CSE 522: Algorithms and Uncertainty Spring 2017

Lecture 5 — April 10, 2017

Lecturer: Nikhil R. Devanur

1 Online Classification

In Online classification, you get a sequence of examples, and the algorithm has to predict the label for each. We proceed in rounds as follows.

For $t = 1..T$ do

- See x_t.
- Predict z_t.
- See y_t.

The goal is to minimize the number of incorrect predictions.

$$\textbf{Minimize} \sum_{t=1}^{T} \mathbf{1}(z_t \neq y_t)$$

We introduce some notation:

- $\ell_t(\text{ALG}) = \mathbf{1}(z_t \neq y_t)$.
- $\ell_{1..t}(\text{ALG}) = \sum_{\tau=1}^{t} \ell_t(\text{ALG})$.
- $\ell_t(h) = \mathbf{1}(h(x_t) \neq y_t)$.
- $\ell_{1..t}(h) = \sum_{\tau=1}^{t} \ell_t(h)$.

Given a hypothesis class $\mathcal{H}$, the performance of the algorithm is measured in terms of the *regret* w.r.t. $\mathcal{H}$, which is defined as

$$\text{REGRET} := \ell_{1..T}(\text{ALG}) - \min_{h \in \mathcal{H}} \ell_{1..T}(h).$$

(We suppress the dependency on T and other parameters for the sake of simplicity of notation.) We will consider only finite $\mathcal{H}$ here, and we will denote the size by $n := |\mathcal{H}|$.

Ideally we would like to get an algorithm whose regret grows as $o(T)$, i.e.,

$$\lim_{T \to \infty} \frac{\text{REGRET}}{T} = 0.$$

1.1 Realizable case:

Assume that there is a hypothesis $h^* \in \mathcal{H}$ such that for all $t = 1..T$, we have $\ell_t(h) = 0$. Let $\mathcal{H}_t^{\checkmark}$ be the set of all hypothesis in $\mathcal{H}$ that have not made an error until time t, i.e.,

$$\mathcal{H}_t^{\checkmark} := \{h \in \mathcal{H} : \ell_{1..t-1}(h) = 0.\}.$$

We will use a majority rule to define the algorithm. Each hypothesis in $\mathcal{H}_t^{\checkmark}$ gets a vote, and the label with the maximum votes is the majority. Define

$$\textsc{majority}(\mathcal{H}, x) = \arg\max_y |\{h \in \mathcal{H} : h(x) = y\}|.$$

The "Majority Algorithm" is:

$$\text{Predict } z_t = \textsc{majority}(\mathcal{H}_t^{\checkmark}, x_t).$$

Theorem 1. *Regret of the Majority Algorithm is at most* $\log_2 n$.

Proof. Every time the algorithm makes a mistake, the size of $\mathcal{H}_t^{\checkmark}$ reduces by more than half, therefore,

$$|\mathcal{H}_{T+1}^{\checkmark}| \leq |\mathcal{H}_1^{\checkmark}|/2^{\ell_{1..T}(\textsc{alg})}.$$

The proof is completed by noting that $|\mathcal{H}_1^{\checkmark}| = n$ and $|\mathcal{H}_{T+1}^{\checkmark}| \geq 1$, since $h^* \in \mathcal{H}_{T+1}^{\checkmark}$. □

Remark 1. *Compare this to the sample complexity bound for the realizable case.*

Randomized Majority: An alternate algorithm is to pick a label with probability proportional to the number of votes it gets. Alternately, the algorithm is:

- Pick $h_t \in \mathcal{H}$ uniformly at random.
- Predict $z_t = h_t(x_t)$.

Exercise 1. *What is the regret of the Randomized Majority algorithm?*

1.2 Non-realizable case

In this case we need to suitably weigh the different hypotheses and take a weighted majority vote. Towards this, we first generalize the majority rule to include weights. Let $w(\cdot) : \mathcal{H} \to \mathbb{R}$ denote a weight function.

$$\textsc{wt-majority}(\mathcal{H}, x, w(\cdot)) = \arg\max_y \sum_{h \in \mathcal{H}: h(x) = y} w(h).$$

We will use the following (exponential) weight functions, where ϵ is some parameter in $(0, 1/2]$:

$$w_t(h) := (1 - \epsilon)^{\ell_{1..t-1}(h)}.$$

The Weighted Majority Algorithm is:

$$\text{Predict } z_t = \textsc{wt-majority}(\mathcal{H}, x_t, w_t(\cdot)).$$

Some more notation:

- $W_t := \sum_{h \in \mathcal{H}} w_t(h)$.
- $W_t^{(y)} := \sum_{h \in \mathcal{H}: h(x)=y} w_t(h)$.

