A Bound on the Precision Required to Estimate a Boolean Perceptron from its Average Satisfying Assignment *

Paul W. Goldberg†
University of Warwick,
Department of Computer Science,
Coventry, CV4 7AL, UK.

November 17, 2005

Abstract

A boolean perceptron is a linear threshold function over the discrete boolean domain $\{0,1\}^n$. That is, it maps any binary vector to 0 or 1 depending on whether the vector's components satisfy some linear inequality. In 1961, Chow showed that any boolean perceptron is determined by the average or "center of gravity" of its "true" vectors (those that are mapped to 1), together with the total number of true vectors. Moreover, these quantities distinguish the function from any other boolean function, not just other boolean perceptrons.

In this paper we go further, by identifying a lower bound on the Euclidean distance between the average satisfying assignment of a boolean perceptron, and the average satisfying assignment of a boolean function that disagrees with that boolean perceptron on a fraction ϵ of the input vectors. The distance between the two means is shown to be at least $(\epsilon/n)^{O(\log(n/\epsilon)\log(1/\epsilon))}$. This is motivated by the statistical question of whether an empirical estimate of this average allows us to recover a good approximation to the perceptron. Our result provides a mildly super-polynomial upper bound on the growth rate of the sample size required to learn boolean perceptrons in the "restricted focus of attention" setting. In the process we also find some interesting geometrical properties of the vertices of the unit hypercube.

Key words. Boolean functions, threshold functions, geometry, inductive learning
AMS Subject classification. 68Q15, 68Q32, 52C07, 52C35
Abbreviated Title. Estimating a Boolean Perceptron

*This work was supported by EPSRC Grant GR/R86188/01. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the author's views. A preliminary version of this paper was presented at the 2001 COLT conference.

†author's home page: `http://www.dcs.warwick.ac.uk/~pwg/`

1 Introduction

A *boolean perceptron* is a linear threshold function over the domain of 0/1-vectors. (Subsequently we usually just say "perceptron", and omit the adjective "boolean".) Thus it is specified by a weight vector $\mathbf{w}$ of n real numbers and a real-valued threshold t, and it maps a binary vector $\mathbf{x}$ to the output value 1 provided that $\mathbf{w}.\mathbf{x} \geq t$, otherwise it maps $\mathbf{x}$ to 0.

In this paper we consider the problem of estimating a perceptron from an approximate value of the mean, or "center of gravity" of its satisfying assignments. Chow (9) originally showed that any boolean perceptron is identified by the exact value of the average of its satisfying assignments, along with the number of satisfying assignments, in the sense that there are no other boolean functions of any kind for which the average and number of satisfying assignments is the same. The question of the extent to which an approximation to the average determines the perceptron, is equivalent to a problem of learning boolean perceptrons in the "restricted focus of attention" setting, described below.

The *Chow parameters* of a boolean function are the coordinates of the vector sum of the satisfying vectors, together with the number of satisfying vectors. Subject to a uniform distribution over boolean vectors, these are essentially equivalent to the conditional probabilities that the i-th component of $\mathbf{x}$ is equal to 1, conditioned on $\mathbf{x}$ being a satisfying assignment. Letting y denote the output value and $\mathbf{x} = ((\mathbf{x})_1, \ldots, (\mathbf{x})_n)$, these are the probabilities $\Pr((\mathbf{x})_i = 1 \mid y = 1)$, for $i = 1, \ldots, n$, together with the value $\Pr(y = 1)$.[1] Chow's result says that these values uniquely define the function, provided that it is a boolean perceptron. (Bruck (8) shows, more generally, that a threshold function G over a set of monomials is characterised by the spectral coefficients of G that correspond to those monomials.) Hence a weights-based parametrisation $(\mathbf{w}, t)$ should in principle be derivable from the Chow parameters; there will be some amount of freedom for $(\mathbf{w}, t)$ to vary while preserving the functional behaviour on binary inputs.

In this paper we show that additive approximations of the Chow parameters determine the approximate behaviour of the function, to within a mildly super-polynomial factor. That is in contrast to the situation for the weights-based parametrisation of a perceptron, for which a tiny perturbation of some parameter may result in a large change to the set of points that are mapped to output value 1. In this sense the Chow parameters, as a description of a boolean perceptron, are a more robust parametrisation.

1.1 Background and Previous Results

Chow's paper gave rise to subsequent work that addressed the algorithmic problem of recovering a weights-based parametrisation of a perceptron from its Chow parameters. This problem and related ones were later reconsidered in the computational learning theory literature, notably work on PAC-learning in the so-called "restricted focus of attention" setting.

Earlier work that followed on from (9) includes an algorithm by Kaszerman (16) for recovering a linear threshold function from its Chow parameters. The algorithm is iterative and somewhat

[1] If the coordinates of the sum of all satisfying vectors are rescaled down by the number of satisfying vectors, one obtains the average satisfying assignment, whose coordinates are the probabilities $\Pr((\mathbf{x})_i = 1 \mid y = 1)$. The Chow parameters are recovered by multiplying this average by $2^n \cdot \Pr(y = 1)$.

related to the perceptron algorithm (19); it does not have a good bound on the number of iterations and assumes exact values of the parameters are given. A paper of Winder (20) compares 7 functions (4 of which were proposed in previous papers) for rescaling Chow parameters to obtains weights for a linear-threshold function. None of these functions has perfect performance, and it is uncertain that any function exists from individual Chow parameters to good weights – it may be necessary to deal with them collectively rather than individually. A further paper by Winder (21) investigates the class of boolean functions that are uniquely defined by their Chow parameters, and shows among other things that it lies properly between the class of linear threshold functions and the class of monotonic functions.

The problem of learning a function f means reconstructing it (exactly or approximately) from a limited collection of observations of its input vectors $\mathbf{x}$ and associated values $f(\mathbf{x})$. There is much known about learning boolean perceptrons in various settings, for example irrelevant attributes (17), classification noise (6) and learning from a source of "helpful" examples (2). Special cases include monomials, decision lists (18; 12) and boolean threshold functions. Further knowledge on this topic occurs in the more general context of perceptrons over the real as opposed to the boolean domain. An example is that they may be PAC-learned in time polynomial in the dimension n and the PAC parameters ϵ and δ, using the VC dimension theory (7). Chapter 24 of (1) and references therein is a good introduction to results on learning boolean perceptrons.

