Lecture 1

Instructor: Haipeng Luo

1 Online Learning: Examples and Models

Below is a list of examples of online learning:

- spam detection (online classification/regression): At each time $t = 1, 2, \ldots$
  - receive an email $x_t \in \mathbb{R}^d$;
  - predict whether it is a spam $\hat{y}_t \in \{-1, +1\}$;
  - see its true label $y_t \in \{-1, +1\}$.
- sequential investment (universal portfolio): Start with capital W_1. At each day $t = 1, 2, \ldots$
  - decide $p_t \in \Delta(N) \stackrel{\text{def}}{=} \{p \in \mathbb{R}_+^N : \sum_{i=1}^N p(i) = 1\}$ and invest $W_t p_t(i)$ on asset i;
  - at the end of the day observe relative prices $r_t \in \mathbb{R}^N$ and arrive at total capital $W_{t+1} = W_t \langle p_t, r_t \rangle$.
- aggregating weather prediction (the expert problem): At each day $t = 1, 2, \ldots$
  - obtain temperature predictions from N experts/models;
  - make the final prediction by randomly following an expert according to $p_t \in \Delta(N)$;
  - on the next day observe the loss of each model $\ell_t \in [0, 1]^N$.
- product recommendation (multi-armed bandits): At each time $t = 1, 2, \ldots$
  - randomly recommend one of the K products a to a customer visiting the website;
  - observe the loss of this product $\ell_t(a)$ (e.g. 0 if clicked, 1 otherwise), but not the losses for the other products.
- multiple-product recommendation (combinatorial bandits): At each time $t = 1, 2, \ldots$
  - randomly recommend k of the K products to a customer visiting the website;
  - observe the losses of the k recommended products but not the other ones.
- personalized product recommendation (contextual bandits): Given N policies $\pi^1, \ldots, \pi^N$, each of which is a mapping from $\mathcal{X}$ to $[K]$. At each time $t = 1, 2, \ldots$
  - observe the contextual information $x_t \in \mathcal{X}$ of a customer (e.g. gender, IP address, purchase history, etc);
  - randomly select one of the N policies π_t and recommend product $\pi_t(x_t)$;
  - observe the loss of this product $\ell_t(\pi_t(x_t))$ but not the other ones.

All of these problems can be (essentially) captured by a learning model called *Online Convex Optimization (OCO)*, first proposed by Zinkevich [2003]. OCO can be viewed as a game between a learner/player and an environment/adversary. Before the game starts, a fixed compact convex set Ω is given to the player as the action space. The game then proceeds for T rounds for some integer T. At each round $t = 1, \ldots, T$,

1. the player first picks a point $w_t \in \Omega$;
2. the environment then picks a convex loss function $f_t : \Omega \to [0, 1]$;
3. the player suffers loss $f_t(w_t)$, and observes some information about f_t.

Depending on the power of the environment, there are several possible settings:

- stochastic setting: $f_1, \ldots, f_T$ are i.i.d samples of a fixed distribution;
- oblivious adversary: $f_1, \ldots, f_T$ are arbitrary, but decided before the game starts (i.e. independent of the player's actions);
- non-oblivious/adaptive adversary: for each t, f_t depends on $w_1, \ldots, w_t$.

Depending on the feedback model, there are also several possible settings:

- full information setting: player observes f_t (or sometimes just (sub)gradient $\nabla f_t(w_t)$);
- bandit setting: player observes only $f_t(w_t)$;
- other partial information settings.

The table below summarizes how OCO captures different kinds of online learning problems.

