CPSC 532D — 10. ONLINE LEARNING: REALIZABLE BINARY CLASSIFICATION

Danica J. Sutherland and Bingshan Hu

University of British Columbia, Vancouver

Fall 2025

We'll now complete our study of what's learnable specifically for binary classification, by doing a change of setting.

So far, we've been considering *batch* or *offline* learning settings: we first get a training set S of m (x, y) pairs, pick a hypothesis h, and then use that h to predict forever on novel xs.

Online learning settings instead model the process of learning over time. An x comes in, we make a prediction based on our current understanding of the world, and then we get to see whether we were right. Then we update our beliefs about the world, and the process repeats, potentially indefinitely.

We still have an instance space $\mathcal{X}$, a label space $\mathcal{Y} = \hat{\mathcal{Y}} = \{-1, 1\}$, a hypothesis class $\mathcal{H}$ of functions $\mathcal{X} \to \mathcal{Y}$, and the 0-1 loss function $l_y(\hat{y}) = \mathbb{1}(y \neq \hat{y})$. We frame this as a sequential game, with two players: Learner, who makes predictions (us), and Nature, who provides us with what's going on in the world.

In each round $t = 1, 2, \ldots, \mathrm{T}$,
1. Nature selects $x_t \in \mathcal{X}$ and reveals it to Learner.
2. Learner predicts $\hat{y}_t \in \hat{\mathcal{Y}}$.
3. Nature plays label $y_t \in \mathcal{Y}$ and reveals it to Learner.
4. Learner suffers loss $l_{y_t}(\hat{y}_t) = \mathbb{1}(y_t \neq \hat{y}_t)$.

There are of course variations other than binary classification; we'll discuss some later on.

If the loss is 1, we say Learner *makes a mistake* in round t. The goal of Learner is to make as few mistakes as possible.

Quite differently than our analysis in the batch/offline setting, we *won't* assume that the data sequence S is iid. In fact, we'll have no statistical assumption at all.

But then it's hopeless to do anything in general; Nature could always simply play $\hat{y}_t = -y_t$. So we'll need to put some constraints on what Nature is allowed to do.

For today, we'll start with something simple: we'll assume that Nature has promised Learner that it will play labels consistently with some $h^* \in \mathcal{H}$, for a $\mathcal{H}$ known to Learner. Now, we can hopefully actually learn something... ideally, we'd eventually identify the correct h^*, and then necessarily never make a mistake again. But how many mistakes might we make before that point?

10.1 FINITE $\mathcal{H}$

We know that Nature is constrained to always play according to some perfect hypothesis; we just don't know what it is. But if we see an (x, y) pair for which

For more, visit https://cs.ubc.ca/~dsuth/532D/25w1/.

$y \neq h(x)$, then we *know* that h^* cannot be h. This suggests the idea of maintaining a set of which hypotheses might be correct. This is usually called a *version space*.

After t rounds, the version space is the set of hypotheses consistent with what we've seen so far:

$$\mathcal{H}_t = \{h \in \mathcal{H} : \forall i \in [t],\ h(x_t) = y_t\}.$$

We start with $\mathcal{H}_0 = \mathcal{H}$, then have $\mathcal{H}_1 \subseteq \mathcal{H}_0$, $\mathcal{H}_2 \subseteq \mathcal{H}_1$, and so on. It's never possible for us to eliminate h^*, so we always have $\{h^*\} \subseteq \mathcal{H}_t$.

This idea gives us an algorithm, called Consistent, when $\mathcal{H}$ is finite:

Set $\mathcal{H}_0 = \mathcal{H}$.
In each round $t = 1, 2, \ldots,$
1. Learner observes x_t.
2. Learner chooses any $h_t \in \mathcal{H}_{t-1}$ and predicts $\hat{y}_t = h_t(x_t)$.
3. Nature reveals the true label $y_t = h^*(x_t)$.
4. Learner updates $\mathcal{H}_t = \{h \in \mathcal{H}_{t-1} : h(x_t) = y_t\}$.

How many mistakes can we make in this process? That is, how big is $\sum_{t=1}^{T} l_{y_t}(\hat{y}_t)$? If we get really lucky, we just happen to pick h^* at first, and we never make *any* mistakes. But in the worst case, at each step, either we were right (and so add zero mistakes), or we we were wrong, in which case $\mathcal{H}_t$ eliminates at least one hypothesis from $\mathcal{H}_{t-1}$ (the one we played). This second case can only possibly happen $|\mathcal{H}| - 1$ times, since we always have $\{h^*\} \subseteq \mathcal{H}_t$. So, no matter how long we play this game, Nature can only force us to make at most $|\mathcal{H}| - 1$ mistakes.

Can we do better than that? Absolutely. The idea is to make sure that if we make a mistake, it gives us a *lot* of information. We can do this if, rather than choosing our prediction according to an arbitrary h_t, we take a *majority vote*. (We can break ties arbitrarily.) Then, either we're right (and so make zero mistakes), or we eliminate *at least half* of the hypotheses in the version space, since a majority of them were wrong. This gives us an algorithm called Halving.

