COS-511: Learning Theory **Spring 2017**

Lecture 17: Intro. to Online Learning

Lecturer: Roi Livni

17.1 Online Learning

So far, the course has focused on the Statistical setting where a learner gets to observe IID sample and needs to return a hypothesis or a concept. In reality, things can be much more complicated. First, sometimes instead of learning over a batch sample, we would like our algorithm to learn in an online manner: where at each point in time, the algorithm gets a new point and update its decision rule. Second, it could be that with time the generating distribution of the sample changes, and we would like our algorithm to react to such changes and adapts itself. Finally, we might not even want to assume that the data is generated from some distribution, but perhaps it is even adversarially chosen to cause our algorithm to fail.

A different yet strongly related model that has seen vast amount of research is the *online learning* setting. The general framework of online learning is as follows:

Online Learning: Problem Setup In a general setting, we assume that at each iteration the learner gets to observe a feedback $\mathbf{x}_t$ and chooses an action $a_t \in \mathcal{A}$ a class of action. The adversery then chooses a loss function and inflict a loss $\ell_t(\mathbf{x}_t, a_t)$. The aim of the learner is then to minimize her regret

$$\text{Regret}_T = \sum_{t=1}^{T} \ell_t(\mathbf{x}_t, a_t) - \min_{a^* \in \mathcal{A}} \sum_{t=1}^{T} \ell_t(\mathbf{x}_t, a^*)$$

A problem is set to be learnable if we can achieve sublinear regret in terms of the horizon T. i.e. $\text{Regret}_T = o(T)$.

A special case of online learning is the setting of online convex optimization, which as we will see captures many of the known interesting instances for online learning. We begin by recalling the setting

Online Convex Optimization: Probem Setup At each round t we get to choose a point $\mathbf{x}^{(t)}$ and suffer a loss given by an arbitrary convex function: $f_t(\mathbf{x}^{(t)})$. The objective of the learner would be to minimize the following term (which is called *regret*):

$$\text{Regret}_T = \sum_{t=1}^{T} f_t(\mathbf{x}^{(t)}) - \min_{\mathbf{x}^* \in \mathcal{K}} \sum_{t=1}^{T} f_t(\mathbf{x}^*)$$

The online setting differs in two major things from the statistical (or *batch*) model:

1. Examples are not presented in a batch but are streamlined throughout time.
2. We avoid making distributional assumptions – we do not necessarily assume that the examples are IID distributed.

Example 17.1 (Prediction from expert advice)**.** *As a first example we consider, what is perhaps the most well known problem in prediction theory:* prediction from expert advice*. At each iteration t a decision maker has to choose an advice from one of n experts. After making her choice a loss between* 0 *and* 1 *is incured. The scenario is repeated iteratively, and at each iteration the costs of the various experts are arbitrary. The goal of the learner is to do as well as the best expert in hindsight.*

Note that this model is closely related to learning a finite hypothesis class. We can think of each hypothesis in a class $\mathcal{H}$ as an expert. At each iteration t we choose a hypothesis h_t and get to observe an example $(\mathbf{x}^{(t)}, y_t)$ the "experts" incurs loss 1 if $h_t(\mathbf{x}^{(t)}) \neq y_t$ and 0 else. The objective is to again to do as well as the best hypothesis in hindsight.

The problem can modeled using convex optimization as follows: We think of the n experts as a vector $\|\mathbf{g}_t\|_\infty \leq 1$ where each coordinate corresponds to the loss incured by the experts. The learner chooses at each iteration a vector $\mathbf{x}_t \in \Delta_n$ in the simplex over $\mathbb{R}^n$ i.e.

$$\Delta_n = \{\mathbf{x} \in \mathbb{R}^n : \sum \mathbf{x}_i = 1 \ , \mathbf{x} \geq 0\} \tag{17.1}$$

At each iteration t the learner chooses an expert according to the distribution $\mathbf{x}_t$ and incurs expected loss $f_t(\mathbf{x}_t) = \mathbf{g}_t \cdot \mathbf{x}_t$.

Thus prediction from expert advice is a special case of OCO where the cost functions are vectors with bounded norm $\|\mathbf{g}\|_\infty\| \leq 1$ and the decision set of the learner is given by the simplex Δ_n the cost function is given by the linear map $\mathbf{g}_t \cdot \mathbf{x}_t$. Note that the optimal point at the simplex Δ_n is always attained by a single expert. i.e. for cost vectors $\mathbf{g}_1, \ldots, \mathbf{g}_T$

$$\min_{\mathbf{x}^* \in \Delta_n} \sum \mathbf{g}_t \cdot \mathbf{x}^* = \min_{i=1,\ldots n} \sum (\mathbf{g}_t)_i$$

Example 17.2 (Online Shortest Path)**.** *In the online shortest path setting, the decision maker is given a directed graph (V, E), and a source-sink pair $u, v \in V$. At each iteration $t \in [T]$, the decision maker chooses a parth $p_t \in \mathcal{P}_{u,v}$ where $\mathcal{P}_{u,v}$ is the set of all paths beginning in u ending in v in the graph. The adversery chooses weight of the deges of the graphs: $\mathbf{w}_t : E \to \mathbb{R}$. The decision maker suffers a loss $\sum_{e \in p_t} \mathbf{w}_t(e)$.*

