Amplification of Weak Learning Under the Uniform Distribution

Dan Boneh* Richard J. Lipton
dabo@cs.princeton.edu rjl@cs.princeton.edu

Department of Computer Science
Princeton University
Princeton, NJ 08544

Endre Szemerédi
szemered@cs.rutgers.edu
Rutgers University

Abstract

We give a new simple proof of the XOR Theorem from [Yao82]. In addition, we show an application of the Theorem to learning theory. Let $\mathcal{F}$ be a class of boolean functions, such as AC^0 or NC^1. We show that if $\mathcal{F}$ satisfies certain closure properties, then a weak learning algorithm for $\mathcal{F}$ over the uniform distribution can be amplified to a strong learning algorithm.

1 Introduction

The XOR Theorem of [Yao82] has many applications in cryptography. Proofs of the XOR Theorem can be found in [Kra86, pp. 180-199] or [Lev87]. In this paper we give a simple proof of the Theorem and show an application of it in learning theory. The results presented here are an extension of the results in [BL93]. Before we state the Theorem we require the following definition.

Definition 1 *Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be a boolean function. We define the* m-th duplication *of f to be the function $f_m : \{0,1\}^{nm} \rightarrow \{0,1\}$ defined by*

$$f_m(x_{1,1},\ldots,x_{1,n},\ldots,x_{m,1},\ldots,x_{m,n}) = \bigoplus_{i=1}^{m} f(x_{i,1},\ldots,x_{i,n})$$

The XOR Theorem of [Yao82] states the following fact:

Theorem 2 *Let $p(n) = n^d$ for some $d > 0$, and $f : \{0,1\}^n \rightarrow \{0,1\}$ a function. Let $C : \{0,1\}^{np(n)} \rightarrow \{0,1\}$ be a poly-size circuit, such that*

$$\Pr[f_{p(n)}(x) = C(x)] > \frac{1}{2} + \frac{1}{n^c}$$

Where $f_{p(n)}$ is the $p(n)$th-duplication of f and $c < 2d$. Then there exists a poly-size circuit $D : \{0,1\}^n \rightarrow \{0,1\}$ such that

$$\Pr[f(x) = D(x)] > 1 - \frac{1}{n^{\frac{d}{2}-c}}$$

Our proof is constructive and gives an efficient algorithm for constructing the circuit D.

To present the application of the XOR Theorem to learning theory we first give the following two definitions. Let $\mathcal{F}$ be a class of boolean functions and let $\delta > 0$. A *strong learning* algorithm for $\mathcal{F}$ over the uniform distribution queries the function $f : \{0,1\}^n \rightarrow \{0,1\}$ and outputs, with probability at least $1 - \delta$, a circuit C such that $\Pr[f(x) \neq C(x)] < \frac{1}{n^{O(1)}}$ where Pr is the uniform distribution. The algorithm is said to be polynomial if it is polynomial in $n, \ln\frac{1}{\delta}$.

A less stringent definition of learnability was given in [KV89]. A *weak learning* algorithm for $\mathcal{F}$ over the uniform distribution must generate, with probability at least $1 - \delta$, a circuit C such that $\Pr[f(x) = C(x)] > \frac{1}{2} + \frac{1}{n^{O(1)}}$.

We study the following problem. Given a class of functions $\mathcal{F}$ which is known to be weakly learnable in polynomial time, is it also strongly learnable in polynomial time? Of specific interest are the following classes of functions: DNF formulae with polynomial number of terms, AC^0, NC^1 and NC. We denote the class of DNF formulae with polynomial number of terms by Poly–DNF.

The XOR Theorem can be used to show that in fact weak and strong learning are equivalent over the uniform distribution, provided the concept class satisfies certain closure properties. This result provides an effective tool for the construction of efficient strong learning algorithms over the uniform distribution. To show that

---

[*] bbb

a class of functions is learnable over the uniform distribution it is sufficient to construct a weak learning algorithm for it.

It has been shown [Fre90, Fre92, Sch90] that for the PAC learning model [Val84] any class of boolean functions which is polynomially PAC learnable in the weak sense of [KV89] is also polynomially PAC learnable in the strong sense of [Val84]. The proof takes advantage of the fact that a PAC learning algorithm can learn over any probability distribution on the input space. The idea is to apply the given learning algorithm to various probability distributions. This provides enough information about the concept to obtain a strong learning algorithm. These techniques can not be applied to the uniform distribution case, since the weak learning algorithm will only produce good hypothesis when measured over the uniform distribution.