Restricted focus of attention (RFA) learning was introduced and developed in the papers (3; 4; 5). The k-RFA setting (where k is a positive integer) allows an algorithm to see only a subset of size k of the input attributes of any training example. The usual assumption has been that the distribution of input vectors $\mathbf{x}$ is known to be a product distribution (with no other information given about it). Clearly, 1-RFA learning (in which only one input attribute of each example is visible) is a very restrictive setting, making positive results of particular interest. In (13) we studied in detail the problem of learning linear-threshold functions over the real domain in the 1-RFA setting, so that each example of input/output behaviour of the target function, has only a single input component value, together with the binary value of the output, revealed to the learning algorithm. We showed that the input distribution (in (13), not necessarily a product distribution) needs to be at least partly known, and that the sample-size required for learning depends sensitively on the input distribution. We identified measures of "well-behavedness" of the input distribution and gave sample size bounds in terms of these measures.

This paper addresses the topic of 1-RFA learning of perceptrons where the input distribution is uniform over V, the vertices of the unit hypercube. From (5) we have that a random sample of 1-RFA data is equivalent, in terms of the information it conveys, to approximations of the conditional probabilities $\Pr(y = 1 \mid (\mathbf{x})_i = b)$, for $b \in \{0, 1\}$ (where $(\mathbf{x})_i$ denotes the i-th component of $\mathbf{x}$), together with the probability $\Pr(y = 1)$, and these approximations have additive error inversely proportional to the sample size. The coordinates of the average satisfying assignment are related as follows:

$$\Pr((\mathbf{x})_i = 1 \mid y = 1) = \frac{\Pr((\mathbf{x})_i = 1)}{\Pr(y = 1)} \Pr(y = 1 \mid (\mathbf{x})_i = 1) = \frac{1}{2\Pr(y = 1)} \Pr(y = 1 \mid (\mathbf{x})_i = 1)$$

Provided that $\Pr(y = 1)$ is not too small, we obtain good estimates of the coordinates of the average satisfying assignment from estimates of probabilities $\Pr(y = 1 \mid (\mathbf{x})_i = 1)$ (and vice versa). Our analysis handles low values of $\Pr(y = 1)$ as a special case.

The reason why the uniform distribution on V (for which bounds of (13) are inapplicable) is of particular interest is that it is the most natural and widely studied input distribution from the perspective of computational learning theory. The question of whether this learning problem is solvable with polynomial time or sample size is previously discussed in (10) and (13), and is currently known to be solvable under the restriction that weights are polynomially bounded. Birkendorf et al. (5) suggest the following rule: for $1 \leq i \leq n$ and $b \in \{0, 1\}$, let p^i_b be the observed conditional probability $\Pr(y = 1 \mid (\mathbf{x})_i = b)$ and let $p = \Pr(y = 1)$. Then take $\mathbf{x}$ to be a positive instance if $\frac{1}{n}\sum_{i=1}^{n} p^i_{(\mathbf{x})_i} > p$, otherwise label $\mathbf{x}$ as negative. It is left as an open problem whether the rule is valid.

We show here that given a perceptron F, and any boolean function that disagrees with F on at least a fraction ϵ of input vectors, their average satisfying assignments must differ by $(\epsilon/n)^{O(\log(n/\epsilon)\log(1/\epsilon)}$ in the L_2 metric. The computational learning-theoretic result that follows is a mildly super-polynomial bound (of the order of $\log(\delta^{-1})(n/\epsilon)^{O(\log(n/\epsilon)\log(1/\epsilon))}$) on the asymptotic growth rate of sample-size requirement for PAC-learning a perceptron from 1-RFA data. This is a purely "information-theoretic" result; we do not have any algorithm whose runtime has an asymptotic growth rate that improves substantially on a brute-force approach.

1.2 Notation and Terminology

Let V be the input domain, i.e. the vertices of the unit hypercube, or 0/1-vectors. By a *vertex* we mean a member of V, i.e. a 0/1-vector of length n.

F will denote a boolean perceptron, typically the "target function", and G will denote a boolean function (not necessarily a boolean perceptron), for example an estimate of F returned by an algorithm. The *positive* (respectively, *negative*) examples of a function are those that are mapped to 1 (respectively, 0). Let $pos(F)$, $neg(F)$, $pos(G)$, $neg(G)$ denote the positive and negative examples of F and G. (So $pos(F) = \{F^{-1}(1)\}$, etc.) F and G divide V into 4 subsets defined as follows.

$$\begin{array}{ll} V_{00} = neg(F) \cap neg(G) & V_{01} = neg(F) \cap pos(G) \\ V_{10} = pos(F) \cap neg(G) & V_{11} = pos(F) \cap pos(G) \end{array}$$

For $R \subseteq \mathbb{R}^n$, let $m(R)$ be the number of elements of V that lie in R. Let $a(R)$ be the vector sum of elements of $V \cap R$. Let $\mu(R)$ denote the (unweighted) average of members of V that lie in the region R, so that $\mu(R) = a(R)/m(R)$, well-defined provided that $m(R) > 0$. The region of disagreement of F and G is $V_{01} \cup V_{10}$; thus the disagreement rate between F and G, over the uniform distribution on V, is $(m(V_{01}) + m(V_{10}))/2^n$.

Throughout, logarithms are to the base 2.

When we refer to subspaces, or spanning, or dimension, we mean in the affine sense, so that a "subspace" does not necessarily contain the origin, and the spanning set of $S \subseteq \mathbb{R}$, denoted $\mathrm{Span}(S)$, is the set of points that are expressible as the sum of one member of the spanning set plus a weighted sum of differences between pairs of points in S. A *line* means a 1-dimensional affine subspace.

We adopt the following usage of alphabetic symbols throughout the paper, which extends to variants embellished with primes or subscripts.

1. H denotes a hyperplane in $\mathbb{R}^n$ (an affine subspace with dimension $n-1$).

2. A denotes an affine subspace with possibly lower dimension.

3. S denotes a finite set of points in $\mathbb{R}^n$.

4. A point in $\mathbb{R}^n$ or a n-dimensional vector will be denoted by a lower-case bold-face letter such as $\mathbf{x}$, and $(\mathbf{x})_i$ denotes the i-th entry or component of $\mathbf{x}$. $\mathbf{v}$ is used to denote an element of V.

For $\mathbf{x} = ((\mathbf{x})_1, \ldots, (\mathbf{x})_n)$ let $\|\mathbf{x}\|$ denote the Euclidean norm of $\mathbf{x}$, i.e. $(\sum_{i=1}^n ((\mathbf{x})_i)^2)^{1/2}$. Let $d_E(\mathbf{x}, Z)$ denote the Euclidean distance between $\mathbf{x} \in \mathbb{R}^n$ and the closest point to $\mathbf{x}$ in $Z \subseteq \mathbb{R}^n$.

2 Geometric Results

In this section we give various geometric results about the vertices of the unit hypercube, which we use in Section 3 to deduce the bound on sample-size requirement in the inductive learning context described in the last section. We start with an informal summary of the results of this section.

1. Lemma 1 gives a simple upper bound on the number of elements of V contained in a linear subspace, in terms of the dimension of that subspace.