Note that the problem can be modeled using the expert advice model when we think of each path as an expert. However, this gives as exponentially many expert (hence, even going over all of the experts advice may become prohibitive). Alternatively the online shortest path can be cast as an online convex optimization problem. First let us consider the class of all feasible paths as integer vectors where $p_t(e) = 1$ if the path passes through edge e. Then the class of all feasible pathses may be characterized as:

$$\sum_{(e=(u,*))} \rho(e) = \sum_{(e=(*,v))} \rho(e) = 1 \qquad \text{path starts and ends and } u \text{ and } v \text{ respectively}$$

$$\forall z \notin u, v \sum_{e=(*,z)} \rho(e) = \sum_{e=(z,*)} \rho(e) \qquad \text{Flow conservation}$$

Taking the convex hull (or considering distribution over pathes) we obtain the path polytope

$$\mathcal{K} = \{p : \sum_{(e=(u,*))} p(e) = \sum_{(e=(*,u))} p(e) = 1, \forall z \notin \{u, v\} \sum_{e=(*,z)} p(e) = \sum_{e=(z,*)} p(e), \forall e \in E, 0 \leq p(e) \leq 1\}$$

Given a distribution $p_t \in \mathcal{K}$ over pathes, the expected loss inflicted on the learner for a set of weights $\mathbf{w}_t$ is given by

$$f_t(p_t) = \mathbf{w}_t \cdot p_t$$

Example 17.3 (Portfolio Selection)**.** *In the portfolio selection model, at each iteration $t \in [T]$, the decision maker chooses a distribution of her wealth of n assets $\mathbf{x}_t \in \Delta_n$. The adversary independently chooses market returns for the assets: a vector $\mathbf{r}_t \in \mathbb{R}^n$. The entry $\mathbf{r}_t(i)$ is the price ratio for the i-th asset. The ratio of the wealth of the investor at iteration $t+1$ and t is $\mathbf{r}_t \cdot \mathbf{x}_t$. Hence the gain in this setting is defined to be the logarithm of this changes: $\log(\mathbf{r}_t \cdot \mathbf{x}_t)$. The goal of the learner is then to minimize the difference:*

$$\max_{\mathbf{x}^* \in \Delta_n} \sum_{t=1}^{T} \log(\mathbf{r}_t \cdot \mathbf{x}^*) - \sum_{t=1}^{T} \log(\mathbf{r}_t \cdot \mathbf{x}_t)$$

17.2 OGD

We begin by recalling the OGD algorithm that we have already analysed:

Theorem 17.1. *Consider the Online Convex Optimization setting. Suppose that at each round t: f_t is a G-Lipschitz convex function and let $\mathcal{K}$ be a set of diameter D. Apply Alg. 1 to the sequence with step size $\eta_t = \frac{DG}{\sqrt{t}}$, then for the output $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(t)}$ we have that:*

$$\textit{Regret}_T = \sum f_t(\mathbf{x}^{(t)}) - \min_{\mathbf{x}^*} \sum f_t(\mathbf{x}^*) = O(DG\sqrt{T})$$

Algorithm 1 Online Gradient Descent

Input: A sequence of arbitrary functions $f_1, \ldots, f_T$ step sizes $\{\eta_t\}$, $\mathbf{x}_1 \in \mathcal{K}$
for $t = 1, 2 \ldots T$ **do**
Observe f_t and suffer cost $f_t(\mathbf{x}^{(t)})$.
Set $\nabla_t = \nabla f_t(\mathbf{x}^{(t)})$.
Update and project:

$$\mathbf{y}_{t+1} = \mathbf{x}^{(t)} - \eta_t \nabla_t$$
$$\mathbf{x}^{(t)} = \Pi_{\mathcal{K}}(\mathbf{y}_{t+1})$$

end for
return $\mathbf{x}_1, \ldots, \mathbf{x}^{(t)}$.

Application to Expert Advice As a first application of OGD, we will apply it to the expert advice setting. In this setting the learner chooses $\mathbf{x}_1 \in \Delta_n$. The diameter of Δ_n is smaller than one as it is contained in the unit ball. At each iteration the adversary chooses a function of the form $f_t = \mathbf{g}_t \cdot \mathbf{x}_t$, where $\|\mathbf{g}_t\|_\infty \leq 1$ which gives us a bound $\|\mathbf{g}_t\|_2 \leq \sqrt{n}$. Since $\nabla f_t = \mathbf{g}_t$ we obtain the following result for Expert Advice:

Claim 17.2. *Apply Alg. 1 to the Expert advice setting then we obtain the following regret bound*

$$\textit{Regret}_T \leq O(\sqrt{nT})$$

The first natural question is whether OGD obtain optimal rate? in terms of T and n. We will see in the next lecture that we can in fact achieve a much better rate of $O(\sqrt{T \log n})$.