18.1 Expert Advice

In the last lecture we’ve applied OGD algorithm to the expert advice problem to obtain a regret bound of \( O(\sqrt{nt}) \). 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 \( O(\sqrt{T \log n}) \). A good way to obtain intuition as to the optimality of a solution is to consider an analogue stochastic setting: Suppose at each iteration the adversary chooses \( g_t \), IID. Given a sequence of IID examples \( g_1, \ldots, g_t \), what is the best strategy for the learning? One thing the learner can do is to apply a learning algorithm. Therefore a simple ERM rule would be

\[
x_{t+1} = \arg \min_{x^* \in K} g_t \cdot x^*.
\]

When discussing statistical learning theory we called this strategy ERM in online learning this is sometimes referred to as follow the leader approach. Since we assume that the samples are chosen IID, we know by radamacher theory that:

\[
E[g_{t+1} \cdot x_{t+1}] \leq \min_{x^* \in K} E[g_{t+1} \cdot x^*] + R_t(K) + O \left( \frac{1}{\sqrt{t}} \right)
\]

We have analysed the Rademacher complexity of the class \( K \): this is the class of \( \ell_1 \) regularized classifiers and we have that \( R_t(K) \leq \sqrt{\log n \frac{1}{t}} \). Thus, if at each iteration we choose the leader or apply an ERM rule we have that:

\[
\sum E(g_t \cdot x_t) - \min_{x^* \in K} E(g_t \cdot x^*) \leq \sum O\left( \sqrt{\frac{\log n}{t}} \right) = O(\sqrt{T \log n})
\]

Therefore we see that, at least in the stochastic case OGD is sub–optimal. As we will see, we can achieve the same regret bound as in the stochastic case, but we will need to use a different algorithm. Namely the Multiplicative Weight algorithm which we next discuss.

**Algorithm 1** Follow the Leader

- **Initialization** \( x_1 \in K \).
- for \( t = 1, 2, \ldots, T \) do
  - Observe \( f_t \) and suffer cost \( f_t(x_t) \).
  - Set \( x_{t+1} = \arg \min_{x^* \in K} \sum_{t' \leq t} f_t(x^*) \)
- end for
- return
18.2 MW Algorithm

We next consider a different algorithm called *Multiplicative Weights*, or *Hedge* to the expert advice problem. Similar to the OGD algorithm, Hedge keeps weights over the different expert at each time step and update those according to the observed losses. In contrast to OGD the update is multiplicative:

**Algorithm 2 Hedge**

**Initialization** \( W_1 = 1 \in \mathbb{R}^n \% \; W_1 = (1, 1, \ldots, 1). \)

SET \( x_1 = \frac{1}{n} W_1. \)

for \( t = 1, 2 \ldots T \) do

Pick \( i_t \propto x_t. \) i.e. \( p(i_t = i) = x_t(i). \)

incure loss \( g_t(i). \)

Update Weights \( W_{t+1}(i) = \frac{W_t(i) \cdot e^{-\epsilon g_t(i)}}{\sum_j W_t(j)}. \)

Set \( x_{t+1}(i) = \frac{W_{t+1}(i)}{\sum_j W_{t+1}(j)}. \)

end for

return

**Lemma 18.1.** Let \( g^2 \) denote the \( n \)-dimensional vector of pointwise square losses (i.e \( g^2(i) = (g(i))^2 \)), let \( \epsilon > 0 \) and assume all losses \( g \) to be non-negative. The Hedge Algorithm satisfies for every expert \( i^* \):

\[
\sum_{t=1}^{T} x_t \cdot g_t \leq \sum_{t=1}^{T} g_t(i^*) + \epsilon \sum_{t=1}^{T} x_t \cdot g^2_t + \frac{\log n}{\epsilon} \tag{18.1}
\]

**Proof.** Set \( \Phi_t = \sum_{t=1}^{T} W_t(i) \) for all \( t \leq T \), and note that \( \Phi_1 = n. \)

Inspecting the sum of weights

\[
\Phi_{t+1} = \sum_i W_t(i) e^{-\epsilon g_t(i)}
\]

\[
= \Phi_t \sum_i x_t e^{-\epsilon g_t(i)}
\]

\[
\leq \Phi_t \sum_i x_t (1 - \epsilon g_t(i) + \epsilon^2 g_t(i))
\]

\[
= \Phi_t (1 - \epsilon x_t \cdot g_t + \epsilon^2 x_t \cdot g^2_t)
\]

\[
\leq \Phi_t e^{-\epsilon x_t \cdot g_t + \epsilon^2 x_t \cdot g^2_t}
\]

And by definition of expert \( i^* \) we have that:

\[
W_T(i^*) = e^{-\epsilon \sum_{t=1}^{T} g_t(i^*)}.
\]

Given that \( W_T(i^*) \) is less then the sum of weights \( \Phi_T \), we have that

\[
W_T(i^*) \leq N e^{-\epsilon \sum x_t \cdot g_t + \epsilon^2 \sum x_t \cdot g^2_t}
\]

Taking logarithm of both sides we get:

\[
-\epsilon \sum_{t=1}^{T} g_t(i^*) \leq \log N - \epsilon \sum_{t=1}^{T} x_t \cdot g_t + \epsilon^2 \sum_{t=1}^{T} x_t \cdot g^2_t
\]

And the result follows immediately. \( \square \)
**Theorem 18.2.** Apply Alg. 2 to the Online Expert problem, with $\epsilon = \frac{\log n}{T}$ then

$$\text{Regret}_T = O(\sqrt{T \log N})$$

*Proof.* First observe that $g^2 \leq 1$ hence $x_t \cdot g_t^2 \leq 1$. Plugging this into Eq. 18.1 we obtain that for every $i^*$:

$$\sum x_t \cdot g_t - g_t(i^*) \leq T \epsilon + \frac{\log n}{\epsilon}$$

The algorithm picks the action of expert $i$ at iteration $i$ according to $x_t$ hence incurs expected loss of $x_t \cdot g_t$ overall we have that for our choice of $\epsilon$:

$$\text{Regret}_T \leq T \epsilon + \frac{\log n}{\epsilon} \leq 2 \sqrt{T \log n}$$

Currently, the MW algorithm and OGD algorithm seem like two completely different algorithms that happen to solve different instances of OCO. In future lectures we will see how they can both be derived from a unifying setting.

