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

Lecture 3: Online Learning in Unrealizable Case

Lecturer: M. K. Hanawal *Scribes: Harshvardhan Tibrewal*

NOTATION
A hypothesis class H containing a set of hypothesis is known beforehand.
We propose an algorithm A which runs over T rounds on sequences of data $(x_t, y_t) \in S$
The algorithm A predicts labels of input data x_t as $\hat{y_t}$

3.1 Recap

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.

ASSUMPTION: Data labels are generated from a hypothesis in the hypothesis class H known to us.
We have to find an algorithm A that can learn the hypothesis that predicts correct labels of the given sequences of data as input.

Below mentioned is a brief of two algorithms seen in class earlier

1) Consistent Algorithm (CA) :

$$V_{t+1} = \{h \in V_t,\ h(x_t) = y_t\} \tag{3.1}$$

where $h_t \in V_t$, $\hat{y_t} = h_t(x_t)$ for hypothesis class H and $V_1 = H$
Hence at each step, the algorithm discards a hypothesis in the set H that predicts an incorrect label to the data

EXERCISE : $\exists H$ *such that* $M_H(CA) = |H| - 1$

$M_H(CA)$ is the mistake bound of algorithm CA on hypothesis class H.
At each step of the consistent algorithm, if a hypothesis $h \in H$ makes a mistake ($\hat{y_t} \neq y_t$) that hypothesis h is discarded from the set H.

If V_1 is the initial set of hypothesis $= H$ and let V_{t+1} be the hypothesis set remaining after round t, then the relation $1 \leq |V_{t+1}|$ holds.
Hence we can at most make $M_H(CA) = |H| - 1$ mistakes as we converge to the best hypothesis when only one hypothesis is remaining while discarding the rest.

2) Halving Algorithm (HA) :

$$V_{t+1} = \{h \in V_t,\ h(x_t) = y_t\} \tag{3.2}$$

where $\hat{y_t}$ is based on majority voting and $V_1 = H$
Hence at each step, the algorithm finds a subset of hypotheses giving incorrect predictions and discards them.

EXERCISE : $\exists H$ *such that* $M_H(HA) = log_2|H|$

$M_H(HA)$ is the mistake bound of algorithm HA on hypothesis class H. At each step of the halving algorithm, we do majority voting.

That is, we see all hypotheses and form a largest set (say W) of hypotheses giving same $\hat{y_t}$.

If V_1 is the initial set of hypothesis $= H$ and let V_t be the hypothesis set remaining after round $t-1$,

- if ($\hat{y_t} \neq y_t$) for this set, then all the hypotheses in this set are discarded and remaining $V_{t+1} = V_t - W$ is retained for the next round.
- else if ($\hat{y_t} = y_t$) then this is not a mistake and only $V_{t+1} = W$ is retained for the next round.

In summary, at each step of the algorithm the size of the remaining hypthosis set is at least halved when there is a mistake.

If we see M mistakes till round t, then

$$1 \leq |V_{t+1}| \leq |V_1|2^{-M} \implies M \leq log_2|V_1| \implies M_H(HA) \leq log_2|H|$$

3.2 Learning in unrealizable case

If $h^* \notin H$, that is the data is not generated from a hypothesis in the class H

Generation of data (x_t, y_t): Incoming sequences of data could have been generated as per the following cases

- $x_t \sim D$, $y_t = h^*(x_t)$ - Stochastic case
- $(x_t, y_t) \sim D$, jointly distributed - Stochastic case
- $(x_t, y_t) \sim D_t$, joint distribution changes over rounds (t) called as *Adversarial case*

$(x_t, y_t) \sim D_t$ generated by nature/adversary in round t
Learner applies Online Machine learning to solve for best predictor

Realizabililty assumption : $y_t = h^*(x_t)$

In adversarial case, the realizability assumption is relaxed (not true)
No longer the labels are generated from the hypothesis class, we require learner to be competitive with the

best predictor in H

Adversarial examples are synthetic examples constructed by modifying real examples slightly in order to make a classifier believe they belong to the wrong class with high confidence
Our evaluation criteria is now **minimization of Regret** (how far the learner is from the best predictor H)

DEFINITION: For a hypothesis class H, running over period T, on a sequence of points S and predictions made by algorithm A the regret is defined as:

$$R_H(A,T) := \sup_{(x_1,y_1),\ldots(x_T,y_T)\in S} \Big\{ \sum_{t=1}^{T} |\hat{y_t} - y_t| - \inf_{h\in H} \sum_{t=1}^{T} |h(x_t) - y_t| \Big\} \tag{3.3}$$

Data - a new sequence of points, so for i^{th} sequence S_i, $(x_1^i, y_1^i), \ldots (x_T^i, y_T^i) \in (x^i, y^i)^T$
where $\hat{y_t}$ denotes the predictions by algorithm A in round t
The first term signifies total loss of Algorithm(A) on the incoming data.
The second term denotes best possible predictions, as infimum over all hypotheses are taken over the set H
We have taken supremum as we want the worst possible regret possible for an algorithm and a given hypothesis set H over multiple data sequences $(x^i, y^i)^T$
Where $\hat{y_t}$ is predicted by algorithm A.

DEFINITION: Learnability
We say that the hypothesis class is learnable if $\exists$ an algorithm such that

$$\lim_{T\to\infty} \frac{R_H(A,T)}{T} = 0 \tag{3.4}$$

That is regret $R_H(A,T)$ should be sub-linear asymptotically.
Average number of mistakes by the algorithm should go to zero in the limit
Example: if $R_H(A,T) = O(T^{0.5})$, then for the algorithm A, the hypothesis class is learnable

Is it always possible to achieve sub-linear regret ?
Let $H = \{h_0, h_1\}$, where $h_0 = 0$ and $h_1 = 1$
Then

$$\inf_{h\in H} \sum_{t=1}^{T} |h(x_t) - y_t| = \min \Big\{ \sum_{t=1}^{T} y_t,\ T - \sum_{t=1}^{T} y_t \Big\}$$

The best predictor in H can hence make at most $T/2$ errors

The adversary can always choose y_t different from what algorithm A predicts $= 1 - \hat{y_t}$ as the adversary has access to current prediction of algorithm

$$\sum_{t=1}^{T} |\hat{y_t} - y_t| = T$$

Hence,

$$R_H(A,T) \geq T - T/2 \implies R_H(A,T) \geq T/2$$

Hence, regret is grows linearly (not sub-linearly)

CLAIM: *In general, for an unrestricted, adversarial environment, without realizability, it is impossible to find a Leaning Algorithm with a Sub-Linear Regret*

Restrictions imposed on adversary

- The environment/adversary has to decide y_t without knowing the outcome of the learner's prediction $\hat{y_t}$. That is, our prediction is not known by adversary beforehand
- The learner can randomize predictions. In this case we calculate **expected regret**

$$E[\ R_H(A,T)\ ] := \sup_{(x,y)^T \in S} \Big\{ \sum_{t=1}^{T} |\hat{P}_t - y_t| - \inf_{h \in H} \sum_{t=1}^{T} |h(x_t) - y_t| \Big\} \tag{3.5}$$

  where $\hat{P}_t = P(\hat{y_t} = 1)$ (for binary classification) as $E[\ |\hat{y_t} - y_t|\ ] = |\hat{P}_t - y_t|$ and S is set of sequences of data