Theorem 2. *For the Weighted Majority Algorithm we have that*

$$\ell_{1..T}(\text{ALG}) \leq 2 \log n + 2(1+\epsilon) \min_{h \in \mathcal{H}} \ell_{1..T}(h).$$

Proof. Every time the algorithm makes a mistake, the total weight W_t reduces by a factor of $1 - \epsilon/2$. This is because the wrong label got more than half the votes, i.e., $W_t^{(1-y_t)} \geq W_t/2$.

$$W_{t+1} = W_t^{(y_t)} + W_t^{(1-y_t)}(1-\epsilon) = W_t^{(y_t)} + W_t^{(1-y_t)} - \epsilon W_t^{(1-y_t)}$$

$$\leq W_t - \epsilon W_t/2 = W_t(1-\epsilon/2).$$

On the other hand, W_{T+1} is lower bounded as follows.

$$W_{T+1} \geq \max_{h \in \mathcal{H}} w_{T+1}(h) = (1-\epsilon)^{\min_{h \in \mathcal{H}} \ell_{1..T}(h)}.$$

The initial total weight, $W_1 = n$. Therefore we get

$$n(1-\epsilon/2)^{\ell_{1..T}(\text{ALG})} \geq (1-\epsilon)^{\min_{h \in \mathcal{H}} \ell_{1..T}(h)}.$$

Now take logs and use the following fact, from the Taylor series expansion of logs, that $x \leq -\log(1-x) \leq x(1+x)$. We get that

$$\log n + \log(1-\epsilon/2)\ell_{1..T}(\text{ALG}) \geq \log(1-\epsilon) \min_{h \in \mathcal{H}} \ell_{1..T}(h)$$

$$\log n - \log(1-\epsilon) \min_{h \in \mathcal{H}} \ell_{1..T}(h) \geq -\log(1-\epsilon/2)\ell_{1..T}(\text{ALG})$$

$$\log n + \epsilon(1+\epsilon) \min_{h \in \mathcal{H}} \ell_{1..T}(h) \geq \epsilon \ell_{1..T}(\text{ALG})/2.$$

$$\ell_{1..T}(\text{ALG}) \leq 2 \log n/\epsilon + 2(1+\epsilon) \min_{h \in \mathcal{H}} \ell_{1..T}(h).$$

□

Exercise 2. *Show that the factor of 2 in the above theorem is unavoidable for deterministic algorithms, i.e., for any deterministic algorithm, there is a sequence of inputs such that*

$$\ell_{1..T}(\text{ALG}) \geq 2 \min_{h \in \mathcal{H}} \ell_{1..T}(h).$$

This shows that no deterministic algorithm can get a sub-linear regret, i.e., a regret such that $\text{REGRET}/T \to 0$ *as* $T \to \infty$.

Randomized Weighted Majority: Analogous to the unweighted case, we define an alternate algorithm by picking a label with probability proportional to the weighted sum of votes it gets. The algorithm is:

- Pick $h_t = h \in \mathcal{H}$ with probability $w_t(h)/W_t$.
- Predict $z_t = h_t(x_t)$.

The following is an equivalent description of the algorithm.

- Predict $z_t = y$ with probability $W_t^{(y)}/W_t$.

From now on, we will use $\ell_t(\text{ALG})$ to denote the *expected* loss of the algorithm in step t. The regret is also measured w.r.t. this expected loss.

Theorem 3. *Regret of the Randomized Weighted Majority Algorithm is at most* $2\sqrt{T \log n}$.

Proof. The proof proceeds very similar to the previous theorem. Every time the algorithm makes a mistake, the total weight W_t reduces by a factor of $e^{-\epsilon \ell_t(\text{ALG})}$. This is where we save the factor of 2, because we "hedge" against either case. From the definition of the algorithm, we have

$$\ell_t(\text{ALG}) = \mathbb{P}_{z_t \neq y_t}[=] W_t^{(1-y_t)}/W_t.$$

Using this, and the fact that $e^{-x} \geq 1 - x$, we get

$$\begin{aligned} W_{t+1} = W_t^{(y_t)} + W_t^{(1-y_t)}(1-\epsilon) &= W_t^{(y_t)} + W_t^{(1-y_t)} - \epsilon W_t^{(1-y_t)} \\ &= W_t - \epsilon W_t \ell_t(\text{ALG}) = W_t(1 - \epsilon \ell_t(\text{ALG})) \\ &\leq W_t e^{-\epsilon \ell_t(\text{ALG})}. \end{aligned}$$