Set $\mathcal{H}_0 = \mathcal{H}$.
In round $t = 1, 2, \ldots,$
1. Learner observes x_t.
2. Learner predicts $\hat{y}_t \in \arg\max_{y \in \hat{\mathcal{Y}}} \left|\{h \in \mathcal{H}_{t-1} : h(x_t) = y\}\right|$.
3. Nature reveals the true label $y_t = h^*(x_t)$.
4. Learner updates $\mathcal{H}_t = \{h \in \mathcal{H}_{t-1} : h(x_t) = y_t\}$.

Theorem 10.1. *The algorithm Halving makes at most* $\log_2|\mathcal{H}|$ *mistakes.*

Proof. After each mistake, the version space is at most half of its previous size. Letting M be the total number of mistakes at time T, this gives us that

$$|\mathcal{H}_T| \leq |\mathcal{H}_0| 2^{-M} = |\mathcal{H}| 2^{-M}.$$

But since $h^* \in \mathcal{H}_T$, $|\mathcal{H}_T| \geq 1$, and so this means that we must have $M \leq \log_2|\mathcal{H}|$. □

10.2 POTENTIALLY INFINITE $\mathcal{H}$

While $\log_2|\mathcal{H}|$ is nice when $\mathcal{H}$ is small, in the batch setting, we could learn lots of infinite hypothesis classes too: those with finite VC dimension. There turns out to be a similar notion that characterizes online learnability, called the *Littlestone dimension* [Lit88], $\text{Ldim}(\mathcal{H})$. We'll see that Nature is going to be able to force any learning algorithm to make at least $\text{Ldim}(\mathcal{H})$ mistakes, no matter the algorithm. On the other hand, we'll actually see a particular algorithm (though it may be very hard to implement) that always makes no more than $\text{Ldim}(\mathcal{H})$ mistakes.

Definition 10.2. A *game tree* is a binary tree describing the possible behaviours of Nature. Each interior node of the tree is associated with a point $x \in \mathcal{X}$; moving from a parent node to its left child is associated with the label -1, while moving to its right child is associated with the label 1. A *path* through the game tree is a sequence $S = \Big((x_1, y_1), (x_2, y_2), \ldots, (x_T, y_T)\Big)$, where x_1 is the point associated with the root node, y_1 is which edge we follow down from the root, x_2 is the point associated with either its left or right child depending on y_1, and so on.

To maximize the number of mistakes, we'll want to make sure that no matter what prediction Learner plays, Nature can always say "no, you're wrong," at least for a while. This is possible exactly if the tree is *shattered*:

Definition 10.3. A game tree is *shattered by* $\mathcal{H}$ if, for every path through the tree, there exists an $h \in \mathcal{H}$ which achieves $h(x_t) = y_t$ for all t in the path.

Definition 10.4. The *Littlestone dimension* of $\mathcal{H}$, denoted $\text{Ldim}(\mathcal{H})$, is the largest integer d such that there exists a tree of depth d shattered by $\mathcal{H}$. If there is no such largest d, the Littlestone dimension is infinite.

Theorem 10.5. *Nature can force any Learner to make at least* $\text{Ldim}(\mathcal{H})$ *mistakes.*

Proof. Consider a shattered tree of depth $\text{Ldim}(\mathcal{H})$. Nature will first play x_1 as the root label of the tree, then wait to see which prediction $\hat{y}_1$ Learner makes; it will declare $y_1 = -\hat{y}_1$, a mistake, then play x_2 according to the child node consistent with label y_1. This process continues down the tree, with Learner making a mistake at each of the $\text{Ldim}(\mathcal{H})$ rounds. Since $\mathcal{H}$ shatters the tree, there does indeed exist an $h^* \in \mathcal{H}$ consistent with the path through the game tree which Nature has followed. □

This gives us our lower bound. The next step to showing Littlestone dimension is the right characterization of online learnability (in the realizable binary classification setting) is to show an algorithm that only makes $\text{Ldim}(\mathcal{H})$ mistakes. This Standard Optimal Algorithm is exactly like Halving, except instead of asking which subset of the version space is larger, we ask which one has higher Littlestone dimension.

Set $\mathcal{H}_0 = \mathcal{H}$.
In round $t = 1, 2, \ldots,$
1. Learner observes x_t.
2. Learner predicts $\hat{y}_t \in \arg\max_{y \in \hat{\mathcal{Y}}} \text{Ldim}\left(\{h \in \mathcal{H}_{t-1} : h(x_t) = y\}\right)$.
3. Nature reveals the true label $y_t = h^*(x_t)$.
4. Learner updates $\mathcal{H}_t = \{h \in \mathcal{H}_{t-1} : h(x_t) = y_t\}$.

