On learning monotone Boolean functions

FOCS '98

Avrim Blum

Carl Burch

John Langford

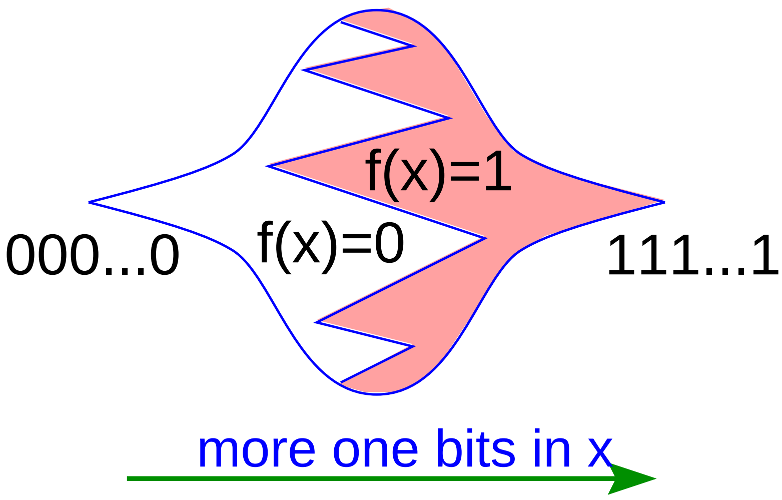
Carnegie Mellon University

Monotone functions

A function $f : \{0,1\}^n \to \{0,1\}$ is *monotone* if

$$f(\cdots 0 \cdots) = 1$$
$$\Downarrow$$
$$f(\cdots 1 \cdots) = 1$$

How to picture monotone function:



Learning

Alg. *learns monotone functions with error* α if, given $poly(n)$ samples of any hidden monotone f, alg. outputs hypothesis h so that

$$\mathbf{Pr}_x[f(x) \neq h(x)] \leq \alpha$$

(All probabilities are over uniform dist.!)

Error $\alpha = 1/2$ is easy; how to do better?

Two possible oracles to sample f:

- $RANDOM$: return unif. random x with $f(x)$
- $MEMBER$: alg. specifies x, return $f(x)$

Summary of results

How well can we learn monotone functions?

	algorithm	hardness result
[Kearns, Li, Valiant]	$\frac{1}{2} - \Omega\left(\frac{1}{n}\right)$	$O\left(\frac{1}{\sqrt{n}}\right)$
[Bshouty, Tamon]	$\frac{1}{2} - \Omega\left(\frac{\log^2 n}{n}\right)$	
us	$\frac{1}{2} - \Omega\left(\frac{1}{\sqrt{n}}\right)$	$\frac{1}{2} - \omega\left(\frac{\log n}{\sqrt{n}}\right)$

(Algorithms use only $RANDOM$ oracle;
hardness results apply even with $MEMBER$.)

Outline

- Definitions

$\rightarrow$ • Algorithm

- Hardness result

- Open question

Algorithm

1. Get $O(n)$ samples from $RANDOM$

2. If $\geq \frac{1}{2} + \frac{0.05}{\sqrt{n}}$ fraction are labeled 1, output $h \equiv 1$

3. If $\leq \frac{1}{2} - \frac{0.05}{\sqrt{n}}$ fraction are labeled 1, output $h \equiv 0$

4. Otherwise, output majority function

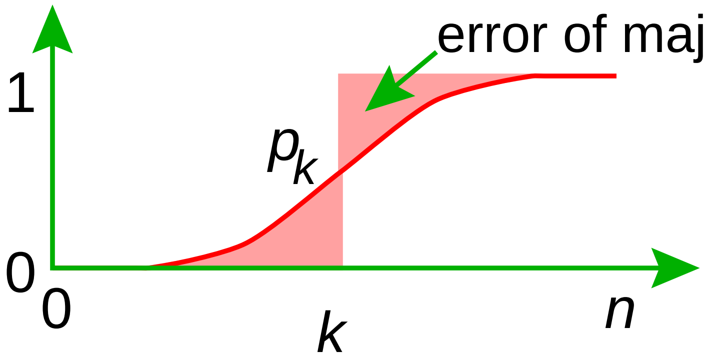
$$h(x) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} x_i > n/2 \\ 0 & \text{otherwise} \end{cases}$$

