CS 446: Machine Learning
Discussion Session

Daniel Khashabi

October 2, 2015

1 The mistake bound for the halving algorithm is not optimal

Describe a concept class C where the worst-case mistake bound of the halving algorithm ($\log |C|$) is not tight. Explain your answer.
Solution: Here is an example: The concept class of all functions $g_x : \{0,1\}^n \to \{0,1\}$, when $g_x(\tilde{x})$ is 1, when $x = \tilde{x}$, and zero otherwise.

2 Halving algorithm is not optimal

Describe a concept class C for which the halving algorithm is not optimal, i.e., you would get a better worst-case mistake bound by not going with the majority vote among the concepts in C consistent with examples observed so far.
Solution: Think about it!

3 Lazy algorithms vs non-lazy algorithms

An online learning algorithm is lazy if it changes its state only when it makes a mistake. Show that for any deterministic online learning algorithm A achieving a mistake bound M with respect to a concept class C, there exists a lazy algorithm A' that also achieves M with respect to C.

Recall the definition: Algorithm A has a mistake bound M with respect to a learning class C if A makes at most M mistakes on any sequence that is consistent with a function in C.
Solution: Assume that there exists a sequence of examples on which A' makes $M' > M$ mistakes. Cross out all examples on which A' doesnt make a mistake, and let S denote the resulting sequence. Both A and A' behave identically on S, and A makes at most M mistakes on any sequence of examples, including S. This leads to the desired contradiction. (Obviously, if the original sequence is consistent with a concept in C, so is S.)

4 Modifications of Winnow

Here is the Winnow studied in the class:

```
• Initialize all weights $w_i = 1$.
• Loop over instances $(x_j, y_j)$
    – Given input instance $x_j$, make prediction $h(x_j)$.
    – If $h(x_j) \neq y_j$ and $y_j = 1$, then $w[i] \leftarrow 2w[i]$, for all $i$ that $x_j[i] = 1$.
    – If $h(x_j) \neq y_j$ and $y_j = -1$, then $w[i] \leftarrow w[i]/2$, for all $i$ that $x_j[i] = 1$.
```

Consider the following modifications:

```
• Initialize all weights $w_i = 1$.
• Loop over instances $(x_j, y_j)$
    – Given input instance $x_j$, make prediction $h(x_j)$.
    – If $h(x_j) \neq y_j$ and $y_j = 1$, then $w[i] \leftarrow 2 \times w[i]$, for all $i$ that $x_j[i] = 1$.
    – If $h(x_j) \neq y_j$ and $y_j = -1$, then $w[i] \leftarrow 0$, for all $i$ that $x_j[i] = 1$.
```

Prove that this algorithm has mistake bound of this algorithm is the following:

$$1 + r(1 + \lg n) = O(r \lg n)$$

Solution: To prove this, we first prove that, the number of the mistakes on positive instances is bounded. At first, each of weights are 1. Also, on positive instances, only the weights that are less than n can be updated (doubled). In other words, each weight can be updated at most $\log_2 n + 1$ times. Thus the number of the mistakes on positive instances is bounded by:

$$M_+ \leq r(1 + \log_2 n)$$

Then we bound the number of the mistakes on negative instances by a constant factor of the number of the mistakes on positive instances, which would result in the final bound. First, note that, when making mistake on negative instances, the value of the total weight decreases at least by n(why?). Also, when making mistake on a positive instance, the total weight increases by at most n. Also, initially the total weight is n. Then,

$$n + M_+ n - M_- n > 0 \Rightarrow M_- < M_+ + 1$$

This would give the over bound on mistakes:

$$M = M_- + M_+ \leq 2r(1 + \log_2 n) + 1 \in O(r(1 + \log_2 n))$$

5 Another modified Winnow

Whenever the correct label is 0 (regardless of whether the algorithm made a mistake or not), set $w[i] = 0$ for each x_i equal to 1.

Prove that this algorithm has mistake bound of this algorithm is the following:

$$O(r \lg n)$$

Solution: By combining question 3 and 4.