descent

In the general setting of online convex optimization, we are given a convex body $K$, which is bounded and closed.

**Definition 1** (Convex Body). A body $K$ is convex if for all $x, y \in K$ and $\alpha \in [0,1]$, $\alpha x + (1-\alpha)y \in K$.

We want to optimize a convex function $f : \mathbb{R}^d \to \mathbb{R}$ on a convex body.

**Definition 2** (Convex Function)**.** A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if for all $\alpha \in [0,1]$ and $x, y \in K$, $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$

Consider $t = 1, \cdots, T : f_t$. We want to return the $x_t$ that minimizes $\sum\limits_{t=1}^{T} f_t(x_t)$. Online gradient descent solves this problem. To start, we consider the standard gradient descent algorithm.

Gradient descent works by moving $x$ in the direction opposite the derivative of $f(x)$ for some step size $\eta$. For a convex function this will eventually get close to the minimum. In particular, the aim of gradient descent for unconstrained convex optimization is to find an $x$ that minimizes $f(x)$ where $f$ is some convex function.

**Algorithm 2:** Gradient Descent for Unconstrained Convex Optimization

```
Set $x_1$ arbitrary
for $t = 1, \cdots, T$ do
    $x_{t+1} = x_t - \eta_t \nabla_t f(x_t)$
Return $x = \text{argmin}_{x_t} f(x_t)$
```

For this algorithm, it is not obvious how to set $\eta_t$, and different settings of $\eta_t$ have been proposed and adopted for different optimization objectives. The convergence time also depends on the initial $x_1$. If $\eta_t$ is set to $\eta$ for all $t$, it takes $\frac{|x_1 - x^\star|}{\eta}$ steps to reach $x^\star$, the minimum. We provide some definitions associated with convexity below, which are used extensively in convex optimization studies. We will only discuss these sparingly.

**Definition 3** (Multidimensional Convexity)**.** Given $f : \mathbb{R}^d \to \mathbb{R}^m$ and $x, y \in \mathbb{R}^d$, $f$ is convex if:

$$f(y) - f(x) \geq \nabla f(x)^\top (y - x)$$

**Definition 4** ($\alpha$-Strongly Convex)**.** Given $f : \mathbb{R}^d \to \mathbb{R}^m$ and $x, y \in \mathbb{R}^d$, $f$ is $\alpha$-Strongly convex if:

$$f(y) - f(x) \geq \nabla f(x)^\top (y - x) + \frac{\alpha}{2} \|y - x\|^2$$

**Definition 5** ($\beta$-Smooth)**.** Given $f : \mathbb{R}^d \to \mathbb{R}^m$ and $x, y \in \mathbb{R}^d$, $f$ is $\beta$-Smooth if it is convex and obeys:

$$f(y) - f(x) \leq \nabla f(x)^\top (y - x) + \frac{\beta}{2} \|y - x\|^2$$

**Definition 6** (Well-conditioned)**.** A function $f : \mathbb{R}^d \to \mathbb{R}^m$ is well-conditioned if it is $\alpha$-strongly convex and $\beta$-smooth, with $\alpha \leq \beta$.

$$\gamma = \frac{\alpha}{\beta}$$

is known as the condition number.

## References

[Haz16] Elad Hazan. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.