2. Theorem 1 shows that if a hyperplane contains a large number of elements of V, then the coefficients of that hyperplane have a large common denominator (a lower bound on the common denominator is given in terms of the number of elements of V contained by the hyperplane).

3. Theorem 2 uses Theorem 1 to show that any hyperplane that "narrowly misses" a large fraction of V can be perturbed slightly so that it actually contains all those vertices. The resulting hyperplane does no longer "narrowly miss" any other vertices. More precisely, if a hyperplane comes within distance $O((1/\alpha)(n \log(n/\alpha))^{\log(n/\alpha)})$ of a fraction α of the 2^n vertices, then all those $\alpha \cdot 2^n$ vertices lie on the perturbed hyperplane.

4. Theorem 3 uses Theorem 2 to derive a lower bound on the distance between $\mu(V_{01})$ and $\mu(V_{10})$ (the means of the two regions of disagreement between two boolean functions, one of which is a perceptron) in terms of their disagreement rate $m(V_{01} \cup V_{10})/2^n$.

Lemma 1 *Any affine subspace A of $\mathbb{R}^n$ of dimension d contains at most 2^d elements of the vertices of the unit hypercube.*

Proof. By induction on d. It clearly holds for $d = 0$, when A consists of a single point.

Suppose $d > 0$. Assume that A contains at least 2 elements of V (if not, we are done). For $\mathbf{v}_1, \mathbf{v}_2 \in V \cap A$, suppose that $\mathbf{v}_1$ and $\mathbf{v}_2$ differ in the i-th component, so that $(\mathbf{v}_1)_i \neq (\mathbf{v}_2)_i$.

Divide V into two subcubes V' and V'', where V' is elements $\mathbf{v} \in V$ such that $(\mathbf{v})_i = 0$ and V'' is elements $\mathbf{v} \in V$ with $(\mathbf{v})_i = 1$. By construction, $A \cap V' \neq \emptyset$ and $A \cap V'' \neq \emptyset$.

Since A intersects V', we have $A \cap \mathrm{Span}(V'')$ is a proper subspace of A, and similarly, $A \cap \mathrm{Span}(V')$ is a proper subspace of A. The inductive hypothesis tells us that each of these subspaces contains at most 2^{d-1} elements of V, for a total of at most 2^d elements of V, as required. $\square$

Observation 1 *Let $S \subseteq V$, $|S| = \alpha \cdot 2^n$ (where $0 \leq \alpha \leq 1$). Let $d = n - \lfloor \log(1/\alpha) \rfloor - 1$. Then given any subset of size d of the n components, there exist two distinct elements of S that agree on all those d components.*

Proof. At most 2^d elements of V can be distinguished from each other via their values on a set of d coordinates. We assumed that $|S| = \alpha \cdot 2^n$. Since $d = n - \lfloor \log(1/\alpha) \rfloor - 1$, we can deduce that $\alpha > 2^{d-n}$, hence $|S| > 2^d$. By the pigeon hole principle, two distinct elements of S agree on the d coordinates. $\square$

Theorem 1 *Let H be a hyperplane in $\mathbb{R}^n$, and suppose that H contains a fraction α of the vertices of the unit hypercube and that H is spanned by the vertices that it contains. Suppose that H is described as the set of points $\{\mathbf{x} : \mathbf{w}.\mathbf{x} = t\}$, with parameters $(\mathbf{w}, t)$ rescaled so that $\|\mathbf{w}\| = 1$. Then all the components of $\mathbf{w}$ are integer multiples of some quantity at least as large as*

$$\left(\sqrt{n}(1 + \lfloor \log(1/\alpha) \rfloor)! n^{(1+\lfloor \log(1/\alpha) \rfloor)}\right)^{-1}.$$

Proof. We construct a linear system that must be satisfied by the weights $\{(\mathbf{w})_i : 1 \leq i \leq n\}$, such that when we solve it (invert a matrix) elements of the inverted matrix have a large common denominator. Initially the system will be satisfied by the $(\mathbf{w})_i$ values when they are rescaled so that their maximum (in absolute value) is equal to 1. Afterwards we will rescale so that $\|\mathbf{w}\| = 1$.

Let $x_1 \in \arg\max_i(|(\mathbf{w})_i|)$. The first linear equality is $(\mathbf{w})_{x_1} = 1$. This does the job of rescaling the $(\mathbf{w})_i$ values such that their maximum (in absolute value) is 1.

Let $d = n - \lfloor \log(1/\alpha) \rfloor - 1$, as in Observation 1. For $\mathbf{v} \in V$, $(\mathbf{v})_i$, the i-th component of $\mathbf{v}$, is equal to 0 or 1. We identify a subset of the component indices $\{x_2, \ldots, x_d\} \subseteq \{1, \ldots, n\}$ together with $2(d-1)$ vertices $\{\mathbf{v}_2, \mathbf{v}'_2, \ldots, \mathbf{v}_d, \mathbf{v}'_d\} \subseteq H \cap V$ such that

$$\begin{array}{rl} (\mathbf{v}_j)_{x_j} - (\mathbf{v}'_j)_{x_j} = 1 & \text{for} \quad 2 \leq j \leq d \\ (\mathbf{v}_j)_{x_i} = (\mathbf{v}'_j)_{x_i} & \text{for} \quad 2 \leq j \leq d; 1 \leq i \leq d; j \neq i. \end{array}$$

For $\mathbf{v}, \mathbf{v}' \in H \cap V$, $\mathbf{w}$ satisfies $(\mathbf{v} - \mathbf{v}').\mathbf{w} = 0$. The next $d-1$ linear equalities are $(\mathbf{v}_j - \mathbf{v}'_j).\mathbf{w} = 0$, for $2 \leq j \leq d$. These linear constraints on $\mathbf{w}$ are independent of each other, since for the subset $\{x_2, \ldots, x_d\} \subset \{1, \ldots, n\}$, the linear constraint $(\mathbf{v}_j - \mathbf{v}'_j).\mathbf{w} = 0$ has coefficient 1 on the x_j-th component of $\mathbf{w}$ and 0 on the other components in L_d. We continue by showing how to find a suitable set $\{\mathbf{v}_2, \mathbf{v}'_2, \ldots, \mathbf{v}_d, \mathbf{v}'_d\}$. Let

$$\begin{array}{rcl} R_1 & = & \{1, \ldots, n\} \setminus \{x_1\} \\ L_1 & = & \{x_1\} \end{array}$$

Choose $\mathbf{v}_2, \mathbf{v}'_2 \in H \cap V$ such that

$$\{\mathbf{v}_2, \mathbf{v}'_2\} \in \arg \max_{\{\mathbf{v}, \mathbf{v}'\} \subseteq H \cap V; \mathbf{v} \neq \mathbf{v}'; (\mathbf{v})_\ell = (\mathbf{v}')_\ell \text{ for } \ell \in L_1} \left(|\{i \in R_1 : (\mathbf{v})_i = (\mathbf{v}')_i\}|\right).$$

Thus $\mathbf{v}_2$ and $\mathbf{v}'_2$ are chosen to be two distinct vertices in $H \cap V$ that have minimum Hamming distance from each other, subject to the requirement that they agree on component x_1.

