Computational Learning Theory: Mistake Bounds

Recommended reading:

Mitchell: Chapter 7.5

Machine Learning 10-701

Tom M. Mitchell
Carnegie Mellon University

Mistake Bounds

So far: how many examples needed to learn?

What about: how many mistakes before convergence?

Let's consider similar setting to PAC learning:

- Instances drawn at random from X according to distribution $\mathcal{D}$
- Learner must classify each instance before receiving correct classification from teacher
- Can we bound the number of mistakes learner makes before converging?

(assume target concept is in H, and noise-free training data)

Mistake Bounds: Find-S

Consider Find-S when H = conjunction of boolean literals

Find-S:

- Initialize h to the most specific hypothesis $l_1 \wedge \neg l_1 \wedge l_2 \wedge \neg l_2 \ldots l_n \wedge \neg l_n$
- For each positive training instance x
  - Remove from h any literal that is not satisfied by x
- Output hypothesis h.

How many mistakes before converging to correct h?

(assume target concept is in H, noise-free training data)

Mistake Bounds: Halving Algorithm

1. Initialize VS to all hypotheses in H
2. For each new training example,
   - remove from VS all hyps. that misclassify this example

Consider the Halving Algorithm:

- Learn concept using version space CANDIDATE-ELIMINATION algorithm
- Classify new instances by majority vote of version space members

How many mistakes before converging to correct h?

- ... in worst case?
- ... in best case?

Optimal Mistake Bounds

Let $M_A(C)$ be the max number of mistakes made by algorithm A to learn concepts in C. (maximum over all possible $c \in C$, and all possible training sequences)

$$M_A(C) \equiv \max_{c \in C} M_A(c)$$

Definition: Let C be an arbitrary non-empty concept class. The **optimal mistake bound** for C, denoted $Opt(C)$, is the minimum over all possible learning algorithms A of $M_A(C)$.

$$Opt(C) \equiv \min_{A \in learning\ algorithms} M_A(C)$$

$$VC(C) \leq Opt(C) \leq M_{Halving}(C) \leq log_2(|C|).$$

Weighted Majority Algorithm

a_i denotes the i^{th} prediction algorithm in the pool A of algorithms. w_i denotes the weight associated with a_i.

– For all i initialize $w_i \leftarrow 1$

– For each training example $\langle x, c(x) \rangle$

  * Initialize q_0 and q_1 to 0
  * For each prediction algorithm a_i
    · If $a_i(x) = 0$ then $q_0 \leftarrow q_0 + w_i$
    If $a_i(x) = 1$ then $q_1 \leftarrow q_1 + w_i$
  * If $q_1 > q_0$ then predict $c(x) = 1$
    If $q_0 > q_1$ then predict $c(x) = 0$
    If $q_1 = q_0$ then predict 0 or 1 at random for $c(x)$
  * For each prediction algorithm a_i in A do
    If $a_i(x) \neq c(x)$ then $w_i \leftarrow \beta w_i$

Weighted vote:

when β=0, equivalent to the Halving algorithm…

Weighted Majority

Even algorithms that learn or change over time…

[Relative mistake bound for WEIGHTED-MAJORITY] Let D be any sequence of training examples, let A be any set of n prediction algorithms, and let k be the minimum number of mistakes made by any algorithm in A for the training sequence D. Then the number of mistakes over D made by the WEIGHTED-MAJORITY algorithm using $\beta = \frac{1}{2}$ is at most

$$2.4(k + \log_2 n)$$

Proof: relative mistake bound for Wtd Majority

- Let
  - D be any sequence of training examples,
  - A be any set of n prediction algorithms,
  - $\beta = 0.5$
  - Let $a_j \in A$ be the prediction algorithm that makes fewest mistakes over D
  - Let k be the number of mistakes made by a_j over D
  - Let M be the number of mistakes made during training by WtdMajority
  - Let W = $\sum_{i=1}^{n} w_i$ be the sum of weights for all n algorithms (initially W=n)

- After training, the final weight w_j of a_j will be ...
- After training, total weight W of entire collection will be …

Proof: relative mistake bound for Wtd Majority

- Let
  - D be any sequence of training examples,
  - A be any set of n prediction algorithms,
  - $\beta = 0.5$
  - Let $a_j \in A$ be the prediction algorithm that makes fewest mistakes over D
  - Let k be the number of mistakes made by a_j over D
  - Let M be the number of mistakes made during training by WtdMajority
  - Let W = $\sum_{i=1}^{n} w_i$ be the sum of weights for all n algorithms (initially W=n)

- After training, the final weight w_j of a_j will be $\beta^k = (1/2)^k$
- After training, total weight W of entire collection will be at most $n(3/4)^M$
  - Note each mistake reduces current W to at most (3/4)W
- But w_j must be less than or equal to W

$$\left(\frac{1}{2}\right)^k \leq n\left(\frac{3}{4}\right)^M \qquad\qquad M \leq 2.4(k + \log_2 n)$$