### 18.3 Online to Batch

We have already seen intuitively that some relation between online guarantees on the generalization bounds in the stochastic setting. For example, we could derive the SGD algorithm from the OGD algorithm, and we could achieve regret bounds that are similar to the performance of the ERM algorithm. We next show that indeed we can always derive generalization bounds for the statistical setting from the online framework.

**Theorem 18.3** (Online to Batch). Let $\mathcal{A}$ be an OCO algorithm whose regret after $T$ iteration is guaranteed to be $\text{Regret}_T(\mathcal{A})$. Let $f_1, \ldots, f_T$ be an IID sequence of convex functions bounded by 1 s.t. $E(f_t) = f$. Then for any $\delta > 0$, with probability at least $(1 - \delta)$ (over the sample of convex functions), it holds for $\bar{x} = \frac{1}{T} \sum x_t$:

$$f(\bar{x}) \leq \min_{x^*} f(x^*) + \frac{\text{Regret}_T(\mathcal{A})}{T} + \sqrt{\frac{8 \log(2/\delta)}{T}}$$

(18.2)

**Application to Learning** Before proving the online to batch result, let us consider its application to learning convex problems: Assume that $(x, y)$ are drawn IID from some arbitrary distribution $D$, and we let for every $w$ $L(w) = E(\ell(w, (x, y)))$, where we assume $\ell$ is a convex function in $(w)$ for every $(x, y)$. Then by drawing and IID sequence of samplex $(x_t, y_t)_{t=1}^T$ and applying on Online algorithm, we obtain (by setting $f_t(w_t) = \ell(w_t, (x_t, y_t))$ with probability $(1 - \delta)$ that

$$L(\bar{w}) \leq L(w^*) + O \left( \frac{\text{Regret}_T(\mathcal{A}) \log 1/\delta}{\sqrt{T}} \right)$$

In particular for every algorithm with regret $O(DG\sqrt{T})$, if $T > \frac{1}{D^2G^2\epsilon^2}$ we have w.h.p

$$L(\bar{w}) \leq L(w^*) + O(\epsilon)$$

Thus, we obtain a learning algorithm.
Note that we have seen this phenomena in the special case where we applied OGD algorithm for learning and obtained the SGD algorithm.

**Proof of Online2Batch** The proof (as always) relies on bounding the expected error and concentration result. The concentration inequality we will use is Azuma’s inequality for martingales:

**Definition 18.4.** A sequence of random variables \( X_1, X_2, \ldots \), is called a martingale if

\[
E(X_{t+1} | X_t, \ldots, X_1) = X_t \quad \forall t > 0
\]

**Theorem 18.5** (Azuma’s inequality). Let \( \{X_i\}_{i=1}^T \) be a martingale of \( T \) random variables that satisfy \(|X_i - X_{i+1}| \leq 1\). Then:

\[
P(|X_T - X_0| > c) \leq e^{-\frac{c^2}{2T}}
\]

For the proof we start by defining a sequence of martingales: Let us write \( Z_t = f(x_t) - f_t(x_t) \) and \( X_t = \sum_{i=1}^t Z_i \). We first verify that \( X_t \) is indeed a martingale. Notice that by definition we have:

\[
E_{f_t \sim D} (Z_t | X_{t-1}) = E(f(x_t)) - E(f_t(x_t)) = 0
\]

Where the last equality is true, since \( f_t \) is independent of \( x_t \). Thus by definition of \( Z_t \) we have that:

\[
E(X_{t+1} | X_1, \ldots, X_t) = E(Z_{t+1} | X_t) + X_t = X_t
\]

In addition by the bound on \( f_t \) we have that

\[
|X_{t+1} - X_t| = |Z_t| \leq 1
\]

Applying Azuma’s inequality to the martingale \( X_1, \ldots, X_T \) we have that:

\[
P(X_T > c) \leq e^{-\frac{c^2}{2T}}
\]

By definition of \( X_T \), dividing by \( T \) and setting \( c = \sqrt{2T \log 2/\delta} \) we have that:

\[
P\left(\frac{1}{T} \sum_{i=1}^T f_t(x_t) - f(x_t) > \sqrt{\frac{2 \log 2/\delta}{T}}\right) \leq \frac{\delta}{2}.
\]

A similar martingale can be defined to \( x^* \) and we obtain analogously

\[
P\left(\frac{1}{T} \sum_{i=1}^T f_t(x^*) - f(x^*) < -\sqrt{\frac{2 \log 2/\delta}{T}}\right) \leq \frac{\delta}{2}
\]

Overall we obtain that

\[
f(\bar{x}_T) f(x^*) \leq \Gamma_1 - \Gamma_2 + \frac{1}{T} \sum_{i=1}^T f_t(x_t) - f_t(x^*) \leq \frac{\text{Regret}_p(A)}{T} + \Gamma_1 - \Gamma_2
\]

Thus with probability at least \( 1 - \delta \) by our bounds for \( \Gamma_1, \Gamma_2 \) we obtain the result.