Since $\mathbf{v}_2 \neq \mathbf{v}'_2$, there exists $x_2 \in R_1$ such that $(\mathbf{v}_2)_{x_2} \neq (\mathbf{v}'_2)_{x_2}$. We may assume that $(\mathbf{v}_2)_{x_2} = 1$ and $(\mathbf{v}'_2)_{x_2} = 0$. Let

$$\begin{aligned} R_2 &= \{i \in R_1 \ : \ (\mathbf{v}_2)_i = (\mathbf{v}'_2)_i\} \\ L_2 &= \{x_1, x_2\}. \end{aligned}$$

R_2 is a maximal subset of R_1 such that two distinct vertices agree on coordinates indexed by R_2 and L_1. By Observation 1, $|R_2| \geq n - \lfloor \log(1/\alpha) \rfloor - 2$.

Generally, for $j > 2$, construct $x_j \in R_{j-1}$ and $R_j \subseteq R_{j-1} \setminus \{x_j\}$ and $L_j = L_{j-1} \cup \{x_j\}$ as follows. Choose $\mathbf{v}_j, \mathbf{v}'_j \in H \cap V$ such that

$$\{\mathbf{v}_j, \mathbf{v}'_j\} \in \arg \max_{\{\mathbf{v},\mathbf{v}'\} \subseteq H \cap V; \mathbf{v} \neq \mathbf{v}'; (\mathbf{v})_\ell = (\mathbf{v}')_\ell \text{ for } \ell \in L_{j-1}} \Big(\big| \{i \in R_{j-1} \ : \ (\mathbf{v})_i = (\mathbf{v}')_i\} \big| \Big).$$

Thus $\mathbf{v}_j$ and $\mathbf{v}'_j$ are chosen to be two distinct vertices in $H \cap V$ that have minimum Hamming distance over coordinates indexed by R_{j-1}, subject to the constraint that they agree on coordinates indexed by L_{j-1}.

We claim that there exists $x_j \in R_{j-1}$ such that $(\mathbf{v}_j)_{x_j} \neq (\mathbf{v}'_j)_{x_j}$.

Suppose that the claim were false. Then $(\mathbf{v}_j)_i = (\mathbf{v}'_j)_i$ for all $i \in R_{j-1}$, and $(\mathbf{v}_j)_\ell = (\mathbf{v}'_j)_\ell$ for all $\ell \in L_{j-1}$ (and note that for $\ell \in L_{j-1}$, $\ell \notin R_{j-1}$). This contradicts the choice of $\{\mathbf{v}_{j-1}, \mathbf{v}'_{j-1}\}$ as a pair of vertices that have minimum Hamming distance on coordinates indexed by R_{j-2} (which contains R_{j-1}) while also agreeing on coordinates indexed by L_{j-2}. Note that

1. $\mathbf{v}_{j-1}$ and $\mathbf{v}'_{j-1}$ agree on coordinates indexed by L_{j-2}. They agree on $|R_{j-1}|$ elements of R_{j-2}.

2. $\mathbf{v}_j$ and $\mathbf{v}'_j$ agree on coordinates indexed by $L_{j-1} = L_{j-2} \cup \{x_{j-1}\}$, where $x_{j-1} \in R_{j-2}$. They also agree on all elements of $R_{j-1} \subseteq R_{j-2}$.

3. From the above two points, amongst pairs of vertices $\mathbf{v}$ and $\mathbf{v}'$ that agree on L_{j-2}, $\mathbf{v}_j$ and $\mathbf{v}'_j$ agree on more elements of R_{j-2} than do $\mathbf{v}_{j-1}$ and $\mathbf{v}'_{j-1}$.

Hence there exists $x_j \in R_{j-1}$ such that $(\mathbf{v}_j)_{x_j} \neq (\mathbf{v}'_j)_{x_j}$, and we can assume $(\mathbf{v}_j)_{x_j} = 1$ and $(\mathbf{v}'_j)_{x_j} = 0$. Let

$$\begin{aligned} R_j &= \{i \in R_{j-1} \ : \ (\mathbf{v}_j)_i = (\mathbf{v}'_j)_i\} \\ L_j &= L_{j-1} \cup \{x_j\}. \end{aligned}$$

R_j is a maximal subset of R_{j-1} (where $|R_{j-1}| \geq n - \lfloor \log(1/\alpha) \rfloor - (j-1)$) such that $\mathbf{v}_j$ agrees with $\mathbf{v}'_j$ on coordinates indexed by R_j (and the $j-1$ coordinates indexed by L_{j-1}). By Observation 1, $|R_j| \geq n - \lfloor \log(1/\alpha) \rfloor - j$.

Recall that $d = n - \lfloor \log(1/\alpha) \rfloor - 1$ as in Observation 1. Since $|R_j| \geq n - \lfloor \log(1/\alpha) \rfloor - j$, the above construction can be carried out for $2 \leq j \leq d$.

By our assumption that $\mathrm{Span}(H \cap V) = H$, there exists a set $\{\mathbf{v}_{d+1}, \mathbf{v}'_{d+1} \ldots, \mathbf{v}_n, \mathbf{v}'_n\} \subset H \cap V$ such that each pair of vertices $\{\mathbf{v}_j, \mathbf{v}'_j\}$ for $d+1 \leq j \leq n$, imposes on $\mathbf{w}$ a new linear constraint $(\mathbf{v}_j - \mathbf{v}'_j).\mathbf{w} = 0$ that is linearly independent of the others.

Let M be a matrix whose first row is all zero apart from the x_1-th entry which contains the value 1. The j-th row (for $2 \leq j \leq n$) is the components of $(\mathbf{v}_j - \mathbf{v}'_j)$. We have $M.\mathbf{w} = \mathbf{r}$ where

$\mathbf{r}$ is all zero apart from $(\mathbf{r})_1 = 1$. Now rearrange the columns of M in the order $x_1, \ldots, x_n$ (where $\{x_{d+1}, \ldots, x_n\} = \{1, \ldots, n\} \setminus \{x_1, \ldots, x_d\}$) and let $\mathbf{r} = (1, 0, \ldots, 0)^T$. We have constructed a linear system $M.\mathbf{w}^P = \mathbf{r}$ where $\mathbf{w}^P$ is a permutation of $\mathbf{w}$ and

1. M is an invertible $n \times n$ matrix with entries in $\{0, 1, -1\}$.

2. the $d \times d$ submatrix of M comprising the first d rows and columns, is the identity matrix.