Theorem 10.6. *The Standard Optimal Algorithm makes at most* $\text{Ldim}(\mathcal{H})$ *mistakes.*

Proof. What we'll want to show is that if Learner makes a mistake in round t, i.e. if $\hat{y}_t \neq y_t$, then $\text{Ldim}(\mathcal{H}_t) \leq \text{Ldim}(\mathcal{H}_{t-1}) - 1$. Since $\text{Ldim}(\{h^*\}) = 1$, this will show our desired result.

We cannot have $\text{Ldim}(\mathcal{H}_t) > \text{Ldim}(\mathcal{H}_{t-1})$, so assume for the sake of contradiction that $\text{Ldim}(\mathcal{H}_t) = \text{Ldim}(\mathcal{H}_{t-1})$. Let's also name $\mathcal{H}_{t-1}^{(y)} = \{h \in \mathcal{H}_{t-1} : h(x_t) = y\}$.

Since Learner played $\hat{y}_t \neq y_t$, $\text{Ldim}(\mathcal{H}_{t-1}^{(\hat{y}_t)}) \geq \text{Ldim}(\mathcal{H}_{t-1}^{(y_t)})$. But since $\mathcal{H}_t = \mathcal{H}_{t-1}^{(y_t)}$, our assumption is that $\text{Ldim}(\mathcal{H}_{t-1}^{(y_t)}) = \text{Ldim}(\mathcal{H}_{t-1})$. We also know, since $\mathcal{H}_{t-1}^{(\hat{y}_t)} \subseteq \mathcal{H}_{t-1}$, that $\text{Ldim}(\mathcal{H}_{t-1}^{(\hat{y}_t)}) \leq \text{Ldim}(\mathcal{H}_{t-1})$. Thus $\text{Ldim}(\mathcal{H}_{t-1}^{(\hat{y}_t)}) = \text{Ldim}(\mathcal{H}_{t-1}) = \text{Ldim}(\mathcal{H}_{t-1}^{(y_t)})$.

Now, construct a tree with x_t at its root, one subtree a tree of depth $\text{Ldim}(\mathcal{H}_{t-1})$ shattered by $\mathcal{H}_{t-1}^{(\hat{y}_t)}$, and the other a tree of depth $\text{Ldim}(\mathcal{H}_{t-1})$ shattered by $\mathcal{H}_{t-1}^{(y_t)}$. This tree, which is of depth $\text{Ldim}(\mathcal{H}_{t-1}) + 1$, is shattered by $\mathcal{H}_{t-1}$: a contradiction. □

Proposition 10.7. *If $\mathcal{H}$ is finite,* $\text{Ldim}(\mathcal{H}) \leq \log_2|\mathcal{H}|$.

Proof. Combine Theorems 10.1 and 10.5. □

Proposition 10.8. *For any $\mathcal{H}$,* $\text{Ldim}(\mathcal{H}) \geq \text{VCdim}(\mathcal{H})$.

Proof. Let $\{x_1, \ldots, x_{\text{VCdim}(\mathcal{H})}\}$ be a set shattered by $\mathcal{H}$. Construct a complete binary tree with root x_1, both children x_2, all four grandchildren of the root x_3, and so on. This is a tree of depth $\text{VCdim}(\mathcal{H})$ which is shattered by $\mathcal{H}$. □

Proposition 10.9. *Let $\mathcal{X} = [0, 1]$ and $\mathcal{H} = \{x \mapsto \mathbb{1}(x \geq a) : a \in [0, 1]\}$ be the class of thresholds. Then, we have* $\text{VCdim}(\mathcal{H}) = 1$, *but* $\text{Ldim}(\mathcal{H}) = \infty$.

Proof. The VC dimension calculation was in Section 6.4.1.1.

For the Littlestone dimension: let x_* be the maximum of the points for which Nature has provided a negative label (or 0 if no negative labels have been given), and x^* the minimum of the points for which Nature has provided a positive label (or 1 if no positive labels have been given). Have Nature play $x_t = (x_* + x^*)/2$, and whichever label $\hat{y}_t$ is given by Learner, assign the label $-\hat{y}_t$, which is indeed realizable by a threshold between x_t and either x_* or x^*. Repeating this process indefinitely, Learner can always be made to make a mistake at each step. □

It turns out that actually, Littlestone dimension also characterizes which hypothesis classes can be *privately* PAC-learned [Alo+22; Lyu25].

REFERENCES

[Alo+22] Noga Alon, Mark Bun, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private and Online Learnability Are Equivalent. *Journal of the ACM* 69.4 (August 2022).

[Lit88] Nick Littlestone. Learning Quickly When Irrelevant Attributes Abound: A New Linear-Threshold Algorithm. *Machine Learning* 2 (1988), pages 285–318.

[Lyu25] Xin Lyu. *Private Learning of Littlestone Classes, Revisited*. 2025. arXiv: 2510.00076.