# Lecture 15: Online Learning and Convex Optimization

Akshay Krishnamurthy
akshay@cs.umass.edu

November 1, 2017

## 1 Recap

Last time we introduced the online learning model and studied the realizable case. We showed how the halving algorithm can achieve logarithmic mistake bound and introduced the Littlestone dimension which characterizes online learnability in the mistake bound model. Finally we introduced the notion of regret in our attempt to relax the realizability assumption.

## 2 Regret and Weighted Majority

Recall the new online learning formulation. We are given a set of hypotheses $\mathcal{H} : \mathcal{X} \to [K]$. Then, for $t = 1, \ldots, T$

1. We receive $x_t \in \mathcal{X}$
2. We choose action $p_t \in \Delta([K])$
3. We receive loss $\langle p_t, \ell_t \rangle$ and we observe $\ell_t$.

Our goal is to minimize the *regret*

$$\text{Regret}(T) = \sum_{t=1}^{T} \langle p_t, \ell_t \rangle - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell_t(h(x_t))$$

This maps to our previous setting with $K = 2$, $\ell_t(y) = \mathbf{1}\{y \neq y_t\}$, but clearly generalizes to multi-class prediction and to other loss functions. Note that it is only more general (and we'll use this formulation later) to ignore the examples altogether and have $p_t \in \Delta(\mathcal{H})$ and $\ell_t \in [0,1]^{|\mathcal{H}|}$. This latter formulation is called the *Experts setting.*

**Weighted Majority/Hedge.** The weighted majority algorithm achieves sublinear regret in this setting. We consider the finite hypothesis class case, $|\mathcal{H}| = N$. We start with $w^{(1)} = (1, \ldots, 1) \in \mathbb{R}^N$ and with $\eta = \sqrt{2\log(N)/T}$, which requires knowing the number of rounds $T$ (although this can be relaxed). Then for $t = 1, \ldots, T$.

1. set $\tilde{w}^{(t)} = w^{(t)}/Z_t$ where $Z_t = \sum_h w^{(t)}(h)$ is the normalization.
2. set $p_t(h) = \tilde{w}^{(t)}(h)$ and predict with $p_t$.
3. observe loss vector $\ell_t \in [0,1]^d$, suffer $\langle p_t, \ell_t \rangle$.
4. update $w^{(t+1)}(h) = w^{(t)}(h) \exp(-\eta \ell_t(h(x_t)))$.

Intuitively whenever a hypothesis makes a mistake we shrink it's weight dramatically so that the distribution $\tilde{w}^{(t)}$ concentrates on the hypotheses that are doing well. This enables us to be competitive with these hypothesis, which leads to a good regret bound.

Before turning to the regret bound, observe that the algorithm has the same flavor of update as AdaBoost. Namely if we map the hypotheses here to the examples in the Boosting setting, then the algorithms are actually identical. In AdaBoost, we maintain a distribution over the examples and we increase the weight if the loss is high and we decrease it if the loss is low. Here we maintain a distribution over the hypotheses but do the same thing. In homework 3 we saw the game-theoretic interpretation of AdaBoost, where we are one of the players in the game. In Weighted Majority, we are just the other player, but running the same algorithm!

**Theorem 1** (Weighted Majority regret bound)**.** *For $T > 2\log(N)$ the Weighted Majority algorithm has regret*

$$\mathit{Regret}(T) = \sum_{t=1}^{T} \langle p_t, \ell_t \rangle - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} \ell_t(h(x_t)) \leq \sqrt{2T\log(N)}$$

*Proof.* The proof is actually quite similar to the Adaboost proof.

$$\log \frac{Z_{t+1}}{Z_t} = \log \sum_h \frac{w^{(t)}(h)\exp(-\eta\ell_t(h(x_t)))}{Z_t} = \log \sum_h \tilde{w}^{(t)}(h)\exp(-\eta\ell_t(h(x_t))) = \log \sum_i p_t(i)\exp(-\eta\ell_t(i))$$
$$\leq \log\left(1 - \eta\langle p_t, \ell_t\rangle + \frac{\eta^2}{2}\langle p_t, \ell_t^2\rangle\right) \leq -\eta\langle p_t, \ell_t\rangle + \frac{\eta^2}{2}\langle p_t, \ell_t^2\rangle \leq -\eta\langle p_t, \ell_t\rangle + \eta^2/2.$$