Problems	Ω	f_t
linear classification linear regression	e.g. $\{w : \|w\|_2 \le 1\}$	$f_t(w) = \ell(\langle w, x_t \rangle, y_t)$, e.g. logistic loss: $\ell(\hat{y}, y) = \ln(1 + e^{-\hat{y}y})$ or square loss: $\ell(\hat{y}, y) = (\hat{y} - y)^2$
universal portfolio	$\Delta(N)$	$f_t(p) = -\ln(\langle p, r_t \rangle)$ (note the unboundedness)
the expert problem	$\Delta(N)$	$f_t(p) = \langle p, \ell_t \rangle$
multi-armed bandits	$\Delta(K)$	$f_t(p) = \langle p, \ell_t \rangle$ (note the feedback model)
combinatorial bandits	$\left\{w = \sum_{j=1}^{M} p(j) v_j \mid p \in \Delta(M)\right\}$ for some $v_1, \ldots, v_M \in \{0,1\}^K$. e.g. $\{w \in [0,1]^K : \sum_{i=1}^{K} w(i) = k\}$	$f_t(w) = \langle w, \ell_t \rangle$ (note the feedback model)
contextual bandits	$\Delta(N)$	$f_t(w) = \sum_{j=1}^{N} w(j) \ell_t(\pi^j(x_t))$ (note the feedback model and the loss structure among policies)

The classic goal of OCO is to minimize the player's regret against the best fixed action in hindsight:

$$\mathcal{R}_T = \sum_{t=1}^{T} f_t(w_t) - \min_{w \in \Omega} \sum_{t=1}^{T} f_t(w).$$

If $\mathcal{R}_T = o(T)$, then it implies that $\lim_{T \to \infty} \mathcal{R}(T)/T = 0$ and thus on average the player is doing almost as well as the best fixed action in hindsight, which is a pretty strong guarantee. Beside minimizing this regret measurement, there are many more harder objectives in OCO that we will cover later.

2 Connection to Statistical Learning

Statistical learning is a classic learning model. Here we explore the connections and differences between statistical learning and online learning.

In statistical learning, a set of training examples $z_1, \ldots, z_T \in \mathcal{Z}$ is given to the learner where each example z_t is an i.i.d. sample of some unknown distribution $\mathcal{D}$. Based on these training examples, the learner outputs an action $w(z_1, \ldots, z_T) \in \Omega$ for some compact convex set Ω. For some loss function $\ell : \Omega \times \mathcal{Z} \to [0,1]$, the training error of the learner is defined as $\frac{1}{T} \sum_{t=1}^{T} \ell(w(z_1, \ldots, z_T), z_t)$ while the generalization error is defined as $\mathbb{E}_{z \sim \mathcal{D}} \ell(w(z_1, \ldots, z_T), z)$. Note that the generalization error is a random variable with respect to the randomness of the training set, and the goal of the learner is to have small generalization error with high probability.

As one can see, distributional assumptions are built in the definition of statistical learning. On the other hand, online learning does not necessarily assume that data is from some fixed distribution, which makes it much more suitable for dealing with time-varying environments. In fact, even if the data is entirely adversarial, which is indeed the case for applications such as spam detection, meaningful and strong guarantees can still be derived for online learning as we will see soon.

Another key advantage is that online learning algorithms are usually more memory-efficient, in the sense that they usually do not need to store data from the past. That is, at each round, the new data is used to update the current states of the algorithm and then discarded. On the other hand, most statistical learning algorithms require storing the training set and touching each example multiple times.

Moreover, it can in fact be shown that online learning is strictly harder than statistical learning in the sense that a full information online learning algorithm can be used to solve statistical learning. The reduction is as follows [Cesa-Bianchi et al., 2004]:

Algorithm 1: Online-to-batch reduction

Input: training set $\{z_1, \dots, z_T\}$, an online learning algorithm $\mathcal{A}$ with action space Ω
for $t = 1, \dots, T$ **do**
  let w_t be the output of $\mathcal{A}$ for this round
  feed $\mathcal{A}$ with loss function $f_t(w) = \ell(w, z_t)$
Output: $\bar{w} = \frac{1}{T}\sum_{t=1}^{T} w_t$.