On the other hand, W_{T+1} is lower bounded as follows.

$$W_{T+1} \geq \max_{h \in \mathcal{H}} w_{T+1}(h) = (1-\epsilon)^{\min_{h \in \mathcal{H}} \ell_{1..T}(h)}.$$

The initial total weight, $W_1 = n$. Therefore we get

$$n e^{-\epsilon \ell_{1..T}(\text{ALG})} \geq (1-\epsilon)^{\min_{h \in \mathcal{H}} \ell_{1..T}(h)}.$$

Now take logs and use the following fact, from the Taylor series expansion of logs, that $x \leq -\log(1-x) \leq x(1+x)$. We get that

$$\log n - \epsilon \ell_{1..T}(\text{ALG}) \geq \log(1-\epsilon) \min_{h \in \mathcal{H}} \ell_{1..T}(h)$$

$$\log n + \epsilon(1+\epsilon) \min_{h \in \mathcal{H}} \ell_{1..T}(h) \geq \epsilon \ell_{1..T}(\text{ALG})$$

$$\ell_{1..T}(\text{ALG}) \leq \log n/\epsilon + (1+\epsilon) \min_{h \in \mathcal{H}} \ell_{1..T}(h).$$

To get the regret bound, we use the trivial fact that $\forall h, \ell_{1..T}(h) \leq T$.

$$\ell_{1..T}(\text{ALG}) - \min_{h \in \mathcal{H}} \ell_{1..T}(h) \leq \log n/\epsilon + \epsilon T.$$

The regret bound now follows by setting

$$\epsilon = \sqrt{\frac{\log n}{T}}.$$

□

2 Online Learning/Learning from Experts

This is a generalization of the online classification problem. Here, each hypothesis is called an "expert". There are no examples and labels; instead in every round the algorithm has to pick one of the experts. Each expert incurs a different loss in each round, and the goal is to minimize the total loss. The losses can be arbitrary real numbers. Assume for now that they are in the interval $[0, 1]$. E.g., this can be used to model probabilistic predictions, where the loss is some "penalty" function based on the predicted probability and the eventual outcome. As we will see in the next class, there are many other applications of this problem. To formally define the problem, we proceed in rounds as follows.

For $t = 1..T$ do

- Pick $h_t \in \mathcal{H}$.
- See $\ell_t(h) \in [0, 1], \forall h \in \mathcal{H}$.

The loss of the algorithm is $\ell_t(\text{ALG}) := \ell_t(h_t)$. The goal is to minimize the total loss $\ell_{1..T}(\text{ALG})$.

Randomized Weighted Majority: The algorithm from the previous section extends pretty much as is to this more general problem. The algorithm now is:

- Pick $h_t = h \in \mathcal{H}$ with probability $w_t(h)/W_t$.

Theorem 4. *Regret of the Randomized Weighted Majority Algorithm for the problem of learning from experts is at most* $2\sqrt{T \log n}$.

Proof. The proof is almost the same as the previous theorem. We will show the following, after which the proof is identical.

$$W_{t+1} \leq W_t(1 - \epsilon \ell_t(\text{ALG})).$$

From the definition of the algorithm, we have

$$\ell_t(\text{ALG}) = \sum_{h \in \mathcal{H}} \mathbb{P}_{h_t = h}[\ell_]t)h = \sum_{h \in \mathcal{H}} w_t(h)\ell_t(h)/W_t.$$

Using this, we get that

$$\begin{aligned} W_{t+1} &= \sum_{h \in \mathcal{H}} w_t(h)(1-\epsilon)^{\ell_t(h)} \\ &\leq \sum_{h \in \mathcal{H}} w_t(h)(1-\epsilon\ell_t(h)) \\ &= \sum_{h \in \mathcal{H}} w_t(h) - \epsilon \sum_{h \in \mathcal{H}} w_t(h)\ell_t(h) \\ &= W_t(1-\epsilon\ell_t(\textsc{alg})). \end{aligned}$$

□

Exercise 3. *This exercise has 2 parts.*

- *What if there are gains as well as losses? Suppose* $\ell_t(h) \in [-1, 1]$. *What is the algorithm? What is the regret bound?*
- *Bonus: generalize this further when* $\ell_t(h) \in [-a, b]$ *for some* $a, b > 0$.