Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm

2. The setting:

1. We assume that learning takes place in a seq. of trials.
2. The order of event in a trial is as follows:
   1. The learner receives an instance $\in \{0,1\}^n$. $\{0,1\}^n$ is referred to as the instance space.
   2. The learner makes a prediction of the correct value $\in \{0,1\}$.
   3. The learner is told whether or not the prediction was accurate, referred to as the reinforcement.
3. We assume that $\exists f: \{0,1\}^n \to \{0,1\}$ which maps each instance to each correct response.
4. This f is called the target function or target concept.
5. An algo. for learning in this setting is called "on-line learning from examples".
6. When they mention learning algo. without further qualification, they mean ↑
7. They only consider deterministic algo.
8. The class of possible target function is called the "target class", referred to a C.

3. The nature of absolute mistake bounds:

1. $\forall$ learning algo. A and $\forall f$, let $M_A(f)$ be the max. over all possible seq. of instances of the no. of mistakes that algo. A makes when the target fun. is f.
2. $\forall A$ and non-empty C, let $M_A(f) = \max_{f \in C} M_A(f)$.
3. $M_A(C) = -1$ if C empty.
4. Any no. greater than or equal to $M_A(C)$ will be called a mistake bound for algo. A applied to a class C.
5. $opt(C) = \min_A M_A(C)$.
6. A is optimal for C if $M_A(C) = opt(C)$

} Def. 1. in paper.

7. $opt(C)$ is the best possible worst case mistake bound for any algo. learning C
8. If compt. resource is not an issue, we can use "halving algo".
9. Given a target class C and an instance x, let $\xi_0(C,x) = \{f \in C \mid f(x) = 0\}$, $\xi_1(C,x) = \{f \in C \mid f(x) = 1\}$.
10. Halving algo. pseudo code:
    1. Set CONSIST to C
    2. For each new instance x:
    3. If $|\xi_1(\text{CONSIST}, x)| > |\xi_0(\text{CONSIST}, x)|$, predicts 1, o.w. predict 0.
    4. Receive reinforcement $\in \{0,1\}$
    5. Set CONSIST $= \xi_1(\text{CONSIST}, x)$ if reinforcement is 1, $\xi_0(\text{CONSIST}, x)$ o.w.

    CONSIST now contains fun. consistent to all past instances.

} algo. 1

11. Let $M_{HALVING}(C)$ be the max. no. of mistakes that the algo makes when: it is run for the target class C
    - . CONSIST is initially set to C
    - . the target fun. $\in$ C.

12. For any non-empty target class C, $M_{HALVING}(C) \leq \log_2 |C|$. ← theorem 1.

13. For any finite target class C, $opt(C) \leq \log_2 |C|$ ← theorem 2.

14. A mistake tree for a target class C over an instance space X is:
    - . a binary tree
    - . each node is a non-empty subset of C
    - . each internal node is labeled with a point in X
    - . the root of the tree is C
    - . given any internal node C' labeled with x, ~~the left chi~~:
        - . the left child of C', if present, is $\xi_0(C', x)$.
        - . the right " " " $\xi_1(C', x)$.

    } def. 2.

15. A complete k-mistake tree is a mistake tree that is a complete binary tree with height k.

16. The height of a tree is the length in edges of the longest path from the root.

17. For any non-empty finite target class C, let K(C) equals the largest int. | $\exists$ a complete k-mistake tree for C.

18. Standard optimal algo pseudo-code:
    1. Set CONSIST to C
    2. For each new instance x:
    3. If $K(\xi_1(\text{CONSIST}, x)) > K(\xi_0(\text{CONSIST}, x))$, predicts 1, ow 0.
    4. Set CONSIST = $\xi_1(\text{CONSIST}, x)$ if reinforcement is 1, o.w. to $\xi_0(\text{CONSIST}, x)$

    one trial {3., 4.}

    } algo 2

19. Whenever a mistake occurs, the remaining consistent fun s have the smaller maximal complete mistake tree.

20. Let X be any instance space.
    SOA denote the standard optimal algo.
    C be any finite class of funs with domain X and range {0, 1}.
    $$opt(C) = M_{SOA}(C) = K(C)$$

    } therem 3.

21. This theorem proof requires 2 lemma:

22. For any target class C, $opt(C) \geq K(C)$ ← lemma 1.

23. Let C be a finite non-empty target class, containing the target fun.
    - . SOA is used as the learning algo.
    - . The seq. of instances is $\{x_1, \dots, x_t$
    - . CONSIST_i : the value of CONSIST at the start of trial i.
    - . $K(\text{CONSIST}_i) = k$

    for any $k \geq 0$ and $i \in \{1, \dots, t\}$, SOA will make at most k mistakes during trials $i, \dots, t$

    } lemma 2.

