CS6781 - Theoretical Foundations of Machine Learning

Lecture 13: Online Optimization

March 5, 2020

Lecturer: Nika Haghtalab *Readings: N/A*
Scribe: Ruihan Wu and Charles Dai

1 Overview

Recall that last time, we began to explore the difference between online and offline learnability of hypothesis classes. While offline learnability is characterized by *finite VC dimension*, we saw an example of a class of hypothesis, namely 1-dimensional thresholds, that has a VC dimension of 1, but cannot be learned in the online setting. In this lecture, we seek an analogous notion to VC dimension for online learnability which will allow us to bound regret for infinite hypothesis classes.

	Finite Hypothesis Classes	Infinite Hypothesis Classes
Offline (error)	$O\left(\sqrt{\frac{\ln(\lvert\mathcal{H}\rvert)}{m}}\right)$	$\Theta\left(\sqrt{\frac{\text{VC-Dim}(\mathcal{H})}{m}}\right)$
Online (avg. regret: $\frac{R_T}{T}$)	$O\left(\sqrt{\frac{\ln\lvert\mathcal{H}\rvert}{T}}\right)$ (last lecture)	**What about infinite classes?**

2 Online Learnability

In order to reason about class complexity in offline learning, we considered the shattering of sets. For online learnability we consider *shattering trees*, like the one that we used in the discussion of 1-dimensional thresholds last lecture.

Definition 2.1 (Shattering a Tree)**.** Consider a full binary tree of depth d such that each node is labeled by an $x \in \mathcal{X}$. This tree is said to be shattered by $\mathcal{H}$ if for every set of labels $\{y_i\}_{i=1}^d \in \{+,-\}^d$, the root-to-leaf path $x_1, \ldots, x_d$ that is defined by starting at the root and taking left child if $y_i = -$ and taking the right child otherwise, is such that $\exists h \in \mathcal{H}, h(x_i) = y_i, \forall i \in [d]$.

Essentially, for a tree to be shattered, for every path in the tree (corresponding to a set of labels y_is) there is a hypothesis in $\mathcal{H}$ that labels elements of this path according to y_is.

Definition 2.2 (Littlestone Dimension)**.** $\text{LDim}(\mathcal{H})$ is the maximal depth of a tree that is shattered by $\mathcal{H}$.

For example, the Littlestone dimension of a 1-dimensional threshold is infinity. Note that in this case, the Littlestone dimension is much larger than the VC dimension. We make three remarks about the magnitude of the Littlestone dimension.

Remark 2.3. For any class $\mathcal{H}$,

$$\mathrm{LDim}(\mathcal{H}) \leq \log_2(|\mathcal{H}|).$$

Because each labelling constructs a different path, each path requires a different hypothesis $h \in \mathcal{H}$. Therefore, the number of paths cannot exceed $|\mathcal{H}|$, and $\log_2$ (# of paths) $= \mathrm{LDim}(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$.

Remark 2.4. For any class $\mathcal{H}$,

$$\mathrm{LDim}(\mathcal{H}) \geq \mathrm{VCDim}(\mathcal{H}).$$

Labelings on a set can be encoded as a tree in which every layer has the same element. Note that if this tree is shattered, the set is shattered as well, so $\mathrm{LDim}(\mathcal{H}) \geq \mathrm{VCDim}(\mathcal{H})$. However, because the Littlestone dimension does not require the same element x on each layer it can be much larger than the VC dimension.

Remark 2.5. There is a class $\mathcal{H}$ such that

$$\mathrm{LDim}(\mathcal{H}) \ll \log(|\mathcal{H}|).$$

To give an example of this fact, consider the hypothesis class $h_a = \mathbb{1}(x = a)$ for $a \in [0, 1]$. Once we have a positive label, we know the hypothesis. Put another way, no tree of depth two can exists because no hypothesis can fit two different positive instances. Therefore, $\mathrm{LDim}(\mathcal{H}) = 1$, but $|\mathcal{H}|$ is infinite.

Our goal is to be able to lower bound the error of an online learning algorithm using this notion of the Littlestone dimension. This next theorem allows us to do so.

Theorem 2.6. *Any algorithm $\mathcal{A}$ for learning $\mathcal{H}$ in the mistake bound model has mistake bound $M \geq \mathrm{LDim}(\mathcal{H})$.*

The sketch of the proof is as follows: take a shattered tree and start at x_1. Let $y_t = -\hat{y}_t$ and take the path defined by $y_1, y_2, \ldots y_d$. Along this path, there are d mistakes that are made. Because by definition, each path is consistent with an $h \in \mathcal{H}$, there exists an h that is consistent.

This next theorem describes the error bound when the Littlestone dimension is infinite.

Theorem 2.7. *If $\mathrm{LDim}(\mathcal{H})$ is infinite, then any algorithm $\mathcal{A}$ has a regret bound $R \geq \Omega(T)$.*

The sketch of the proof is as follows: Let $y_t \sim \mathrm{Unif}\{+, -\}$ and take any path along the tree. Because the tree is shattered, there exists a hypothesis h that is consistent with any path, so $\mathrm{OPT} = 0$. However, the labels are random, so the expected error at any step is $\frac{1}{2}$. Therefore, regret is $R \geq T \cdot \frac{1}{2} = T/2$.

