IE613: Online Machine Learning **Jan-Apr 2018**

Lecture 2: Introduction to Online Machine Learning

Lecturer: Manjesh K. Hanawal *Scribes: Ansuma Basumatary*

Note: The material in this note is extracted from Chapter 21 of the book 'Understanding Machine Learning: From Theory to Algortihm' by Shai Shalev- Shwartz and Shai Ben-David

2.1 Online Learning (Adversarial Setting)

What is online learning?

"Online Machine Learning is a method of machine learning in which data becomes available in a sequential order and is used to update our best predictor for future data at each step, as opposed to batch learning techniques which generate the best predictor by learning on the entire training data set at once." - Wikipedia

We call any algorithm that is of the form below as an Online Learning Algorithm (OLA). For simplicity we focus on binary lables and $0-1$ loss function and assume that the size of hypothesis class if finite.

General Online Learning Algorithm A

Input: Hypothesis class $\mathcal{H}$
for $t = 1, 2, 3, \cdots$ **do**
    Receive sample x_t
    Select an hypothesis $h \in \mathcal{H}$
    Predict label $\hat{y}_t = h(x_t)$
    Receive true label y_t and loss is $|\hat{y}_t - y_t|$
Output: Return a hypothesis $h \in \mathcal{H}$.

Note that we do not make any assumption on the way samples are generated – they could be generated stochastically, deterministically or even adversarially. The true label is revealed in every round after the prediction is made using some hypothesis. Depending on how the choice of an hypothesis is made in each round, we get different OLAs. Each OLA is like a game between an enviroment and a learner which procedes in rounds. In each round, the environment generates a sample and its associated label, and the learner predicts the label only seeing the sample.

For any given sequence $\mathcal{S} = \{(x_i, y_i) : i = 1, 2, \ldots, T\}$, where T is an integer, let $M_{\mathcal{S}}(A) = \sum_{i=1}^{T} |\hat{y}_t - y_t|$ denote the be the number of mistakes algorithm A makes on $\mathcal{S}$, where $\hat{y}_t$ is the prediction made by algorithm A in round t.

Definition 1 (Mistake Bound) *Let $M_{\mathcal{H}}(A) = \sup_{\mathcal{S}} M_{\mathcal{S}}(A)$ denote the maximum number of mistakes made by algorithm A. A bound of the form $M_{\mathcal{H}}(A) \leq B < \infty$ is called a mistake bound.*

Definition 2 (Online Learnability) *We say that the hypothesis class $\mathcal{H}$ is 'learnable' if there exists an algorithm A and constant $B < \infty$ (independent of S) such that $M_{\mathcal{H}}(A) < B$.*

We would be interested in an OLA that has the smallest B on $\mathcal{H}$. We first make the following assumption on the way labels are generated

Assumption 1 (Realizability) *All labels are generated by some hypothesis $h^* \in \mathcal{H}$, i.e., $y = h^*(x) \ \forall\, x$.*

Under the realizability assumption learning essentially boils down to search for the hypothesis in $\mathcal{H}$ that generates the labels. In this setting, one can think of using an online algorithm that in every round eliminates the hypotheses that are not consistent, i.e., make incorrect predictions. Based on this idea we have the following algorithm named Consistent Algorithm (CA). The CA algorithm maintain a set, V_t, of all the hypothesis that are consistent on the samples observed so far, i.e., $\{(x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t)\}$ and selects a hypothesis from this set for prediction in the next round.

Consistent Algorithm (CA)

Input: Hypothesis class $\mathcal{H}$
Initialize: $V_1 = \mathcal{H}$
for $t = 1, 2, 3, \cdots$ **do**
    Receive sample x_t
    Select an hypothesis $h \in V_t$
    Predict label $\hat{y}_t = h(x_t)$
    Receive true label y_t. Loss is $|\hat{y}_t - y_t|$
    Update $V_{t+1} = \{h \in V_t : h(x_t) = y_t\}$
Output: Return a hypothesis $h \in \mathcal{H}$.

Claim: The CA algorithm makes at most $|\mathcal{H}| - 1$ mistakes, i.e., $M_{\mathcal{H}}(CA) \leq |\mathcal{H}| - 1$.
Proof: Observe that if the CA algoithm makes M mistakes at the end of t rounds, it must be the case that $|V_t| \leq |\mathcal{H}| - M$, as atleast one hypothesis gets eliminated after a mistake. By realizability assumption we also know that $|V_t| \geq 1$ for all t. Hence maximum number of mistakes, $M_{\mathcal{H}}(CA)$, should be such that $1 \leq \mathcal{H} - M_{\mathcal{H}}(CA)$.

Can we do better than the CA algorithm? We next present an algorithm, named Halving Algorithm (HA) that has exponentially smaller mistake bound than the CA.

Halving Algorithm (HA)

Input: Hypothesis class $\mathcal{H}$
Initialize: $V_1 = \mathcal{H}$
for $t = 1, 2, 3, \cdots$ **do**
    Receive sample x_t
    Predict label $\hat{y}_t = \arg \max_{\gamma \in \{0,1\}} |\{h \in V_t : h(x_t) = \gamma\}|$
    Receive true label y_t. Loss is $|\hat{y}_t - y_t|$
    Update $V_{t+1} = \{h \in V_t : h(x_t) = y_t\}$
Output: Return a hypothesis $h \in \mathcal{H}$.

In any round t, the HA uses the set of hypothesis, V_t, that are consistent with the observations made so far and predicts the lable of sample x_t to that label which is the prediction made by the maximum number of the hypothesis in V_t. Thus, HA uses majority voting to predict labels in each round.

Claim: The HA algorithm makes at most $\log_2(|\mathcal{H}|)$ mistakes, i.e., $M_{\mathcal{H}}(HA) \leq \log_2(|\mathcal{H}|)$.
Proof: If a mistake is made in round t, it must be the case that $|V_{t+1} \leq |V_t|/2$ as atleast half of the hypotheses from V_t are dicarded. Also, by the realizability assumption we have $|V_t| \geq 1$ for all t. Applying this repeatedly, we get $1 \leq V_{T+1} \leq |V_1|2^{-M}$ if there are M mistakes after T rounds. Hence maximum number of mistakes $M_{\mathcal{H}}(MH)$ must satisfy $1 \leq |\mathcal{H}|2^{-M_{\mathcal{H}}(HA)}$. Rearranging, the claimed mistake bound is obtained.