CS 395T Computational Learning Theory
Lecture 7: September 25, 2006 Lecturer: Adam Klivans
Scribe: Clinton Jones

7.1 Learning Disjunctions in the Mistake Bounded Model

We have seen that the mistake bound for learning disjunctions in the Mistake Bounded Model is $O(n)$. Now assume that the disjunction is on k¡¡n variables. Theoretically, it is possible to achieve a $O(k \log n)$ bound. When an algorithm has a mistake bound proportional to the number of attributes rather than the number of variables, it is called "Attribute Efficient".

7.1.1 An Algorithm For Learning Disjunctions

For simplicity, assume there are no negations. This can safely be allowed, and will be explained later.

Initial Hypothesis: $x_1 + x_2 + \ldots + x_n \geq n$ (Linear Threshold Function on all variables)

Take and example (e.g., 10101011), and predict according to the current linear threshold function. If a mistake is made, it will either be a false positive or a false negative. We will first look at false positives, which we will call Type 1 mistakes. In this we remove all variables that are set to 1 (since these variables are guaranteed not to be in the target function). For false negatives (Type 2 mistakes), we simply double the coefficient of every feature labeled 1.

7.1.2 Mistake Bound Analysis

Question 1: What is the bound on Type 2 mistakes?
Answer (3 parts):
1 - After each Type 2 mistake, we are doubling at least one variable that actually exists in the true disjunction.
2 - A variable can only be doubled at most $\log n$ times (After $\log n$ Type 2 mistakes on any given variable, the corresponding term in the threshold function will have weight at least n. This will prevent further Type 2 mistakes for that variable).
3 - Since there are only k relevant variables in the disjunction, in the worst case we will have to double each relevant variable independently from the rest, and therefore will have to repeat part **2** k times.

Therefore, the number of Type 2 mistakes is at most $k \log n$

Question 2: What is the bound on Type 1 mistakes?

Answer: When we make a mistake of Type 1, consider w, the sum of the weights of all the variables we remove. We know at this point that $w \geq n$, because we returned a false positive. Since every Type 2 mistake adds at most weight n, we know that the number of Type 1 mistakes cannot exceed the number of type 2 mistakes, which is $k \log n$.

So the total mistake bound is $O(k \log n)$.

Now, if we were to include negations we would increase the size of the original disjunction to 2n by adding in the negated versions of all variables. This means that we can only guarantee that there will be no more than $(k \log 2n)$ Type 2 mistakes, and therefore no more than another $(k \log 2n)$ Type 1 errors. But this does not affect the $O(k \log n)$ bound we proved.

This is known as the **Winnow-light algorithm**, introduced (along with the Mistake Bounded Model) in [1].

7.2 Best Experts

What if we have a number of "experts" for a particular problem (such as stock picking or horse racing) who are willing to give us a prediction, and we want to make the best prediction possible? We don't know which expert is the best before we see the true answer, so we should take into account all the experts' predictions when making our own. But we want the number of our own mistakes to be very close to that of the best expert. How can we do this, and what is our mistake bound?

7.2.1 Weighted Majority Algorithm

To more formally state the problem, we have k experts, so for each prediction we have a k-length string $c_i, \ldots, c_k$, with $c_i \in \{-1, 1\}$.

To combine the experts, we can use the **weighted majority algorithm**:

$w_1 c_1 + w_2 c_2 + \ldots + w_k c_k \geq 0$

Initially, all weights are set to 1. Each time expert c_i makes a mistake, its weight w_i is cut in half.

Question: If the best expert has mistake bound m, how good will this algorithm perform?

Answer: $O(m + \log k)$. Of course, we should not make claims without backing them up, so we will now explain how this is achieved.

7.2.2 Mistake Bound Analysis

First we will examine the sum of the weights, which is initially k. Every time a mistake is made, then at least $\frac{k}{2}$ experts are wrong and their weight is halved. So we lose at least $\frac{k}{4}$ of the weight, leaving at most $\frac{3k}{4}$ of the weight.

After t mistakes, the remaining weight will therefore be $k(\frac{3}{4})^t$. Now, the best expert will have weight

$(\frac{1}{2})^m$, since its mistake bound is m and its weight will be halved for each of those mistakes. Because that weight is included in the sum of the weights, we know that $(\frac{1}{2})^m \leq k(\frac{3}{4})^t$. t is the number of mistakes we make with our algorithm, and we will change the notation for this to M, so that $(\frac{1}{2})^m \leq k(\frac{3}{4})^M$. Rearranging this quantity, we get $M \leq -\frac{m+\log k}{\log \frac{3}{4}}$, which is approximately $2.4(m + \log k)$. Therefore, we have shown that this algorithm achieves a mistake bound of $O(m + \log k)$.

This analysis comes from [2].

It is also possible to obtain a bound of $M \leq (1+\epsilon)m + (\frac{1}{\epsilon})\log k$, through the use of a randomized algorithm that will not be discussed here.

References

[1] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. Machine Learning, 2:285–318, 1988.

[2] N. Littlestone and M. Warmuth. The weighted majority algorithm. Information and Computation, 108:212–261, 1994.