Online Classification – Part 1

Source:

Online Learning and Online Convex Optimization, by Shai Shalev-Shwartz, Found. and Trends in ML, Vol. 4, No. 2, 2011 (Chapters 1, 3).

Setting: feature space $\mathcal{X}$. Classifiers: $\mathcal{H}$
Label space $\mathcal{Y}$

$h \in \mathcal{H}$: $h : \mathcal{X} \longrightarrow \mathcal{Y}$ i.e., $x \in \mathcal{X}$, $y \in \mathcal{Y}$

$h(x) = y$, a classifier.

Typically, $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{0, 1\}$ (or $\{-1, +1\}$)

$\longrightarrow$ binary classification

Online problem — evolves in discrete time steps, $t = 1, 2, \ldots, T$.

At each time step $t = 1, 2, 3, \ldots, T$:

Adversary / Environment: Choose $x_t \in \mathcal{X}$.

Player: Predicts a label $p_t \in \mathcal{D}$ (prediction space, either $\mathcal{D} = \{0, 1\}$ (deterministic algo) or $\mathcal{D} = [0, 1]$ (randomized algo).)

Adversary: Shows true answer $y_t \in \mathcal{Y}$.

Loss to player: $\ell(p_t, y_t)$

Goal: Minimize regret with respect to the best fixed classifier in hindsight.

i.e., $\text{Regret}_T(h^*) = \sum_{t=1}^{T} \ell(p_t, y_t) - \sum_{t=1}^{T} \ell\left(h^*(x_t), y_t\right)$

- ℓ: loss function
- $h^* \in \mathcal{H}$: $\mathcal{H}$ = hypothesis class (set of possible classifiers)
- h^*: best fixed classifier in hindsight

π a policy, i.e., $\pi = \pi_1, \pi_2, \ldots \pi_T$ of player,

where $\pi_t : I_{t-1} \longrightarrow \mathcal{H}$ → hypothesis class.

I_{t-1} → information until $(t-1)$

$= \{(x_1, y_1), (x_2, y_2), \ldots, (x_{t-1}, y_{t-1})\}$

Mistake bound: $M_\pi(\mathcal{H})$ = maximum number of mistakes that π can make when applied to the hypothesis class $\mathcal{H}$.

First Take: $Y = D = \{0, 1\}$, X arbitrary.

$\ell(p, y) = |p - y|$, $p, y \in \{0, 1\}$.

Finite Hypothesis Class, i.e., $|\mathcal{H}| < \infty$.

$$\text{Regret}_T(\mathcal{H}) = \max_{h \in \mathcal{H}} \left(\sum_{t=1}^{T} |p_t - y_t| - \sum_{t=1}^{T} |h(x_t) - y_t| \right)$$

Claim: We cannot get sub-linear (in T) regret even when $|\mathcal{H}| = 2$, with $h_0(x_t) = 0 \;\forall\; x_t$

$h_1(x_t) = 1 \;\forall\; x_t$.

(Cover's counter-example).

Recall setting: Adversary chooses x_t, knows policy of player in choosing h_t, and choose y_t. In this case, player is deterministic, so, adversary knows y_t.

— Adversary always chooses $y_t \neq p_t$.

$$\therefore \quad \sum_{t=1}^{T} |p_t - y_t| = T.$$

However $$\min_{h \in \{h_0, h_1\}} \sum_{t=1}^{T} |h(x_t) - y_t| \leq \frac{T}{2}$$

(by pigeonhole principle).

$$\therefore \quad R_T(\mathcal{H}) \geq \frac{T}{2} \implies \text{linear regret.}$$

To control this, either realizable or randomize.

Realizability $\exists\ h^* \in \mathcal{H}$ s.t. $h^*(x_t) = y_t$

$\forall\ t = 1, 2, \ldots, T$. Thus, we are assuming that there exists a perfect (no-mistake) classifier within $\mathcal{H}$, the allowable set of classifiers.

Goal in realizability setting: minimize $M_\pi(\mathcal{H})$.
π
policy.

Second Take Finite hypothesis class $\mathcal{H}$ + Realizability

Consistent Policy (Algorithm):

Input: $\mathcal{H}$, with $|\mathcal{H}| < \infty$, $h^* \in \mathcal{H}$, with $h^*(x_t) = y_t$

Players:

Initial: $V_1 = \mathcal{H}$.

At each time $t = 1, 2, \ldots, T$:

(i) Observe $x_t \in X$. (chosen by adversary)

(ii) Arbitrarily choose $h_t \in V_t$, and predict $p_t = h_t(x_t)$.