24. A set $S \subseteq X$ is shattered by a target class C if for every $U \subseteq S$, $\exists f \in C \mid f$ is 1 on U and 0 on $S - U$. } def 3 (3)

25. The VCdim(C) is the cardinality of the largest set that is shattered by C. ← def 4

26. For any target class C, VCdim(C) $\leq$ opt(C) ← theorem 4.

General transformations

1. A equivalence (equi.) query is a request by an algorithm that asks if the tgt fun. matches some fun. described in the query.
2. Whenever an alg. receives a neg. ans. to an equi. query, it also receives a counter-example, a pt at which the tgt fun. & proposed fun. ~~disg~~ disagree.
3. The equi. query algo. that they consider receive no examples as input other than the counter-example in the queries.
4. In this section, they use "query algo" to refer to an algo that learns using equi. queries, and "online learning alg", "mistake-bounded alg.", "alg. for learning from examples" to refer to the alg. of the type discussed before.
   to refer to the alg. of the type of online learning from examples.
5. Def: current hypothesis for an alg. for online learning
   . the curr. hypo. is defined initially & b/t trials.
   . the " " is a fun from the instance space to {0,1}
   . its value at any instance x is defined to be the response that the algo. would give at the next trial if the instance received in the next trial is x.
6. The state of an algo. at the conclusion of a trial is a representation of the current hypo. of the algo.
7. Using this rep. to represent the fun appearing in queries, an algo. that learns from examples can be transformed to a query algo.
8. Transform mistake-bounded algo → query algo ← algo. transformation 1.
   Given: a mistake-bounded algo A. → query algo. B
   . the first query of B is the initial hypo of A.
   . for each query of B: . if the resp. indicates that the query specifies the correct tgt. fun., B halts and reports the correct tgt fun.
   . if not, B tell A its prediction was wrong.
   B then takes the new hypo. of A as the next query.
9. The no. of queries needed by the derived alg. B to exactly identify the tgt. fun. f is bounded by $M_A(f)+1$.

10. Similarly, we can transform a query algo to a mistake-bounded algo.

$A \vee B$: A true or B true or both true.

The linear-threshold algorithm

1. The first describe the class of target function.
2. They will consider linearly-separable Boolean functions, which are those functions that can be computed by a one-layer linear-threshold network.
3. A function from $\{0,1\}^n$ to $\{0,1\}$ is said to be linearly separable if $\exists$ a hyperplane in R^n ~~separ~~ separating the points on which the function is 1 from the points on which it is 0.
4. Monotone disjunctions constitute one class of linearly-separable functions.
5. A monotone disjunction is a disjunction in which no literal appears negated, that is, a function of the form: $f(x_1, \ldots, x_n) = x_{i_1} \vee \ldots \vee x_{i_k}$. } def 5
6. The hyperplane given by $x_{i_1} + \ldots + x_{i_k} = \frac{1}{2}$ is a separating hyperplane for $f(x_1, \ldots, x_n) = x_{i_1} \vee \ldots \vee x_{i_k}$.
7. The first variant of their algorithm is specialized for learning monotone disjunctions.
8. WINNOW1 ← algorithm 3 :
   - The instance space is $X = \{0,1\}^n$
   - The algo. maintains non-negative weights $\{w_i\}_{i=1}^n$, each having an initial value of 1.
   - The algo. also uses a no. referred to as the threshold, θ.
   - When the learner receives an instance $(x_1, \ldots, x_n)$, the learner responds as follows:
     - if $\sum_{i=1}^n w_i x_i > \theta$, then it predicts 1.
     - if $\sum_{i=1}^n w_i x_i \le \theta$, then it predicts 0.
   - Update :

learner's prediction	correct response	update action	update name
1	0	$w_i := 0$ if $x_i = 1$ w_i unchanged if $x_i = 0$	elimination step
0	1	$w_i := \alpha \cdot w_i$ if $x_i = 1$ w_i unchanged if $x_i = 0$	promotion step

9. Suppose that the tgt. fun. is a k-literal monotone disjunction given by $f(x_1, \ldots, x_n) = x_{i_1} \vee \ldots \vee x_{i_k}$. If WINNOW1 is run with $\alpha > 1$ and $\theta \ge \frac{1}{\alpha}$, then for any sequence of instances the total no. of mistakes will be bounded by $\alpha k (\log_\alpha \theta + 1) + \frac{n}{\theta}$. ← theorem 7.
10. 3 lemmas used in the proof:
    - let u be the no. of promotion steps that have occured by the end of some seq. of trials.
    - let v be the no. of elimination steps " " " " ~~same~~ " same " " "
    1. $v \le \frac{n}{\theta} + (\alpha - 1) u$ ← lemma 3
    2. $\forall i,\ w_i \le \alpha \theta$
    3. After u promotion steps & an arbitrary no. of elimination steps, $\exists$ some i for which $\log_\alpha w_i \ge \frac{u}{k}$.