In section 2 we present the main lemma used for proving the XOR Theorem. The proof of the XOR Theorem is given in section 3. The application to learning is described in section 4.

2 The Main Lemma

Our proof of the XOR Theorem relies on a combinatorial fact which we prove in this section.

Let K be a set of elements. Typically we set $K = \{0,1\}^n$ for some n. Let $G \subseteq K^m$ and let X be a random variable uniformly distributed on G. For $1 \le i \le m$, define the random variables X_i as

$$\Pr[X_i = t] = \Pr[\ i\text{'th component of } X \text{ is } t]$$

For all $t \in K$. The variable X_i is the projection of X in dimension i.

Lemma 3 *If $|G| \ge \frac{1}{100}|K^m|$ then there exists i_0 such that for any $\beta > 0$*

$$\#\left\{t \ \ s.t. \ \ \left|\Pr[X_{i_0} = t] - \frac{1}{|K|}\right| < \frac{10}{\beta|K|}\right\} > (1 - \frac{\beta}{\sqrt{m}})|K|$$

Informally the lemma says that X_{i_0} is close to being uniformly distributed on K except for a small fraction of at most $O(\frac{1}{\sqrt{m}})$ of the elements.

Proof Define $\delta = |G|/|K|$. The entropy of X is

$$H(X) = \sum_{x \in K^m} -\Pr[X = x] \log \Pr[X = x] = \log|G| = m\log|K| - \log\frac{1}{\delta}$$

All log are taken base 2. By sub-additivity of entropy we get

$$\sum_{i=1}^{m} H(X_i) \ge H(X) = m\log|K| - \log\frac{1}{\delta}$$

Hence there must exist an i_0 such that

$$H(X_{i_0}) \ge \log|K| - \frac{1}{m}\log\frac{1}{\delta}$$

From an inequality due to Csiszar [CK81, p. 58] it follows that if a random variable Y over a probability space Ω satisfies $H(Y) \ge \log|\Omega| - C$ then

$$\sum_{y \in \Omega}\left|\Pr[Y = y] - \frac{1}{|\Omega|}\right| \le 2\ln 2\sqrt{C}$$

Hence, in our case we get:

$$\sum_{t \in K}\left|\Pr[X_{i_0} = t] - \frac{1}{|K|}\right| \le 2\ln 2\sqrt{\frac{-\log\delta}{m}}$$

Let $\beta > 0$. At most $\frac{\beta}{\sqrt{m}}|K|$ elements can satisfy

$$\left|\Pr[X_{i_0} = t] - \frac{1}{|K|}\right| \ge \frac{2\ln 2\sqrt{-\log\delta}}{\beta|K|}$$

for otherwise, summing over these elements alone would violate the inequality. By the Lemma's assumption, $\delta > \frac{1}{100}$. Hence $2\ln 2\sqrt{-\log\delta} < 10$. This shows that at least $1 - \frac{\beta}{\sqrt{m}}$ of the elements in K must satisfy

$$\left|\Pr[X_{i_0} = t] - \frac{1}{|K|}\right| < \frac{10}{\beta|K|}$$

□

3 Proof of the XOR Theorem

Theorem 2 (XOR Theorem) *Let $p(n) = n^d$ for some $d > 0$, and $f : \{0,1\}^n \to \{0,1\}$ a function. Let $C : \{0,1\}^{np(n)} \to \{0,1\}$ be a poly-size circuit, such that*

$$\Pr[f_{p(n)}(x) = C(x)] > \frac{1}{2} + \frac{1}{n^c}$$

Where $f_{p(n)}$ is the $p(n)$th-duplication of f and $c < 2d$. Then there exists a poly-size circuit $D : \{0,1\}^n \to \{0,1\}$ such that

$$\Pr[f(x) = D(x)] > 1 - \frac{1}{n^{\frac{d}{2}-c}}$$

Proof We use Lemma 3 with the following parameters:

$$\begin{aligned} K &= \{0,1\}^n \\ m &= p(n) \\ G &= \{x \in K^m \ \mid \ C(x) = f_{p(n)}(x)\} \end{aligned}$$