(iii) Observe $y_t \in Y$ (chosen by adversary using h^*)

(iv) Update $V_{t+1} = \{ h \in \mathcal{H} : h(x_t) = y_t \text{, for all observed } (x_t, y_t) \}$

Observe that the consistent algorithm eliminates at least one hypothesis every time a mistake is observed, i.e., if $h_t(x_t) \neq y_t$, then $V_{t+1} \notin h_t$.

$\therefore$

$$1 \leq |V_t| \leq |\mathcal{H}| - M$$

(M: number of mistakes made)

$$\Rightarrow M \leq |\mathcal{H}| - 1$$

$$\therefore M_{\text{consistent}}(\mathcal{H}) \leq |\mathcal{H}| - 1$$

A better algorithm is the halfing algorithm that

uses all remaining hypotheses in V_t, predicts using the majority recommendation, and thus can eliminate atleast half the classifiers each time a mistake is made (instead of only 1 in the consistent algo.).

Halving Policy (Algorithm):

Input: $\mathcal{H}$, with $|\mathcal{H}| < \infty$, $h^* \in \mathcal{H}$, with $h^*(x_t) = y_t$

Player:

Initial: $V_1 = \mathcal{H}$.

At each time $t = 1, 2, \ldots, T$:

(i) Observe $x_t \in X$. (chosen by adversary)

(ii) Predict $p_t = \underset{r \in \{0,1\}}{\operatorname{argmax}} \left| \{h \in V_t : h(x_t) = r\} \right|$

majority over all remaining classifiers

(iii) Observe $y_t \in Y$ (chosen by adversary using h^*)

(iv) Update $V_{t+1} = \{ h \in \mathcal{H} : h(x_t) = y_t$, for all observed $(x_t, y_t) \}$

Note that every mistake eliminates atleast half the classifiers (because the majority is chosen), i.e., $V_{t+1} \leq \frac{|V_t|}{2}$ for each mistake.

$$\Rightarrow \quad 1 \leq |V_{T+1}| \leq |\mathcal{H}| 2^{-M}$$

$$\Rightarrow \quad M_{\pi_{\text{halving}}}(\mathcal{H}) \leq \log_2 |\mathcal{H}|$$

Third Take Randomization of prediction using secret bits to make decision.

Model: $x_t \in X$, $y_t \in \{0,1\}$, $p_t \in [0,1]$

$D = [0,1]$ instead of $\{0,1\}$.

$\ell(p_t, y_t) = |p_t - y_t|$ → expected loss → Convex domain, convex loss fn.

i.e., predict $\hat{y}_t = 1$ w.p. p_t. Then, observe that

$P(\hat{y}_t \neq y_t) = |p_t - y_t|$ → expected # of loss at time t.

(why? $y_t = 0$: $P(\hat{y}_t = 1) = p_t = |p_t - y_t|$
$y_t = 1$: $P(\hat{y}_t = 0) = (1 - p_t)$
$= |p_t - 1|$)

Thm: $\sum_{t=1}^{T} |p_t - y_t| - \min_{h \in \mathcal{H}} \sum_{t=1}^{T} |h(x_t) - y_t| \leq \sqrt{0.5 \ln(|\mathcal{H}|)\, T}$

$p_t = h_t(x_t)$ — expert

h — fixed expert

finite class

Connections to Set 13: (Exp4 and Oblivious Experts.)

Consider a 2-armed bandit, where arm 1 is labeled "predict 0" and arm 2 is labeled "predict 1". The adversary chooses (x_t, y_t), at time t. x_t is the context (more on that below), and y_t is the label. If $y_t = 0$, assign loss for arms as $(0, 1)$ and assign $(1, 0)$ if $y_t = 1$.

arm 1 ↩ ↪ arm 2

Each hypothesis / classifier $h \in \mathcal{H}$ is an expert. The experts are deterministic, i.e., uses the prob. vector

$$E_h^{(t)} = \begin{pmatrix} 1 - h(x_t) & h(x_t) \end{pmatrix}, \quad h \in \mathcal{H} \quad (|\mathcal{H}| \text{ is finite})$$

The algorithm chooses expert h w.p. $q_h^{(t)}$ (i.e., Q_t is the vector over experts) Thus, we effectively

have a vector $(1-p_t \quad p_t)$, corresponding to the prob we play the two arms at each time. Then, the expected loss at any time is as follows:

$$Z_t \sim p_t \text{ , i.e., } Z_t = \begin{cases} 1 & \text{w.p } p_t \\ 0 & \text{w.p } 1-p_t. \end{cases}$$

(p_t: function of history)

$$L_t = \chi_{\{Z_t \neq y_t\}} \Rightarrow E\left[L_t \mid I_{t-1}\right] = P\left(Z_t \neq y_t \mid I_{t-1}\right)$$

$$= |p_t - y_t|.$$

Thus, from the oblivious expert setting and analysis, we have Regret $\sim \sqrt{\ln(|\mathcal{H}|) \cdot T}$.