3. $\mathbf{r} = (1, 0, \ldots, 0)^T$.

Hence $\mathbf{w}^P = M^{-1}\mathbf{r}$. The (i,j)-th entry of M^{-1} is given by $\det(M_{i,j})/\det(M)$ where $\det(M)$ denotes the determinant of matrix M, and $M_{i,j}$ is the submatrix of M obtained by removing column i and row j. We will upper-bound the determinant of M.

Construct M' by adding (respectively, subtracting) row j (for $1 \leq j \leq d$) to row j' (for $d+1 \leq j' \leq n$) whenever the j-th entry of row j' is equal to -1 (respectively, 1). $M' = (m)_{ij}$ satisfies

$$\begin{aligned} m_{ij} &= 0 \quad \text{for } d+1 \leq i \leq n, 1 \leq j \leq d \\ -n \leq m_{ij} &\leq n \quad \text{for } d+1 \leq i \leq n, d+1 \leq j \leq n \end{aligned}$$

$\det(M') = \det(M)$, the first d rows and columns of M' is still the identity matrix, so from the features of M' noted above, $\det(M')$ is equal to $\det(M'')$ where M'' is the $(n-d) \times (n-d)$ submatrix of M' in the bottom right-hand corner of M'.

Now observe that the determinant of any $i \times i$ matrix with entries in $\{-n, -(n-1), \ldots, n-1, n\}$ is upper bounded[2] by $i!n^i$, so that $|\det(M)| \leq (n-d)!n^{n-d}$. Accordingly entries of M^{-1} (and consequently, components of $\mathbf{w}$) must be integer multiples of a quantity greater than or equal to

$$\Big((n-d)!n^{n-d}\Big)^{-1} = \Big((1 + \lfloor\log(1/\alpha)\rfloor)!n^{(1+\lfloor\log(1/\alpha)\rfloor)}\Big)^{-1}$$

and so components of $\mathbf{w}$ are also integer multiples of this quantity.

The maximum absolute value of a component of $\mathbf{w}$ (or $\mathbf{w}^P$) is 1, so $1 \leq \|\mathbf{w}\| \leq \sqrt{n}$. Rescaling $\mathbf{w}$ to get $\|\mathbf{w}\| = 1$, components of $\mathbf{w}$ are integer multiples of a quantity at least as large as the above, divided by $\sqrt{n}$. That is, $\Big(\sqrt{n}(1 + \lfloor\log(1/\alpha)\rfloor)!n^{(1+\lfloor\log(1/\alpha)\rfloor)}\Big)^{-1}$ as in the statement of the theorem. □

We use Theorem 1 to prove the following.

Theorem 2 *Given any hyperplane in* $\mathbb{R}^n$ *whose* β*-neighborhood contains a subset* S *of vertices of the unit hypercube, where* $|S| = \alpha \cdot 2^n$*, there exists a hyperplane which contains all elements of* S *provided that*

$$0 \leq \beta \leq \Big((2/\alpha) \cdot n^{(5+\lfloor\log(n/\alpha)\rfloor)} \cdot (2 + \lfloor\log(n/\alpha)\rfloor)!\Big)^{-1}$$

[2]There is not a substantially better upper bound on the determinant of this matrix that uses the fact that it is over integers with absolute value at most n; from Hadamard (14) the determinant of a $i \times i$ matrix over $\{1, -1\}$ may be as high as $i^{i/2}$. This becomes $n^i.i^{i/2}$ when the entries 1 and -1 are replaced with n and $-n$ respectively.

Proof. Let $H = \{\mathbf{x} \ : \ \mathbf{w}.\mathbf{x} = t\}$, where by rescaling we can assume $\|\mathbf{w}\| = 1$. Assume that the β-neighborhood of H contains S. Then for $\mathbf{v} \in S$, we have $\mathbf{w}.\mathbf{v} \in [t - \beta, t + \beta]$.

Define a new weight vector $\mathbf{w}'$ derived from $\mathbf{w}$ by taking each weight in $\mathbf{w}$ and rounding it off to the nearest integer multiple of β (rounding down in the event of a tie). Then we claim that scalar products $\mathbf{w}'.\mathbf{v}$ can take at most $n+2$ distinct values for $\mathbf{v} \in S$. To see this, note that for $\mathbf{v} \in S$,

1. $\mathbf{w}'.\mathbf{v} < \mathbf{w}.\mathbf{v} + n\beta/2 \leq t + \beta + n\beta/2$
2. $\mathbf{w}'.\mathbf{v} \geq \mathbf{w}.\mathbf{v} - n\beta/2 \geq t - \beta - n\beta/2$
3. $\mathbf{w}'.\mathbf{v}$ is an integer multiple of β for $\mathbf{v} \in V$.

Items (1) and (2) show that $\mathbf{w}'.\mathbf{v}$ lies in a semi-open interval of length $\beta(n+2)$, and with (3) there are only at most $(n+2)$ possible values in the interval. Let T be the set of these $n+2$ values.

Let t' be the member of T which maximizes the number of vertices $\mathbf{v} \in S$ satisfying $\mathbf{w}'.\mathbf{v} = t'$. Then there are at least $\alpha \cdot 2^n/(n+2)$ vertices $\mathbf{v} \in S$ that satisfy $\mathbf{w}'.\mathbf{v} = t'$. Let

$$\begin{array}{rcl} A_1 & = & \mathrm{Span}(\{\mathbf{v} \in S \ : \ \mathbf{w}'.\mathbf{v} = t'\}) \\ H_1 & = & \{\mathbf{x} \in \mathbb{R}^n \ : \ \mathbf{w}'.\mathbf{x} = t'\} \end{array}$$

$|A_1 \cap V| \geq \alpha \cdot 2^n/(n+2)$, hence by Lemma 1,

$$\dim(A_1) \geq n - \log(1/\alpha) - \log(n+2). \tag{1}$$

We next show that for all $\mathbf{v} \in S$,

$$d_E(\mathbf{v}, H_1) \leq 2n\beta. \tag{2}$$

$\|\mathbf{w}' - \mathbf{w}\| \leq \sqrt{n}\beta/2$. $\|\mathbf{w}\| = 1$ and since the Euclidean norm is a metric,

$$\|\mathbf{w}'\| \in [1 - \sqrt{n}\beta/2, 1 + \sqrt{n}\beta/2].$$

For $\mathbf{v} \in S$, $\mathbf{w}'.\mathbf{v} - t' \in [-(n+2)\beta, (n+2)\beta]$. Let $(\mathbf{w}'', t'')$ be $(\mathbf{w}', t')$ rescaled so that $\|w''\| = 1$. Then

$$\begin{array}{rrcll} & \mathbf{w}''.\mathbf{v} - t'' & \in & [-(n+2)\beta/(1 - \sqrt{n}\beta/2), (n+2)\beta/(1 - \sqrt{n}\beta/2)] & \\ \Rightarrow & \mathbf{w}''.\mathbf{v} - t'' & \in & [-2n\beta, 2n\beta] & (\text{since } \sqrt{n}\beta \ll 1) \\ \Rightarrow & \mathbf{w}''.\mathbf{v} & \in & [t'' - 2n\beta, t'' + 2n\beta] & \end{array}$$

Since $\|\mathbf{w}''\| = 1$, $\mathbf{v}$ is within Euclidean distance $2n\beta$ of H_1. This establishes Equation (2).