For any $1 \le i \le p(n)$ and $t \in K$ define $G_i(t)$ to be the set of elements in G whose i'th coordinate is t. Setting $\beta = 10n^c$, Lemma 3 guarantees the existence of $1 \le i_0 \le p(n)$ such that

$$\left|\frac{|G_{i_0}(t)|}{|G|} - \frac{1}{|K|}\right| < \frac{1}{n^c|K|} \qquad (1)$$

For at least $1 - \frac{10n^c}{\sqrt{p(n)}}$ of the $t \in K$. Let $T \subseteq K$ be the set of elements for which the inequality is satisfied. Then, since $p(n) = n^d$,

$$\frac{|T|}{|K|} > 1 - \frac{10n^c}{\sqrt{p(n)}} = 1 - \frac{10}{n^{\frac{d}{2}-c}}$$

An algorithm for building the circuit D can find i_0 by testing all $1 \leq i \leq p(n)$. To test a given i the algorithm would estimate the size of T by randomly picking $x \in K$ and testing if $x \in T$.

By definition of the circuit C we have $\delta = |G|/|K^m| = \frac{1}{2} + \frac{1}{n^c}$. Multiplying inequality (1) by $\frac{|G|}{|K|^{m-1}}$ leads to

$$|\Pr[x \in G_{i_0}(t)] - \delta| < \frac{\delta}{n^c}$$

Hence,

$$\begin{aligned}\Pr[x \in G_{i_0}(t)] \quad &> \quad \delta - \frac{\delta}{n^c} = \delta(1 - \frac{1}{n^c}) = \\ (\frac{1}{2} + \frac{1}{n^c})(1 - \frac{1}{n^c}) &= \frac{1}{2} + \frac{1}{2n^c} - \frac{1}{n^{2c}} > \frac{1}{2} + \frac{1}{4n^c}\end{aligned}$$

and the inequality is satisfied for all $t \in T$. Since T contains most of K, the above inequality implies that for most $t \in K$ the circuit C agrees with $f_{p(n)}$ on the majority of the inputs whose i_0'th coordinate is t.

To construct the required circuit D, randomly pick

$$\begin{array}{cccc} y_1^1 & y_2^1 & \cdots & y_{p(n)-1}^1 \\ y_1^2 & y_2^2 & \cdots & y_{p(n)-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ y_1^w & y_2^w & \cdots & y_{p(n)-1}^w \end{array}$$

where each $y_i^r \in \{0,1\}^n$. The value of w will be determined later. For each $1 \leq r \leq w$ compute

$$\alpha_r = \bigoplus_{j=1}^{p(n)-1} f(y_j^r)$$

Note that computing α_r requires a polynomial number of samples of f. An algorithm for constructing D will need to be able to sample f. For proving the existence of the circuit D it is sufficient to say that these α_i exist.

On input x the circuit D performs the following two operations:

1. For $r = 1, \ldots, w$ constructs the vectors z_r defined by

$$z_r = (y_1^r, \ldots, y_{i_0-1}^r, x, y_{i_0}^r, \ldots y_{p(n)-1}^r)$$

2. Computes the value:

$$D(x) = \mathrm{MAJ}\,\{C(z_r) \oplus \alpha_r \;\; ; \;\; 1 \leq r \leq w\}$$

By definition of T, if $x \in T$ then C agrees with $f_{p(n)}$ on at least $\frac{1}{2} + \frac{1}{4n^c}$ of the elements whose i_0'th coordinate is x. Hence by choosing w large enough, we can satisfy for all $x \in T$ $\Pr[D(x) = f(x)] > 1 - 1/n^{d/2-c}$. The exact bound on w is determined using the Chernoff bounds.

$$\Pr[D(x) \neq f(x)] \quad < \quad \overbrace{\left[1 - \frac{10}{n^{\frac{d}{2}-c}}\right] \frac{1}{n^{\frac{d}{2}-c}}}^{x \in T} + \overbrace{\frac{10}{n^{\frac{d}{2}-c}}}^{x \notin T} < \frac{11}{n^{\frac{d}{2}-c}}$$

This completes the proof of the XOR theorem. □

4 Application to learning

In this section we show that the XOR Theorem can be used to show that given an oracle $\mathcal{O}$ for weakly learning a class of boolean functions $\mathcal{F}$, one can construct a polynomial algorithm A which learns $\mathcal{F}$ in the strong sense. The algorithm A will use the oracle $\mathcal{O}$ as an internal routine. The amplification procedure only uses random samples of the target function. Hence, if $\mathcal{O}$ does not query the function, neither will the algorithm A.

