CS747: Foundations of Intelligent and Learning Agents **July - Nov 2017**

Guest Lecture 1: August 23

Lecturer: Manjesh K. Hanawal *Scribe: 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

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

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

Relaxing The Realizability Assumption

In general the hypothesis that generates the true labels may not belong to the the hypothesis class $\mathcal{H}$ over which we want to learn, i.e., $h^* \notin \mathcal{H}$. In this case, we may aim to find the best hypothesis in $\mathcal{H}$ and the best hypothesis may not label all the samples correctly. In this case, we evaluate the performance of an OLA by comparing the number of mistakes it makes and that made by the best hypothesis in $\mathcal{H}$ under the worst case scenario. Specifically, *regret* of A that predicts label of x_t as $\hat{y}_t$ in round t is defined as follows:

$$R_{\mathcal{H}}(A,T) = \sup_{(x_1,y_1),\ldots,(x_T,y_T)} \left[\sum_{t=1}^{T} |\hat{y}_t - y_t| - \inf_{h\in\mathcal{H}} \sum_{t=1}^{T} |h(x_t) - y_t| \right].$$

Definition 3 *We say that the hypothesis class $\mathcal{H}$ is learnable if there exists and an algorithm A such that $\lim_{T\to\infty} \frac{R_{\mathcal{H}}(A,T)}{T} = 0$, i.e., the regret of A on $\mathcal{H}$ is sub-linear.*

Our goal is to find an OLA with smallest sub-linear regret relative to $\mathcal{H}$. Is this goal achievable? Notice that in the general OLA we described, the true label is revealed after the predictions is made in every round. If the labels are generated by an adversary, he can make the algorithm err in every rounds by just declaring the true label to be opposite of what is predicted, i.e., $y_t = 1 - \hat{y}_t$. Then the total number of mistakes made by any algorithm over T rounds is T. This can result in linear regret for any OLA. To see this, consider a simple hypothesis class with two hypotheses, $\mathcal{H} = \{h_0, h_1\}$, where hypothesis h_0 predicts label 0 and h_1 predicts 1 on all samples. For any sequence of labels $y_1, y_2, \cdots, y_T$, we have

$$\inf_{h\in\mathcal{H}} \sum_{t=1}^{T} |h(x_t) - y_t| = \min\left\{ \sum_{i=1}^{T} y_t, T - \sum_{i=1}^{T} y_t \right\} \leq T/2.$$

Hence, for any OLA A, $R_{\mathcal{H}}(A,T) \geq T - T/2 \geq T/2$, i.e., linear in T and sub-linear regret cannot be achieved.

To overcome this impossibility result, one needs to restrict the power of the environment or (the adversary). We do so by restricting the environment to decide the label in each round without knowing the prediction made by the learner in that round. Further, we allow the learner to use randmonized strategies to make predictions in each round, i.e., the learner can toss a coin of certain bias and decide on the prediction based on the outcome of the toss. This further restricts the power of the environment – if the learner uses a deterministic strategy in each round, the environment can figure it out (from the past observations) and change its label generation strategy so that the strategy adopted the learner makes more mistakes. In summary, we make the following assumptions:

1. The learner can randomize the predictions.

2. The environment has to decide y_t without knowing the actual outcome of learner's random prediction in round t.

Since we allow the learner to randomize the predictions, we henceforth consider the expected regret denoted as $\mathbb{E}[R_{\mathcal{H}}(A,T)]$, where the expectation is over the randomness of the algorithm (or learner's strategy). For notational convenience, we drop the subscript $\mathcal{H}$ and write it as $\mathbb{E}[R(A,T)]$, the underlying hypothesis class to be learned should be clear from the context.

Let $p_t = \Pr\{\hat{y}_t \neq y_t\}$ denote the probability that the learner predicts label as 1 in round t. This probability could depend on all the past observations made by the learner. We have $\mathbb{E}[|\hat{y}_t - y_t|] = |p_t - y_t|$. Indeed, $\mathbb{E}[|\hat{y}_t - y_t|] = p_t|1 - y_t| + (1 - p_t)|0 - y_t|$. If $y_t = 0$, we get $\mathbb{E}[|\hat{y}_t - y_t|] = p_t = |p_t - 0|$ and if $y_t = 1$, we get $\mathbb{E}[|\hat{y}_t - y_t|] = 1 - p_t = |p_t - y_t|$. The two cases can be compactly written as $\mathbb{E}[|\hat{y}_t - y_t|] = |p_t - y_t|$. Then, the expected regret is given by

$$\mathbb{E}[R(A,T)] = \sup_{(x_1,y_1),\ldots,(x_T,y_T)} \left[\sum_{t=1}^{T} |\hat{p}_t - y_t| - \inf_{h \in \mathcal{H}} \sum_{t=1}^{T} |h(x_t) - y_t| \right].$$

This modified set up can be interpreted as relaxing the predictions $\hat{y}_t$ to take values in $[0,1]$ instead of either 0 or 1 only. Is sub-linear regret possible for this setup?

Theorem 1.1 *For every hypothesis class $\mathcal{H}$, there exits an OLA, A, whose predictions come from [0,1] and has regret bound such that,*

$$\forall h \in \mathcal{H},\ \mathbb{E}[R_{\mathcal{H}}(A,T)] \leq \sqrt{2\log(|\mathcal{H}|) \cdot T} \approx O(\sqrt{T}).$$

Furthermore, no algorithm can achieve an expected regret bound smaller tha $\Omega(\sqrt{T})$.

Weighted Majority Algorithm

The Weighted Majority (WM) algorithm maintains weights $(\tilde{w}_i^{(t)})$ for each of the hypotheses. A hypothesis is selected in a round according to a probability $(w_i^{(t)})$ that is in proportion to its weight. After a hypothesis is selected in round t, the true label is revealed from which los vectors for all the hypotheses $l_t\{l_{t,i}, i \in [d]\}$ is obtained. For example, loss for i^{th} hypothesis can be set as $l_{t,i} = |h_i(x_t) - y_t|$. Then the expected loss in round t is given by $|p_t - y_t| = \sum_{i=1}^{d} w_i^{(t)} l_{t,i} =< w^{(t)}, l_t >$. Depending on the loss values, the weights $(\tilde{w}^{(t)})$ are updated. The pseudo-code of the WM is given below.

Weighted Majority (WM)

Input: Hypothesis class $\mathcal{H}$ and number of rounds T
Parameter: $\eta = \sqrt{2\log(|\mathcal{H}|)/T}$
Initialize: $\tilde{w}^{(1)} = (1, 1, \cdots, 1)$
for $t = 1, 2, 3, \cdots$ **do**
    Set $w_i^{(t)} = \frac{\tilde{w}_i^{(t)}}{\sum \tilde{w}_i^{(t)}}\ \forall i = 1, 2, ..., d,$ where $|\mathcal{H}| = d$
    Receive loss vector $l_t := \{l_{t,i} : i \in [d]\}$
    Compute expected cost $< w^{(t)}, l_t >$
    Update $\forall i,\ \tilde{w}_i^{(t+1)} = \tilde{w}_i^{(t)} \cdot e^{-\eta l_{t,i}}$

Theorem 1.2 *Let $d = |\mathcal{H}|$ and $T > 2\log(d)$, then $R(WM,T) \leq \sqrt{2\log(d)T}$, i.e., WM is order optimal.*

The above the algorithms are based on *full Information* setting as losses of all hypotheses are known in each round. The setting is also referred to as prediction with expert advice.