Here we used two tricks. First observe that $\eta\ell_t(i) \in [0,1]$ and use the second order taylor expansion of $e^{-x} \leq 1-x+x^2/2$. Then, use that $\log(1-x) \leq -x$ for $x \in (0,1)$, which it is here. Of course we know that $\log(Z_1) = \log(N)$ and so we get a telescoping sum

$$\log Z_{T+1} = \sum_{t=1}^{T} \log \frac{Z_{t+1}}{Z_t} + \log(Z_1) \leq -\eta \sum_t \langle p_t, \ell_t\rangle + \frac{T\eta^2}{2} + \log(N)$$

The last step is to lower bound $\log(Z_{T+1})$. For any $h \in \mathcal{H}$

$$\log Z_{T+1} = \log \sum_h \exp(-\eta \sum_t \ell_t(h(x_t))) \geq -\eta \sum_t \ell_t(h(x_t)).$$

Combining we get

$$\eta\left(\sum_{t=1}^{T} \langle p_t, \ell_t\rangle - \min_h \sum_t \ell_t(h(x_t))\right) \leq \log(N) + \frac{T\eta^2}{2}$$

Now diving by $\eta$ and using our choice proves the theorem. □

**Corollary 2.** *There exists a randomized prediction algorithm for online classification that enjoys a regret bound of (where $q_t \in [0,1]$ is the probability of predicting label 1)*

$$\forall h \in \mathcal{H} \sum_{t=1}^{T} |q_t - y_t| - \sum_{t=1}^{T} |h(x_t) - y_t| \leq \sqrt{2T\log(|\mathcal{H}|)}$$

*Proof.* We just have to translate from the absolute loss which is easy since we are doing binary classification. Let the loss vector have coordinates $\ell_t(i) = \mathbf{1}\{y_t \neq i\}$ then clearly $|h(x_t) - y_t| = \ell_t(h(x_t))$ but it is also true that $\langle p_t, \ell_t\rangle = |q_t - y_t|$ where $p_t(1) = q_t, p_t(0) = 1 - q_t$. Thus after translation we may apply Multiplicative weights. □

**Theorem 3.** *There is an algorithm for binary classification that enjoys*

$$\mathit{Regret}(T) \leq \sqrt{2T\,\mathit{Ldim}(\mathcal{H})\log(eT)}$$

The idea here is to run multiplicative weights on a subset of the now infinite hypothesis space. Instead of using the individual hypotheses as the *experts*, each expert will itself be a learning algorithm, that is kind of like the Halving algorithm but for Littlestone classes. Each one of these algorithms is initialized with a sequence of $L \leq \text{Ldim}(\mathcal{H})$ time indices, which correspond to the rounds where it will disagree from its majority. Otherwise it will predict like the majority. It can be shown that for every sequence $x_1, \ldots, x_T$ and every hypothesis $h$, there exists one of these base learners that behaves exactly like $h$, via the mistake bound for the halving algorithm. Then when weighted majority is run on top we achieve the regret bound.

## 3 Online Convex Optimization

We start with introducing a more general setting, that of *online convex optimization*. Here the learner has a decision set $S$ and at each round chooses $w_t \in S$, while the adversary chooses a convex loss function $f_t : S \to \mathbb{R}$. The learner suffers loss $f_t(w_t)$ and would like to minimize the regret

$$\text{Regret}(T) = \sum_t f_t(w_t) - \min_u \sum_t f_t(u).$$

**Example 1** (Linear regression)**.** *Let $S = \mathbb{R}^d$ and let $f_t(w) = (\langle w, x_t \rangle - y_t)^2$ which is clearly convex. Then using an algorithm for OCO we can solve online linear regression. At each round OCO gives us a vector $w_t$ which we use to make the prediction $\langle w_t, x_t \rangle$.*

**Example 2** (Experts)**.** *Let $S = \Delta(d)$ and $f_t(w) = \langle w, \ell_t \rangle$ be a linear function, which is also convex. Then we can use an OCO algorithm for the experts problems. Here for convexity it is important that we can play a distribution over the experts, which as we saw last time was important for achieving non-trivial regret in the experts setting.*

**Follow-the-leader** The most obvious algorithm for OCO is *follow the leader*. At each round we play

$$w_t \in \operatorname*{argmin}_{w \in S} \sum_{i=1}^{t-1} f_i(w),$$

which is the best choice based on all the past data. This is also like using the empirical risk minimizer at round $t$ to predict. We will first analyze Follow-the-leader.