The XOR Theorem can be used to generate good approximations of $f \in \mathcal{F}$ from weak approximations of f_m (f duplicated m times) for some large m. In order for the oracle $\mathcal{O}$ to work on f_m, we must have $f_m \in \mathcal{F}$. This gives rise to the following definition.

Definition 4 *Let $\mathcal{F}$ be a class of boolean functions. Given a function $\tau : N \rightarrow N$ we say that $\mathcal{F}$ is closed under $\tau - \oplus$ iff for all $f \in \mathcal{F}$ over n variables, the $\tau(n)$-th duplication of f is in $\mathcal{F}$.*

As examples, we note that AC^0 is closed under $c - \oplus$ for any constant $c > 0$, and NC^1 is closed under $\mathrm{poly} - \oplus$. We are now ready to state the main result.

Theorem 5 *Let $\mathcal{F}$ be a class of boolean functions closed under $\mathrm{poly} - \oplus$. Given an oracle $\mathcal{O}$ that weakly learns $\mathcal{F}$, there is a polynomial algorithm that strongly learns $\mathcal{F}$.*

Proof Let $\delta > 0$ and $f \in \mathcal{F}$. We wish to construct an algorithm A that with probability at least $1 - \delta$ will output a circuit D such that $\Pr[D(x) = f(x)] > 1 - \frac{1}{n}$.

Let $q(n)$ be a polynomial such that for any $f \in \mathcal{F}$ the oracle $\mathcal{O}$ can build approximations C of f such that $\Pr[C(x) = f(x)] > \frac{1}{2} + \frac{1}{q(n)}$. Define $p(n) = n^{2\deg(q)+1}$ By $f_{p(n)}$ we mean the $p(n)$'th duplication of f. The algorithm A will use the oracle $\mathcal{O}$ to build a circuit C such that with probability at least $1 - \frac{\delta}{2}$, C satisfies

$$\Pr[C(x) = f_{p(n)}] > \frac{1}{2} + \frac{1}{q(n)}$$

The procedure described in the proof the XOR Theorem can now be used to build the circuit D such that

$$\Pr[D(x) = f(x)] > 1 - \frac{1}{n}$$

Note that the entire amplification procedure does not query f. Only random samples of f are used. Hence if the oracle $\mathcal{O}$ does not query f then neither does the entire amplification process.

The oracle $\mathcal{O}$ will succeed in building the circuit C with probability of at least $1 - \frac{\delta}{2}$. The construction of the circuit D can be made to succeed with probability of at least $1 - \frac{\delta}{2}$. Hence the entire construction will succeed with probability at least $1 - \delta$. This completes the proof of Theorem 5. □

References

[BL93] D. Boneh, R. Lipton, "Amplification of Weak Learning Under the Uniform Distribution", in COLT 1993.

[CK81] I. Csiszár, J. Körner, "Information Theory: Coding theorems for discrete memoryless ststems", Academic Press, 1981.

[Fre90] Y. Freund, "Boosting a weak learning algorithm by majority", in COLT 1990, pp. 202-231.

[Fre92] Y. Freund, "An improved boosting algorithm and its implications on learning complexity", in COLT 1992, pp. 391-398.

[Kra86] E. Kranakis, "Primality and Cryptography", Wiley-Teubner Series in Computer Science, 1986.

[KV89] M. Kearns, L. G. Valiant, "Crytographic limitations on learning Boolean formulae and finite automata", in 21st ACM STOC, pp. 433-444, 1989.

[Lev87] L. A. Levin, "One-Way Functions and Pseudorandom Generators", Combinatorica, 7(4):357-363, 1987. Also STOC 1985.

[Sch90] R. E. Schapire, "The strength of weak learnablility", Machine Learning, 5(2):197-227, 1990.

[Val84] L. G. Valiant, "A theory of the learnable", Communications of the ACM, 27(11):1134-1142, 1984.

[Yao82] A. C. Yao, "Theory and Application of Trapdoors Functions", in 23rd IEEE FOCS, pp. 80-91, 1982.