CS 598: Theoretical Machine Learning

Lecturer: Pranjal Awasthi
Scribe: Abhishek Bhrushundi

Lecture # 9
Oct 6th, 2017

In the last lecture we saw a variant of the halving algorithm (in which we restart from scratch every time the halving algorithm has eliminated all the functions in the class) that makes at most $\log|H| \cdot m$ mistakes, where m is the number of mistakes made by the best function in H.

In this lecture we will see an algorithm that makes $O(m + \log|H|)$ mistakes.

1 The multiplicative weights update method: The weighted majority algorithm

The multiplicative weights update method is quite general, and has applications in several areas, including, but not limited to

- Solving linear and convex programs
- Playing games
- Learning graphical models
- Sparsification, linear systems
- Studying evolutionary algorithms

The high level idea is that the algorithm maintains a distribution of weights over the set of functions H, and iteratively updates the weights of the functions that make mistakes using a multiplicative update rule. The first variant that we will analyze in this lecture is known as the *weighted majority* algorithm.

Weighted majority algorithm

- At $t = 0$, the algorithm initializes all weights to 1, i.e. $\forall h \in H$, $w_h^{(0)} = 1$.
- At time step t, on input x_t, let

$$w_+ = \sum_{h \in H : h(x_t) = +1} w_h^{(t-1)}$$

$$w_- = \sum_{h \in H : h(x_t) = -1} w_h^{(t-1)}.$$

  If $w_+ \geq w_-$, the algorithm outputs 1, otherwise the algorithm outputs -1.

- After receiving the correct answer y_t for x_t, for every $h \in H$ that made a mistake on x_t, i.e. $h(x_t) \neq y_t$, the algorithm updates the weight of h as follows:

$$w_h^{(t)} = \frac{1}{2} \cdot w_h^{(t-1)}.$$

Theorem 1. *For any input sequence, if M is the number of mistakes made by the weighted majority algorithm, and m is the number of mistakes made by the best algorithm in H, then*

$$M \leq 2.4(m + \log |H|).$$

Proof. The analysis we will see below follows a commonly used strategy in analyzing online algorithms. The idea is to choose a suitable potential function, and study how it evolves with time.

A natural candidate for the potential function is the sum of all weights at time step t:

$$W^t = \sum_{h \in H} w_h^{(t)}.$$

Note that $W^0 = |H|$. Also, since the best function would have made at most m mistakes after t steps, it follows that

$$W^t \geq \frac{1}{2^m}. \tag{1}$$

On the other hand, notice that if the algorithm makes a mistake on x_t, and if H_{bad}^t denotes the set of functions in H that made a mistake on x_t, then it must be the case that

$$\sum_{h \in H_{bad}^t} w_h^{(t-1)} \geq \frac{W^{t-1}}{2}.$$

Since the weighted majority algorithm will bring down the weight of every function in H_{bad}^t by a half, we may conclude that

$$W^t \leq \frac{3}{4} \cdot W^{t-1}.$$

If the weighted majority algorithm makes M mistakes in t time steps, we may conclude that

$$W^t \leq \left(\frac{3}{4}\right)^M \cdot W^0 = \left(\frac{3}{4}\right)^M \cdot |H|. \tag{2}$$

Combining (1) and (2), we get the desired relationship between M and m:

$$\begin{aligned} \left(\frac{4}{3}\right)^M &\leq |H| \cdot 2^m \\ \implies M &\leq \log_{4/3}(2)(m + \log |H|) \\ \implies M &\leq 2.4(m + \log |H|). \end{aligned}$$

□

2 Randomized Multiplicative Weights (RMW) algorithm

For any given input sequence, $x_1, \ldots, x_T$, let M be the number of mistakes made by the weighted majority algorithm, and m be the number of mistakes made by the best algorithm in H. Then, we have the following guarantee from the previous section:

$$\frac{M}{T} - \frac{m}{T} \leq \frac{1.4(m + \log |H|)}{T}.$$

The quantity in the LHS above is known as the *regret* of the algorithm. It's the regret/loss of the algorithm for not having known the input sequence in advance.

Note that if m is large, say even around $T \cdot (0.5 + \epsilon)$ for a small ϵ, the above guarantee says nothing nontrivial about the upper bound on $\frac{M}{T}$ — even a trivial algorithm, that errs at every time step, has an upper bound of $\frac{M}{T} \le 1$. In other words, the weighted majority algorithm will have constant regret in the regime where m is large. Thus, we see that the weighted majority algorithm is useful only when m is small.

It turns out it is possible to design an algorithm that has the property that

$$\lim_{T\to\infty} \frac{M}{T} - \frac{m}{T} \to 0,$$

i.e. has zero regret. This algorithm is essentially a randomized variant of the weighted majority algorithm.