One can show the following:

Theorem 1. *If $w \to \mathbb{E}_{z\sim\mathcal{D}}\,\ell(w, z)$ is convex, then with probability at least $1-\delta$, the generalization error of the output of Algorithm 1 satisfies*

$$\mathbb{E}_{z\sim\mathcal{D}}\,\ell(\bar{w}, z) \le \mathbb{E}_{z\sim\mathcal{D}}\,\ell(w^\star, z) + \frac{\mathcal{R}_T}{T} + 2\sqrt{\frac{2\ln(2/\delta)}{T}}$$

where $w^\star \in \operatorname{argmin}_{w\in\Omega} \mathbb{E}_{z\sim\mathcal{D}}\,\ell(w, z)$ and $\mathcal{R}_T$ is the regret of the $\mathcal{A}$ after T rounds.

To prove the theorem, we first state the following two important concentration results in probability theory which will be used extensively in the rest of the course.

Lemma 1 (Hoeffding's inequality)**.** *Let $X_1, \dots, X_T \in [-B, B]$ for some $B > 0$ be independent random variables such that $\mathbb{E}[X_t] = 0$ for all $t \in [T]$, then we have for all $\delta \in (0, 1)$,*

$$\Pr\left(\sum_{t=1}^{T} X_t \ge B\sqrt{2T\ln\tfrac{1}{\delta}}\right) \le \delta.$$

Lemma 2 (Azuma's inequality)**.** *Let $X_1, \dots, X_T \in [-B, B]$ for some $B > 0$ be a martingale difference sequence (i.e. $\mathbb{E}[X_t | X_{t-1}, \dots, X_1] = 0$ for all $t \in [T]$), then we have for all $\epsilon > 0$,*

$$\Pr\left(\sum_{t=1}^{T} X_t \ge B\sqrt{2T\ln\tfrac{1}{\delta}}\right) \le \delta.$$

Proof of Theorem 1. With probability at least $1-\delta$, we have

$$\begin{aligned}
\mathbb{E}_{z\sim\mathcal{D}}\,\ell(\bar{w}, z) &= \mathbb{E}_{z\sim\mathcal{D}}\,\ell\left(\frac{1}{T}\sum_{t=1}^{T} w_t, z\right) \\
&\le \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}_{z\sim\mathcal{D}}\,\ell(w_t, z) && \text{(Jensen's inequality)} \\
&= \frac{1}{T}\sum_{t=1}^{T} \ell(w_t, z_t) + \sqrt{\frac{2\ln(2/\delta)}{T}} \\
& && \text{(Azuma's inequality with } X_t = \mathbb{E}_{z\sim\mathcal{D}}\,\ell(w_t, z) - \ell(w_t, z_t))
\end{aligned}$$

$$
\begin{aligned}
&= \min_{w \in \Omega} \frac{1}{T} \sum_{t=1}^{T} \ell(w, z_t) + \frac{\mathcal{R}_T}{T} + \sqrt{\frac{2\ln(2/\delta)}{T}} && \text{(by definition of regret)} \\
&\leq \frac{1}{T} \sum_{t=1}^{T} \ell(w^\star, z_t) + \frac{\mathcal{R}_T}{T} + \sqrt{\frac{2\ln(2/\delta)}{T}} \\
&\leq \mathbb{E}_{z \sim \mathcal{D}}\, \ell(w^\star, z) + \frac{\mathcal{R}_T}{T} + 2\sqrt{\frac{2\ln(2/\delta)}{T}}, \\
& && \text{(Hoeffding's inequality with } X_t = \ell(w^\star, z_t) - \mathbb{E}_{z \sim \mathcal{D}}\, \ell(w^\star, z))
\end{aligned}
$$

which completes the proof. □

For many problems we will show that $\mathcal{R}_T = \mathcal{O}(\sqrt{T})$ and therefore the online-to-batch approach provides a convergence rate of $1/\sqrt{T}$ for the generalization error, which is known to be optimal for many cases.

References

Nicolo Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, 2004.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, 2003.