We want to show that $\dim(\mathrm{Span}(S)) \leq n - 1$. We next find a hyperplane H_k that contains A_1 and other elements of S such that $\mathrm{Span}(H_k \cap S) = H_k$ (allowing Theorem 1 to apply to H_k) and such that we also obtain a bound on $d_E(\mathbf{v}, H_k)$ for $\mathbf{v} \in S$.

We know that $\dim(A_1) < n$. If $\dim(A_1) = n - 1$ then put $k = 1$ and use $H_k = H_1 = A_1$. Suppose that $\dim(A_1) < n - 1$. Then let A_1' be a subspace of H_1 such that $\dim(A_1') = n - 2$ and $A_1 \subseteq A_1'$. Let $\mathbf{v}_1 \in \arg\max_{\mathbf{v} \in S}(d_E(\mathbf{v}, A_1'))$.

Let H_2 be the hyperplane $\mathrm{Span}(A_1' \cup \{\mathbf{v}_1\})$. Then for all $\mathbf{v} \in S$, using (2),

$$d_E(\mathbf{v}, H_2) \leq d_E(\mathbf{v}, H_1) + d_E(\mathbf{v}_1, H_1) \leq 4n\beta.$$

Let $A_2 = \mathrm{Span}(A_1 \cup \{\mathbf{v}_1\})$. Since $\mathbf{v}_1 \notin A_1$ we have $\dim(A_2) = \dim(A_1) + 1$.

Generally, for $j \geq 1$, if $A_j \subset H_j$, $A_j \neq H_j$, construct A_{j+1} and H_{j+1} from A_j and H_j as follows. Choose A'_j of dimension $n-2$ such that

$$A_j \subseteq A'_j \subset H_j.$$

Then choose

$$\mathbf{v}_j \in \arg\max_{\mathbf{v}\in S}(d_E(\mathbf{v}, A'_j)).$$

Then let $H_{j+1} = \mathrm{Span}(A'_j \cup \{\mathbf{v}_j\})$ and $A_{j+1} = \mathrm{Span}(A_j \cup \{x_j\})$. Then for all $\mathbf{v} \in S$,

$$d_E(\mathbf{v}, H_{j+1}) \leq d_E(\mathbf{v}, H_j) + d_E(\mathbf{v}_j, H_j) \leq 2^{j+1} n\beta.$$

$A_{j+1} \subseteq H_{j+1}$ and $\dim(A_{j+1}) = 1 + \dim(A_j)$. The maximum value that j can take is

$$k = n - \dim(A_1) \leq \log(1/\alpha) + \log(n+2) \tag{3}$$

(the inequality follows from Equation (1)), at which point we obtain $A_k = H_k$ with $\dim(H_k) = n-1$. H_k satisfies

1. $H_k = \mathrm{Span}(H_k \cap S)$
2. $\dim(H_k) = n - 1$
3. $|H_k \cap S| \geq \alpha \cdot 2^n/(n+2)$
4. For all $\mathbf{v} \in S$, $d_E(\mathbf{v}, H_k) \leq 2^k n\beta \leq (1/\alpha)(n+2)n\beta$ using Equation (3).

Hence by Properties (1-3) above and Theorem 1, H_k takes the form

$$H_k = \{\mathbf{x} \ : \ \mathbf{w}_k.\mathbf{x} = t_k\}$$

where $\|\mathbf{w}_k\| = 1$ and entries of $\mathbf{w}_k$ and t_k are multiples of

$$E = \left(\sqrt{n}(1 + \lfloor \log(\frac{n+2}{\alpha})\rfloor)! n^{(1+\lfloor \log((n+2)/\alpha)\rfloor)}\right)^{-1}$$

(the expression from Theorem 1 with $\alpha/(n+2)$ plugged in for α).

$\mathbf{w}_k.\mathbf{v}$ is an integer multiple of E for all $\mathbf{v} \in V$. Hence if $t_k - E < \mathbf{w}_k.\mathbf{v} < t_k + E$ then $\mathbf{w}_k.\mathbf{v} = t_k$.

From Property 4 of H_k, for all $\mathbf{v} \in S$, $\mathbf{w}_k.\mathbf{v} = t_k$, provided we have

$$(1/\alpha)(n+2)n\beta < E.$$

Equivalently,

$$\beta < \left((1/\alpha)(n+2)n\sqrt{n}(1 + \lfloor \log(\frac{n+2}{\alpha})\rfloor)! n^{(1+\lfloor \log((n+2)/\alpha)\rfloor)}\right)^{-1}.$$

The expression for β given in the statement of this theorem satisfies the inequality. $\square$

Theorem 3 *Let F be a boolean perceptron and let G be a boolean function that disagrees with F on a fraction ϵ of the 2^n elements of V. Assume also that $|V_{01}| \geq \frac{1}{4}\epsilon \cdot 2^n$ and $|V_{10}| \geq \frac{1}{4}\epsilon \cdot 2^n$. Then the Euclidean distance between $\mu(V_{01})$ and $\mu(V_{10})$ is lower bounded by*

$$\Big((4/\epsilon) \cdot n^{(5+\lfloor \log(2n/\epsilon) \rfloor)} \cdot (2 + \lfloor \log(2n/\epsilon) \rfloor)!\Big)^{-4\log(1/\epsilon)}$$

which is $(\epsilon/n)^{O(\log(n/\epsilon)\log(1/\epsilon))}$.

Proof. If l is a line and S is a set of points, let $l(S)$ denote the set of points obtained by projecting elements of S onto their closest points on l.