11. Let $\tilde{C}_k$: class of k-literal monotone disjunctions ~~function~~.

C_k: class of all monotone disjunctions that have at most k-literals.

12. lower bound on the no. of mistakes needed to learn $\tilde{C}_k$ & C_k.

. For $1 \leq k \leq n$, $opt(C_k) \geq opt(\tilde{C}_k) \geq k \lfloor \log_2 \frac{n}{k} \rfloor$.

. For $n > 1$, $opt(C_k) \geq \frac{k}{8}\left(1 + \log_2 \frac{n}{k}\right)$

theorem 8

13. In the space of d variables, the k-DNF functions have the following structure:

$T_1 \vee T_2 \vee \ldots \vee T_r$

where r is the no. of disjunctive terms and each term T_x represents the conjunction (and) of k out of d variables $(x_1, x_2, \ldots, x_d)$.

14. k-literal monotone conj. are 1-term monotone k-DNF.

. ℓ-literal monotone disj. are ℓ-term monotone 1-DNF.

15. For $1 \leq k \leq n$ and $1 \leq \ell \leq \binom{n}{k}$,

let: . C be the class of funs expressible as ℓ-term monotone k-DNF formulas.

. m be any integer, $k \leq m \leq n \mid \binom{m}{k} \geq \ell$.

$\Rightarrow$ $VCdim(C) \geq k\ell \lfloor \log_2 \frac{n}{m} \rfloor$

16. The algo. (WINNOW 1?) can be modified to work on larger class of Boolean functions.

For any instance space $X \subseteq \{0,1\}^n$,

for any $\delta \mid 0 < \delta \leq 1$.

Let F be the class of fun. from X to $\{0,1\}$ with the following property:

for each $f \in F(X, \delta)$, $\exists\, \mu_1, \ldots, \mu_n \geq 0 \mid \forall (x_1, \ldots, x_n) \in X$:

$$\sum_{i=1}^{n} \mu_i x_i \geq 1 \quad \text{if } f(x_1, \ldots, x_n) = 1 \quad \text{and} \quad (1)$$

$$\sum_{i=1}^{n} \mu_i x_i \leq 1 - \delta \quad \text{if } f(x_1, \ldots, x_n) = 0 \quad (2)$$

In words: the inverse images of 0 and 1 are linearly separable with a minimum separation that depends on δ.

17. WINNOW 2:

learner's prediction	correct response	update action	update name
1	0	$w_i := \frac{w_i}{\alpha}$ if $x_i = 1$ $w_i := w_i$ if $x_i = 0$	demotion step
0	1	$w_i := \alpha \cdot w_i$ if $x_i = 1$ $w_i := w_i$ if $x_i = 0$	promotion step

18. They will use $\alpha = 1 + \frac{\delta}{2}$ for learning a tgt. fun. in $F(X, \delta)$

19. For $0 < \delta \leq 1$, if:

- the tgt fun. $\in F(X, \delta)$
- $\mu_1, \dots, \mu_n$ satisfies (1) & (2).
- $\alpha = 1 + \frac{\delta}{2}$
- $\theta \geq 1$
- the algo. receives instance from $X \subseteq \{0,1\}^n$,

then the no. of mistake will be bounded by:

$$\frac{8}{\delta^2} \frac{n}{\theta} + \left(\frac{5}{\delta} + \frac{14 \ln \theta}{\delta^2} \right) \sum_{i=1}^{n} \mu_i .$$

theorem 9

20. 3 lemmas used to prove theorem 9:

- $v \leq \frac{\alpha}{\alpha - 1} \cdot \frac{n}{\theta} + \alpha u$ lemma 7
- $\forall i, \; w_i \leq \alpha \theta$ lemma 8
- After u promotion step & v elimination steps, $\exists i \mid$
  $$\log w_i \geq \frac{u - (1-\delta) v}{\sum_{i=1}^{n} \mu_i} \log \alpha .$$