**Lemma 4** (Be-the-leader Lemma)**.** *For any $u \in S$*

$$\textit{Regret}(u) = \sum_{t=1}^{T} f_t(w_t) - f_t(u) \le \sum_{t=1}^{T} f_t(w_t) - f_t(w_{t+1})$$

Note that $w_{t+1}$ is the leader *after* seeing the $t^{\text{th}}$ loss function, so it is not something we can actually use. However, the lemma says that the regret is small whenever the predictions are *stable*, in the sense that seeing $f_t$ does not radically change your prediction on $f_t$. Another way people say this is that Be-the-leader has negative regret, but unfortunately it is not something we can implement.

*Proof.* Since $f_t(w_t)$ is on both sides, instead we show that be-the-leader has negative regret

$$\sum_{t=1}^{T} f_t(w_{t+1}) - f_t(u) \le 0$$

We prove this by induction. Clearly at round 1, by definition of $w_2$ we know that for all $u$, $f_1(w_2) \le f_1(u)$. Now at time $\tau > 1$, for any $u$

$$\sum_{t=1}^{\tau} f_t(w_{t+1}) = \sum_{t=1}^{\tau-1} f_t(w_{t+1}) + f_\tau(w_{\tau+1}) \le \sum_{t=1}^{\tau} f_t(w_{\tau+1}) \le \sum_{t=1}^{\tau} f_t(u).$$

The first inequality here is the induction hypothesis, that Be-The-Leader up to round $\tau - 1$ is better than any fixed hypothesis, including $w_{\tau+1}$. The second inequality uses that $w_{\tau+1}$ is the ERM. □

**Example 3** (FTL with quadratic loss)**.** *If the losses are of the form $f_t(w) = \frac{1}{2}\|w - z_t\|_2^2$ and $\|z_t\| \le L$ for all $t$ then FTL, with $S = \mathbb{R}^d$ has a closed form solution $w_t \leftarrow \frac{1}{t-1}\sum_{i=1}^{t-1} z_i$. Moreover the stability is bounded*

$$\begin{aligned} f_t(w_t) - f_t(w_{t+1}) &= \frac{1}{2}\|w_t - z_t\|_2^2 - \frac{1}{2}\|(1 - 1/t)w_t + 1/tz_t - z_t\|_2^2 \\ &= \frac{1}{2}(1 - (1 - 1/t)^2)\|w_t - z_t\|_2^2 \le 4(1/t)L^2. \end{aligned}$$

*Hence the regret is $O(L^2 \log(T))$.*

**Example 4** (FTL with linear loss)**.** *Unfortunately for linear losses the stability term cannot be controlled. Consider* $S = [-1, 1]$ *and the loss function* $f_t(w) = wz_t$ *where the sequence of* $z_t s$ *is* $(-0.1, 1, -1, 1, -1, \ldots)$*. The* $w_t s$ *that FTL predicts are* $(0, -1, 1, -1, 1, -1, 1, \ldots)$ *which have widly oscillating behavior and are clearly unstable.*

# 4 Follow-the-Regularized Leader

Above we saw that FTL works well when the losses or the algorithm is stable. The idea with FTRL is to force the algorithm to be stable by using regularization. Let $R : S \to \mathbb{R}$ be some function, which we will call a regularizer from now on. Then FTRL plays

$$w_t \in \operatorname*{argmin}_{w \in S} R(w) + \sum_{i=1}^{t-1} f_i(w)$$

FTL is the same algorithm where $R(w) = 0$. The point now is that $R$ is forcing stability of the algorithm. Before turning to the general case let us just study the quadratic regularizer $R(w) = \frac{1}{2\eta}\|w\|_2^2$ with linear loss functions:

$$w_t \in \operatorname*{argmin}_{w \in S} \frac{1}{2\eta}\|w\|_2^2 + \sum_{i=1}^{t-1} w\ell_i = \operatorname*{argmin}_{w \in S} \frac{1}{2\eta}\|w\|_2^2 - \langle w, \theta_t \rangle$$

where $\theta_t = -\sum_{i=1}^{t-1} \ell_i$ is the sum of the loss functions so far. When $S = \mathbb{R}^d$ FTRL has a closed form

$$w_t = \eta\theta_t$$

which we can equivalently write as $w_t = w_{t-1} - \eta\ell_{t-1}$ which is the *Online Gradient Descent* algorithm.

If $S \neq \mathbb{R}^d$ then we are doing a lazy projection step where

$$w_t = \Pi_S(\eta\theta_t),$$

This is called a lazy projection since we don't project the iterates $\theta_t$ but only project when we need to make a prediction. This is also called *Nesterov's Dual Averaging* algorithm.