Let H_F denote a hyperplane defining F and let l_1 be a line normal to H_F. We may assume H_F does not contains any elements of V. Observe that members of $l_1(V_{01})$ are separated from members of $l_1(V_{10})$ by the point of intersection of l_1 and H_F (which itself is $l_1(H_F)$). Let

$$\beta = \Big((4/\epsilon) \cdot n^{(5+\lfloor \log(2n/\epsilon) \rfloor)} \cdot (2 + \lfloor \log(2n/\epsilon) \rfloor)!\Big)^{-1} \tag{4}$$

(where we have plugged $\epsilon/2$ in for α into the expression for β in the statement of Theorem 2). Our analysis uses a sequence of $\lfloor \log(1/\epsilon) \rfloor$ cases.

Case 1. Suppose that at least a fraction $\beta^{4\log(1/\epsilon)-2}$ of elements of $V_{01} \cup V_{10}$ (i.e. at least $(\epsilon \cdot 2^n)\beta^{4\log(1/\epsilon)-2}$ vertices altogether) have projections onto l_1 that are more than β distant from $l_1(H_F)$. In this case we have

$$\|\mu(V_{01}) - \mu(V_{10})\| \geq \beta \cdot \beta^{4\log(1/\epsilon)-2}.$$

The alternative is that at least a fraction $(1 - \beta^{4\log(1/\epsilon)-2})$ of elements of $V_{01} \cup V_{10}$ (thus, at least $(\epsilon \cdot 2^n)(1 - \beta^{4\log(1/\epsilon)-2})$ points altogether) have projections onto l_1 that are less than β distant from $l_1(H_F)$.

In this case we apply Theorem 2 to obtain a hyperplane A_1 that contains all these points, that is, at least a fraction $1-\beta^{4\log(1/\epsilon)-2}$ of elements of $V_{01} \cup V_{10}$. (Theorem 2 applies since $\epsilon(1-\beta^{4\log(1/\epsilon)-2})$ plays the role of α, and $\epsilon(1 - \beta^{4\log(1/\epsilon)-2}) > \frac{1}{2}\epsilon$ (so with Equation (4) the corresponding β-value is sufficiently small).

Case 2. Let $A'_2 = H_F \cap A_1$; since H_F does not contain any elements of V, H_F does not contain A_1. $A'_2 \subset A_1$ separates $V_{01} \cap A_1$ from $V_{10} \cap A_1$. Let $l_2 \subseteq A_1$ be a line normal to A'_2.

Now suppose that at least a fraction $\beta^{4\log(1/\epsilon)-4}$ of elements of $V_{01} \cup V_{10}$ lie in A_1 and have projections onto l_2 that are more than β distant from $l_2(A'_2)$. Then

$$\|\mu(A_1 \cap V_{01}) - \mu(A_1 \cap V_{10})\| \geq \beta \cdot \beta^{4\log(1/\epsilon)-4}.$$

$|(V_{01} \setminus A_1)|/|V_{01}| \leq \epsilon\beta^{4\log(1/\epsilon)-2}/(\epsilon/4)$ and since all vertices lie within $\sqrt{n}$ of each other, the distance $\|\mu(V_{01}) - \mu(V_{01} \setminus A_1)\|$ is at most $(4\sqrt{n})\beta^{4\log(1/\epsilon)-2}$. A similar argument applies to V_{10}. Hence we have

$$\|\mu(V_{01}) - \mu(V_{10})\| \geq \beta \cdot \beta^{4\log(1/\epsilon)-4} - 2(4\sqrt{n})\beta^{4\log(1/\epsilon)-2} = \beta^{4\log(1/\epsilon)-4}(\beta - \beta^2 8\sqrt{n}) \geq \beta^{4\log(1/\epsilon)}.$$

It remains to cover the cases where a fraction less than $\beta^{4\log(1/\epsilon)-4}$ of the members of $V_{01} \cup V_{10}$ have projections onto l_2 that are more than β distant from $l_2(A'_2)$. Generally case j arises when a subspace A_{j-1} of dimension $n-(j-1)$ has been identified that contains at least a fraction $1-\sum_{\ell=1}^{j-1}\beta^{(4\log(1/\epsilon)-2\ell)}$ of the elements of $V_{01} \cup V_{10}$ (and we have not yet found a hyperplane separating enough of V_{01} from V_{10} with a sufficiently large margin).

Case j**.** Subspace A_{j-1} with $\dim(A_{j-1}) = n-(j-1)$ satisfies

$$\frac{|A_{j-1} \cap (V_{01} \cup V_{10})|}{|V_{01} \cup V_{10}|} \geq 1 - \sum_{\ell=1}^{j-1} \beta^{(4\log(1/\epsilon)-2\ell)}.$$

Let $A'_j = A_{j-1} \cap H_F$. $\dim(A'_j) = n-j$. Let $l_j \subseteq A_{j-1}$ be a line normal to A'_j.

Suppose that at least a fraction $\beta^{(4\log(1/\epsilon)-2j)}$ of elements of $V_{01} \cup V_{10}$ lie in A_{j-1} and have projections onto l_j that are more than β distant from $l_j(A'_j)$. Then

$$\|\mu(A_{j-1} \cap V_{01}) - \mu(A_{j-1} \cap V_{10})\| \geq \beta \cdot \beta^{(4\log(1/\epsilon)-2j)}.$$

Note that

$$\frac{|(V_{01} \cup V_{10}) \setminus A_{j-1}|}{|V_{01} \cup V_{10}|} \leq \sum_{\ell=1}^{j-1} \beta^{(4\log(1/\epsilon)-2\ell)}.$$

Since $\beta < \frac{1}{2}$ this fraction is less than $2\beta^{(4\log(1/\epsilon)-2(j-1))}$. Hence

$$\begin{aligned}\|\mu(V_{01}) - \mu(V_{10})\| &\geq \beta \cdot \beta^{(4\log(1/\epsilon)-2j)} - (4\sqrt{n})2\beta^{(4\log(1/\epsilon)-2(j-1))} \\ &= \beta^{(4\log(1/\epsilon)-2j)}(\beta - 2\beta^2 4\sqrt{n}) \\ &\geq \beta^{4\log(1/\epsilon)}\end{aligned}$$

If alternatively a fraction at least $1-\beta^{(4\log(1/\epsilon)-2j)}$ of elements of $V_{01} \cup V_{10}$ have projections onto l_j at most β from $l_j(A'_j)$, then we construct A_j of dimension $n-j$ that contains all these points.

Let $V_j \subseteq (V_{01} \cup V_{10})$ denote this set of points. Let S_j be a set of $j-1$ vertices such that $\dim(\mathrm{Span}(A_{j-1} \cup S_j) = n$. The hyperplane $\mathrm{Span}(A'_j \cup S_j)$ lies within Euclidean distance β of elements of V_j, where $|V_j| \geq \frac{1}{2}\epsilon \cdot 2^n$. (For $j \leq \lfloor \log(\epsilon^{-1}) \rfloor$, the fraction of elements of $V_{01} \cup V_{10}$ that are in V_j is at least $1-\beta^{(4\log(1/\epsilon)-2j)}$, so that $|V_j| \geq \frac{1}{2}\epsilon$.) Use Theorem 2 (and Equation (4)) to obtain hyperplane H_j that contains $V_j \cup S_j$. Let $A_j = H_j \cap A_{j-1}$. H_j cannot contain A_{j-1} since H_j also contains S_j and we have $\mathrm{Span}(A_{j-1} \cup S_j) = n$. Hence $\dim(A_j) = n-j$.