Randomized Multiplicative Weights (RMW) algorithm

- At $t = 0$ the algorithm initializes all weights to 1, i.e. $\forall h \in H$, $w_h^{(0)} = 1$.
- Let D^t be the distribution over H in which a function $h \in H$ has probability mass $\frac{w_h^{(t-1)}}{\sum_{h\in H} w_h^{(t-1)}}$.
  At time step t, on input x_t, the algorithm picks a function h^t randomly according to the distribution D^t, and predicts using h^t.
- After receiving the correct answer y_t for x_t, for every $h \in H$ that made a mistake on x_t, i.e. $h(x_t) \neq y_t$, the algorithm updates the weight of h as follows:

$$w_h^{(t)} = (1 - \epsilon) \cdot w_h^{(t-1)}.$$

The value that the parameter ϵ is set to depends on the number of time steps T for which the algorithm is being run. In particular, we will set

$$\epsilon = \sqrt{\frac{\log |H|}{T}}.$$

It can be shown that no deterministic algorithm can achieve zero regret. Furthermore, if one thinks about the entire setup as a two player game, i.e. the algorithm is the first player with the motive of minimizing regret, and the adversary (who makes the requests/provides the input sequence) is the second player trying to maximize the regret, the minimax theorem from game theory already implies the existence of a *mixed strategy* (i.e. a randomized strategy) that the first player (the algorithm) can follow in order to drive the regret down to zero. However, the implication is only existential and doesn't provide us with an explicit algorithm.

Theorem 2. *For any input sequence of length T, let M be the number of mistakes made by RMW, and m the number of mistakes made by the best algorithm in H, then*

$$\frac{\mathbb{E}[M]}{T} - \frac{m}{T} \le 2\sqrt{\frac{\log |H|}{T}}.$$

Note that the above theorem statement shows that the expected regret goes to zero. Using standard concentration bounds that we've encountered in previous lectures we can show that the actual regret of RMW is zero with high probability.

Proof. The proof uses the same potential function based approach as in the case of weighted majority. In fact, we will use the same potential function W^t as before.

As before, since the best algorithm makes at most m mistakes in T steps, we have that

$$W^T \geq (1-\epsilon)^m. \tag{3}$$

At time step t, on input x_t, let H^t_{bad} be the set of functions that make a mistake on x_t, and let

$$p^t = \frac{\sum_{h \in H^t_{bad}} w_h^{(t-1)}}{\sum_{h \in H} w_h^{(t-1)}}.$$

Since RMW scales down the weight of each $h \in H^t_{bad}$ while leaving other weights unchanged, we have

$$W^t = p^t W^{t-1}(1-\epsilon) + (1-p^t)W^{t-1} = W^{t-1}(1-\epsilon p^t).$$

Using the fact that $W^0 = |H|$, we conclude

$$\begin{aligned} W^T &= |H| \cdot \prod_{i=1}^{T} (1-\epsilon p^i) \\ &\leq |H| e^{-\epsilon p^1} \dots e^{-\epsilon p^T} \\ &= |H| e^{-\epsilon \sum_{i=1}^T p^i} \\ &= |H| e^{-\epsilon \mathbb{E}[M]}, \end{aligned} \tag{4}$$

where the last equality follows from the fact that $\mathbb{E}[M] = \sum_{i=1}^T p^i$.

Combining (3) and (4), we get that

$$\begin{aligned} & m \log(1-\epsilon) \leq \log |H| - \epsilon \mathbb{E}[M] \\ \implies & \epsilon \mathbb{E}[M] \leq \log |H| + m \log \frac{1}{1-\epsilon} \\ & \leq \log |H| + m(\epsilon + \epsilon^2) \text{ (Since } \log \tfrac{1}{1-x} \leq x + x^2.) \ . \end{aligned}$$

Dividing by T and rearranging the terms, we get

$$\begin{aligned} & \frac{\mathbb{E}[M]}{T} \leq \frac{\log |H|}{\epsilon T} + \frac{m}{T} + \frac{\epsilon m}{T} \\ \implies & \frac{\mathbb{E}[M]}{T} - \frac{m}{T} \leq \frac{\epsilon m}{T} + \frac{\log |H|}{\epsilon T} \\ \implies & \frac{\mathbb{E}[M]}{T} - \frac{m}{T} \leq \frac{\log |H|}{\epsilon T} + \frac{\epsilon m}{T} \\ \implies & \frac{\mathbb{E}[M]}{T} - \frac{m}{T} \leq \epsilon + \frac{\log |H|}{\epsilon T}, \end{aligned}$$

where the last inequality follows from the fact that $m \leq T$. Setting $\epsilon = \sqrt{\frac{\log |H|}{T}}$ completes the proof:

$$\frac{\mathbb{E}[M]}{T} - \frac{m}{T} \leq 2 \cdot \sqrt{\frac{\log |H|}{T}}.$$

□

References

http://www.cs.cmu.edu/ avrim/Papers/survey.pdf.