Weighted Majority Algorithm: This is the same as Exp4 without IS estimation.

$\eta \in (0,1)$: learning rate

$\mathcal{H} = \{h_1, h_2, \cdots, h_d\}$ — d experts / classifiers / hypotheses.

$q = (q_1 \cdots q_d)$, a weight vector on the simplex.

$\in [0,1]$ ← $\sum_{i=1}^{d} q_i^{(t)} h_i(x_t)$ → $\{0,1\}$.

$$|p_t - y_t| = \left| \sum_{i=1}^{d} q_i^{(t)} h_i(x_t) - y_t \right|$$

$$= \sum_{i=1}^{d} q_i^{(t)} |h_i(x_t) - y_t|$$

Initial: $q_1 = \left(\frac{1}{d} \cdots \frac{1}{d}\right)$.

At time t: predict using q_t, i.e., draw a random expert / hypothesis i using q_t, and predict using it's advice. Then, update:

$$q_i^{(t+1)} = \frac{q_i^{(t)} e^{-\eta \cdot z_i^{(t)}}}{\sum_j q_j^{(t)} e^{-\eta z_j^{(t)}}}$$

where $z_t = \left(|h_1(x_t) - y_t|, |h_2(x_t) - y_t|, \ldots \right.$

$\left. \ldots |h_d(x_t) - y_t| \right)$

= vector of losses under each possible expert (we have full feedback, as we can compute the loss under each expert).

Then, we can compute regret for this using analysis identical to Exp 4, or Prop. 28.7 in text (see

pp. 18 in Set 20 of notes).

Beyond Finite Hypothesis Classes: Realizable Case

$\exists$ a "no mistake" $h \in \mathcal{H}$.

Let π be a policy, and let $M_\pi(\mathcal{H})$ be the mistake bound with π. Formally,

pick any $h \in \mathcal{H}$, and let $S = \{(x_i, y_i),\ i = 1, 2, \ldots, T\}$ be a sequence of features, and with the corresponding labels $y_i = h(x_i)$. For this sequence S, let $M_{\pi,S}$ be the number of mistakes that policy π makes, i.e.,

$$M_{\pi,S} = \sum_{t=1}^{T} \mathbb{1}\{y_t \neq h_t^{\pi_t}(x_t)\}$$

$h_t^{\pi_t}$: classifier selected by π at time t.

Then, $M_\pi(\mathcal{H}) = \sup_{\substack{h \in \mathcal{H}, \\ |S| = T \\ \{x_1, \ldots, x_T\}}} M_{\pi,S}$

over all length T sequences with any fixed classifier $h \in \mathcal{H}$.

The Littlestone Dimension — Ldim($\mathcal{H}$)

- Environment chooses x_t
- Learner predicts label $p_t \in \{0, 1\}$
- Environment reveals true label $y_t \in \{0, 1\}$

Environment $\longrightarrow$ goal to induce mistakes on all T time steps. $\therefore$ $y_t = 1 - p_t$ is desired.

BUT: environment constrained to use some classifier $h \in \mathcal{H}$ to generate labels (realizable setting).

$\therefore$ Environment needs to carefully choose $\{x_1, \dots, x_T\}$ sequentially based on $\{p_1, p_2, \dots, p_{t-1}\}$, at each time t.

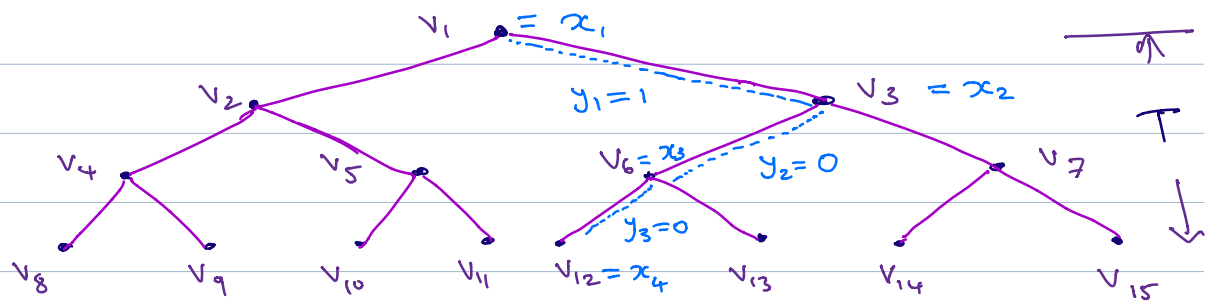
Tree: nodes $\{v_1, \dots, v_{2^T-1}\}$, a binary tree.