Ben-David et al. [2009] showed that the regret of any learner is in fact captured by the Littlestone dimension. Additionally, Ben-David et al. [2009] also established an algorithm that recovers the same bound up to logarithmic factors.

Theorem 2.8 (Ben-David et al. [2009]). *Any algorithm $\mathcal{A}$ for learning $\mathcal{H}$ in the online learning setting has regret bound $\Omega\left(\sqrt{\mathrm{LDim}(\mathcal{H}) \cdot T}\right)$. Additionally, there is an online learning algorithm that has regret bound $O\left(\sqrt{\mathrm{LDim}(\mathcal{H}) \cdot T \cdot \log T}\right)$.*

Ignoring the logarithmic term in T, this expression is exactly what we were looking for in the table in the beginning, an upper and lower bound on regret that has dependence on $\mathrm{LDim}(\mathrm{H})$. We can therefore write the *average regret* in the online setting for infinite classes is bounded by

$$\frac{\textsc{Regret}}{T} \in \tilde{\Theta}\left(\sqrt{\mathrm{LDim}(\mathcal{H})/T)}\right)$$

where the $\tilde{\Theta}$ hides the logarithmic term in T.

The main takeaway from this section is that statistically, online and offline are very different! In the next section, we will look at offline learning algorithms and how they perform in the online setting.

3 Offline Algorithms in the Online Setting

We already know ERM works well for offline learning setting. How well will it work for online learning setting? The following theorem shows that for some online sequences, ERM or any other deterministic algorithm has very bad guarantee for regret.

Theorem 3.1. *For any deterministic algorithm (including ERM) $\mathcal{A}$, there is an online sequence such that* $\textsc{Regret}(\mathcal{A}) \geq \left(1 - \frac{1}{n}\right) T$*, where n is the number of experts in the problem.*

Proof. At each iteration t, since $\mathcal{A}$ is deterministic, the adversary can design his cost function as follows: $c^t(i^t) = 1$ if i^t is the expert that the player will choose by $\mathcal{A}$ and $c^t(i) = 0$ for all $i \neq i^t$. In this case, $\sum_{t=1}^{T} c^t(i^t) = T$, while $\min_{i \in [n]} \sum_{t=1}^{T} c^t(i) \leq \frac{T}{n}$ because at least one of the n experts appeared in less than T/n time steps. Thus $\textsc{Regret}(\mathcal{A}) \geq T - \frac{T}{n} = \left(1 - \frac{1}{n}\right) T$. □

This theorem shows the performance of ERM against an worst-case adversary. What made this sequence very challenging to learn was that the best expert so far, i.e., ERM's outcome, kept changing at most time steps. That is, the algorithm was running behind the ERM at the next step. Indeed, this is the main challenge when it comes to online learning. To begin, let's see the following lemma.

Lemma 3.2. *If the strategy of the player is ERM, i.e., $x^t = \arg\min_x \sum_{\tau=1}^{t-1} c^\tau(x)$, then*

$$\textsc{Regret}(\mathsf{ERM}) = \sum_{t=1}^{T} c^t\left(x^t\right) - \min_x \sum_{t=1}^{T} c^t\left(x\right) \leq \sum_{t=1}^{T} \left(c^t\left(x^t\right) - c^t\left(x^{t+1}\right)\right).$$

Before the proof, let us first think about what we can get from this lemma. Note that if there are only k times steps, for which $x^t \neq x^{t+1}$, then $\textsc{Regret}(\mathsf{ERM}) \leq k$. Moreover, even if $x^t \neq x^{t+1}$, but c^t has good Lipschitz property and x^{t+1} doesn't change too much from x^t, the lemma above will still give us a meaningful bound. We see more of this next time.

Proof. The required bound can be rewritten as

$$\sum_{t=1}^{T} c^t\left(x^{t+1}\right) \leq \min_x \sum_{t=1}^{T} c^t\left(x\right) = \sum_{t=1}^{T} c^t\left(x^{T+1}\right),$$

where the second equality holds by the definition of x^{T+1}. We prove this new inequality by induction. When $T = 1$, the claim follows by definition. Assume

$$\sum_{t=1}^{T-1} c^t\left(x^{t+1}\right) \leq \min_x \sum_{t=1}^{T-1} c^t(x).$$

Then,

$$\begin{aligned}
\sum_{t=1}^{T} c^t\left(x^{t+1}\right) &= \sum_{t=1}^{T-1} c^t\left(x^{t+1}\right) + c^T\left(x^{T+1}\right) \\
&\leq \min_x \sum_{t=1}^{T-1} c^t(x) + c^T\left(x^{T+1}\right) && \text{(Induction Hypothesis)} \\
&\leq \sum_{t=1}^{T-1} c^t(x^{T+1}) + c^T\left(x^{T+1}\right) && \text{(replacing } x \leftarrow x^{T+1}\text{)} \\
&= \sum_{t=1}^{T} c^t\left(x^{T+1}\right) \\
&= \min_x \sum_{t=1}^{T} c^t(x).
\end{aligned}$$

as required. □

References

Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *Proceedings of the 22nd Annual Conference on Learning Theory (COLT)*, 2009.