Proof outline

Key fact *If* $\mathbf{Pr}_x[f(x)=1]=1/2$ *for monotone* f*, majority function has error at most* $\frac{1}{2}-\frac{0.1}{\sqrt{n}}$

Given f, define p_k as

$$p_k := \mathbf{Pr}_{x \text{ has } k \text{ 1 bits}}[f(x)=1]$$



$$\mathbf{Err}(\text{maj}) = \sum_{k=0}^{\lfloor n/2 \rfloor} \frac{\binom{n}{k}}{2^n} p_k + \sum_{k=\lceil n/2 \rceil}^{n} \frac{\binom{n}{k}}{2^n}(1-p_k)$$

Obviously $p_k \le p_{k+1}$.

In fact, can show $p_{n/2-c\sqrt{n}} \le p_{n/2+c\sqrt{n}} - \frac{c'}{\sqrt{n}}$ using Kruskal-Katona Theorem.
Fact follows naturally.

Hardness result

Theorem *Cannot achieve error* $\frac{1}{2} - \omega\left(\frac{\log^{3/2} n}{\sqrt{n}}\right)$.

Define s as number of samples alg. uses.

Choose target function from distribution on disjunctions of terms of $t = \lg(3sn)$ variables: Include each t-term indep. with prob. p, where

$$(1-p)^{\binom{n/2}{t}} = \frac{1}{2}$$

Think of target selection this way:

> When query on x is received, go through uncommitted t-terms satisfied by x flipping bias-p coin until getting heads ($f(x) = 1$) or running out of terms ($f(x) = 0$).

After s queries, w.h.p. we commit to $\leq 2s/p$ terms.

Hardness result, cont'd.

Choose random example x uniformly.
With prob. $\geq 1 - \dfrac{3}{\sqrt{n}}$ the following occur:

- No term satisfied by x committed to being *in* f. (There are at most s such terms, each satisfied with prob. $2^{-t} = \dfrac{1}{3sn}$.)
- At most $\dfrac{2}{3p\sqrt{n}}$ terms satisfied by x committed to being *outside* f. (Each of $\dfrac{2s}{p}$ terms satisfied with prob. $\dfrac{1}{3sn}$; expect $\dfrac{2}{3pn}$.)
- x has $\dfrac{n}{2} \pm O(\sqrt{n \log n})$ one bits.

Assume alg. predicts correctly at x where these don't happen.
When all occur, we want to bound $\mathbf{Pr}_f\,[f(x) = 0]$ above and below.

Hardness result, cont'd.

$$\begin{aligned}
\mathbf{Pr}_f\left[f(x)=0\right] &\leq (1-p)^{\binom{n/2-O(\sqrt{n\log n})}{t}-\frac{2}{3p\sqrt{n}}} \\
&\leq (1-p)^{\binom{n/2-O(\sqrt{n\log n})}{t}}\left(1+\frac{2}{3\sqrt{n}}\right) \\
\cdots &\leq \frac{1}{2}+O\left(\frac{\log^{3/2} n}{\sqrt{n}}\right)
\end{aligned}$$

$$\begin{aligned}
\mathbf{Pr}_f\left[f(x)=0\right] &\geq (1-p)^{\binom{n/2+O(\sqrt{n\log n})}{t}} \\
\cdots &\geq \frac{1}{2}-O\left(\frac{\log^{3/2} n}{\sqrt{n}}\right)
\end{aligned}$$

Whether alg. predicts that $f(x)$ is 0 or 1, it has prob. $\geq \frac{1}{2}-O\left(\frac{\log^{3/2} n}{\sqrt{n}}\right)$ of erring.

Open question

Learning monotone functions in uniform dist. closed to an $O(\log n)$ factor.

But what if the target function is a monotone DNF of $poly\,(n)$ terms?

$$\underbrace{(x_1 \vee x_2 \vee x_8) \wedge (x_2 \vee x_8 \vee x_9) \wedge \cdots \wedge (x_5 \vee x_3)}_{poly\,(n)\text{ terms}}$$

- Given $MEMBER$, can achieve any error ε [Angluin], [Jackson].
- Given $RANDOM$, our $\frac{1}{2} - \Omega\left(\frac{1}{\sqrt{n}}\right)$ guarantee is best known.
- No hardness result known with $RANDOM$.