Valiant's theory of the learnable

Keith Briggs

Keith.Briggs@bt.com

www.btexact.com/people/briggsk2

BTexact

2003 April 14 1330

typeset 11th April 2003 14:05 in pdfLaTeX on a linux system

Introduction

L G Valiant *A theory of the learnable*, Comm. ACM **27**, 1134-42 (1984)

We want to learn an unknown Boolean function (predicate) F

- to do this in general would take exponential time - must test n-ary F for 2^n input combinations
- so we restrict F, and then can achieve polynomial time

Learning protocol

t Boolean variables $p_1, p_2, \ldots, p_t$

vector: $\{0, 1, *\}^t$ ($*$=undetermined). *Total* means no $*$ in vector

We have available EXAMPLE()

- gives us a positive exemplification of F
- that is, an assignment of variables making F true
- for example, $F(p) = p_1 p_2 + p_3$
  - ▷ *EXAMPLE*$() \to (*, *, 1)$
  - ▷ *EXAMPLE*$() \to (1, 1, 0)$

and ORACLE(x)

- tells us if F is true for some given assignment x of variables
- for example:
  - ▷ *ORACLE*$(1, 0, 0) \to 0$
  - ▷ *ORACLE*$(0, 0, 1) \to 1$

Let D be a probability distribution on the set of vectors v such that $F(v) = 1$

Learnability

A predicate is *learnable* if $\exists$ an algorithm such that:

- it runs in polynomial time in t and in a parameter h
- with probability $1-1/h$, the deduced predicate g never outputs 1 when it should not, but outputs 1 almost always when it should

$L(h,s)$ is defined (for $\mathbb{R} \ni h > 0, s \in \mathbb{Z}^+$) as the smallest integer such that in L independent Bernoulli trials each with probability $1/h$ of success, the probability of having fewer than s successes is less than $1/h$

- For $s \geqslant 1$ and $h > 1$, $L(h,s) \leqslant 2h(s+\log h)$
-

h	s	$L(h,s)$	bound
10	2	38	86
10	5	78	146
10	10	140	246
100	2	662	1321
100	5	1157	1921
100	10	1874	2921

Finite CNF expressions

A conjunctive normal form (CNF) is a product of sums

- that is, an *and* of *ors*
- Valiant requires each clause c_i in a CNF to be a sum of literals, where a literal is either a variable p_j or a negation of a variable
- For example, $p_2+\overline{p_3}+p_6$ is a clause
- In a k-CNF, each clause contains at most k literals

Theorem A: for each $k > 0$, any k-CNF is learnable via an algorithm that uses $L(h, (2t)^{k+1})$ calls of EXAMPLE and no call of ORACLE

Algorithm A

$g =$ product of all possible k-clauses

For $n = 1, 2, \ldots, L$

- $v =$ EXAMPLE()
- for each c_i in g
    - *if* $v \not\Rightarrow c_i$*, then delete* c_i *from* g

DNF expressions

A disjunctive normal form (DNF) is a sum of products

- that is, an *or* of *and*s
- Valiant requires the DNF to be *monotone*, that is, no variable is notted
- For example, $p_1p_3p_4+p_2+p_3p_6$ is in DNF

Theorem B: any monotone DNF of degree d is learnable via an algorithm that uses $L(h,d)$ calls of EXAMPLE and dt calls of ORACLE, where t is the number of variables

Algorithm B

$g = 0$

For $n = 1, 2, \ldots, L$

- $v =$ EXAMPLE()
- if $v \not\Rightarrow g$, then for $i = 1, 2, \ldots, t$
  - ▷ *if p_i is determined in v (i.e. is not $*$), then*
    - ◊ *set v equal to $\overline{v}$ but with $p_i = *$*
    - ◊ *if ORACLE$(\overline{v}) = 1$ then $v = \overline{v}$*
  - ▷ *$m =$ product of all literals q such that $v \Rightarrow q$*
  - ▷ *$g += m$*