Initial: $x_1 = v_1$

At each time: $x_t = v_{i_t}$, i_t = current node in tree.

$\longrightarrow$ descend along left branch if $y_t = 0$

right branch if $y_t = 1$



i.e., $i_{t+1} = 2i_t + y_t$, $t = 1, 2, \ldots$

$\Rightarrow$ $i_t = 2^{t-1} + \sum_{j=1}^{t-1} y_j 2^{t-1-j}$, $t = 1, 2, \ldots, T$.

With this construction, it now follows that the environment can succeed in creating T mistakes for the learner if for any given $\{y_1, y_2, \ldots, y_T\}$ (a potential path on the tree from root to a leaf), $\exists\ h \in \mathcal{H}$ s.t. $y_t = h(x_t)$, $t = 1, 2, \ldots, T$.

$\mathcal{H}$-shattered Tree: The T-depth binary tree is $\mathcal{H}$ shattered if there exists an assignments of features $\{x_j\}$ to every node s.t. for any path from root to leaf, $\exists\ h \in \mathcal{H}$ generating the corresponding labels. In other words, $\exists\ \{v_1, v_2, \ldots, v_{2^T-1}\} \in \mathcal{X}$ s.t. for any

fixed path $\{v_{i_t}, t=1,2,\dots,T\}$, we have that $\exists h \in \mathcal{H}$ s.t. $h(v_{i_t}) = y_t$.

$L\dim$ = Littlestone Dimension: $L\dim(\mathcal{H})$ is the largest T s.t. the T-depth binary tree is $\mathcal{H}$-shattered.

Lemma (3.2, Chap 3): For any π, $M_\pi(\mathcal{H}) \geq L\dim(\mathcal{H})$.

Pf: Essentially, by defn. of $\mathcal{H}$-shatter. Specifically, $x_t = v_{i_t}$, and assign $y_t = 1 - p_t$. ∎

Corrollary 1: $L\dim(\mathcal{H}) \leq \log_2(|\mathcal{H}|)$, if $|\mathcal{H}|$ is finite.

Corrollary 2: $\mathcal{X} = \{1, 2, \dots, d\}$ and $\mathcal{H} = \{h_1, \dots, h_d\}$ with $h_d(x) = 1 \iff x = d$. Then, $L\dim(\mathcal{H}) = 1$.

Generalize the Halfing Algorithm

Recall: Each time, label with majority, and

eliminate all of them if a mistake is observed. Now, instead of majority, predict using the set with larger Ldim. Specifically,

SOA (Standard Optimal Algorithm)

Initial: $V_1 = \mathcal{H}$ (all hypotheses, possibly infinite set)

for each $t = 1, 2, \ldots$

- Observe x_t
- Construct $V_t^{(0)} = \{h \in V_t : h(x_t) = 0\}$

  $V_t^{(1)} = \{h \in V_t : h(x_t) = 1\}$
- Predict $p_t = \begin{cases} 0 & \text{if } Ldim(V_t^{(0)}) > Ldim(V_t^{(1)}) \\ 1 & \text{otherwise} \end{cases}$
- Observe y_t (the correct label)
- Update $V_{t+1} = V_t^{(y_t)}$ (i.e., the set corresponding to the correct label)

Lemma (3.3, Chap 3): $M_{\pi_{SOA}}(\mathcal{H}) \leq Ldim(\mathcal{H})$.

Proof: We claim that if at some time a mistake occurs, then $Ldim(V_{t+1}) \leq Ldim(V_t) - 1$.

Suppose not, i.e., $Ldim(V_{t+1}) = Ldim(V_t)$, but a mistake occured at time t.

$$\therefore \quad Ldim(V_{t+1}) = Ldim(V_t^{(y_t)}) = Ldim(V_t).$$

update step ↙ contrapositive assumption ↙

Further, by definition of p_t (choose the label with the larger $Ldim(\cdot)$), we have that $Ldim(V_t^{(1-y_t)}) \geq Ldim(V_t^{(y_t)})$

$$\Rightarrow \quad Ldim(V_t) = Ldim(V_t^{(0)}) = Ldim(V_t^{(1)})$$

x_t observed at t: $\{h : h(x_t) = 0\}$ $\{h : h(x_t) = 1\}$.

$\Rightarrow$ We can construct a shattered tree for V_t with depth $Ldim(V_t) + 1$.

This is a contradiction. ∎

Corrollary: (Lem 3.2 + 3.3) $M_{\pi_{SOA}}(\mathcal{H}) = Ldim(\mathcal{H})$.