For $j < \lfloor \log(\epsilon^{-1}) \rfloor$,

$$\frac{|A_j \cap (V_{01} \cup V_{10})|}{|V_{01} \cup V_{10}|} = \frac{|V_j|}{|V_{01} \cup V_{10}|} \geq 1 - \beta^{4\log(1/\epsilon)-2j} > 1 - \sum_{\ell=1}^{j} \beta^{4\log(1/\epsilon)-2\ell} > \frac{1}{2}\epsilon$$

so for $j < \lfloor \log(\epsilon^{-1}) \rfloor$ we are ready for case $j+1$.

By Lemma 1 the number of cases (and hence j) is indeed upper bounded by $\lfloor \log(\epsilon^{-1}) \rfloor$ since otherwise the subspace A_j does not have sufficient dimension to hold a fraction $\frac{1}{2}\epsilon$ of elements of V. Each of these cases provides a lower bound on $\|\mu(V_{01}) - \mu(V_{10})\|$ of $\beta^{4\log(1/\epsilon)}$, which is $\left((4/\epsilon) \cdot n^{(5+\lfloor \log(2n/\epsilon) \rfloor)} \cdot (2 + \lfloor \log(2n/\epsilon) \rfloor)!\right)^{-4\log(1/\epsilon)}$ as in the statement of the theorem. □

3 Statistical Learning-theoretic Consequences

For domain $V = \{0,1\}^n$ let $U(V)$ denote the uniform distribution on V. For a boolean function G having at least one satisfying assignment, let $Y_{G,0}$ be the following Bernoulli random variable: $Y_{G,0} = 1$ if for $\mathbf{v} \sim U(V)$ we have $G(\mathbf{v}) = 1$. Recall that $(\mathbf{v})_i$ denotes the 0/1 value of the i-th component of $\mathbf{v}$. For $1 \leq i \leq n$ let $Y_{G,i}$ be the following Bernoulli random variable: $Y_{G,i} = 1$ if for $\mathbf{v} \sim U(\{\mathbf{u} \in V \ : \ (\mathbf{u})_i = 1\})$ we have $G(\mathbf{v}) = 1$.

To learn a boolean perceptron in the 1-RFA régime (over the uniform distribution on $V = \{0,1\}^n$), a "target perceptron" F is selected by an adversary. A learning algorithm may (in unit time) generate an observation $(\mathbf{v}, \ell)$ where $\mathbf{v} \sim U(V)$ and $\ell = F(\mathbf{v})$. The algorithm has access to the value ℓ and may select $i \in \{1, \ldots, n\}$, so as to observe the value $(\mathbf{v})_i$. The remainder of $\mathbf{v}$ is not available to the algorithm. This is equivalent to being given access to repeated observations of the random variables $Y_{F,i}$ above, for $0 \leq i \leq n$. The objective is to output, with probability $1 - \delta$, a function G (the "hypothesis", an estimate of F) such that G disagrees with F on a fraction at most ϵ of elements of V. (An alternative formulation of RFA learning assumes that the indices of the observed components of an input vector $\mathbf{v}$ are selected uniformly at random. We noted in (13) that for 1-RFA learning this is equivalent, for the purpose of obtaining polynomial bounds, to the assumption that it is chosen at random.)

We continue by using the results of Section 2 to obtain a bound on the sample size required to learn a boolean perceptron in the 1-RFA setting. Thus we show how a computationally unbounded (but with limited sample size) algorithm can select a good hypothesis from the entire set of boolean perceptrons, using sample size $\log(\delta^{-1}) \cdot (n/\epsilon)^{\log(n/\epsilon)\log(1/\epsilon)}$, where δ is the probability that the hypothesis has error greater than ϵ. For any boolean function G let

$$\begin{aligned} p_{G,0} &= \Pr_{\mathbf{v} \sim U(V)}(G(\mathbf{v}) = 1) \\ p_{G,i} &= \Pr_{\mathbf{v} \sim U(V)}(G(\mathbf{v}) = 1 \mid (\mathbf{v})_i = 1) \end{aligned}$$

For a boolean function G define cost function $c_F(G)$ and empirical cost function $\hat{c}_F(G)$ as

$$\begin{aligned} c_F(G) &= \max_{0 \leq i \leq n}(|p_{G,i} - p_{F,i}|) \\ \hat{c}_F(G) &= \max_{0 \leq i \leq n}(|p_{G,i} - \hat{p}_{F,i}|) \end{aligned}$$

where $\hat{p}_{F,i}$ is defined in Figure 1. Note that $c_F(F) = 0$.

Lemma 2 *Let F be a boolean perceptron that is satisfied by at least $(\epsilon/2) \cdot 2^n$ input vectors. Let boolean function G disagree with F on at least a fraction ϵ of inputs. Then*

$$c_F(G) \geq \left(\frac{\epsilon^2}{32\sqrt{n}}\right)\left((4/\epsilon) \cdot n^{(5+\lfloor \log(2n/\epsilon) \rfloor)} \cdot (2 + \lfloor \log(2n/\epsilon) \rfloor)!\right)^{-4\log(1/\epsilon)}$$

Proof. We consider two cases. As in the proof of Theorem 3 let $\beta = ((4/\epsilon) \cdot n^{(5+\lfloor \log(2n/\epsilon) \rfloor)} \cdot (2 + \lfloor \log(2n/\epsilon) \rfloor)!)^{-1}$.
Case 1. $|p_{F,0} - p_{G,0}| \geq \frac{\epsilon^2}{32\sqrt{n}} \cdot \beta^{4\log(1/\epsilon)}$ (that is, there is a difference at least $\frac{\epsilon^2}{32\sqrt{n}} \cdot \beta^{4\log(1/\epsilon)}$ between the probability that $F(\mathbf{v}) = 1$ and the probability that $G(\mathbf{v}) = 1$). Then $c_F(G) \geq \frac{\epsilon^2}{32\sqrt{n}} \cdot \beta^{4\log(1/\epsilon)}$ which implies the statement of the Lemma.

Case 2. If $|p_{F,0} - p_{G,0}| < \frac{\epsilon^2}{32\sqrt{n}} \cdot \beta^{4\log(1/\epsilon)}$ then $|V_{01}| \geq (\epsilon/4) \cdot 2^n$ and $|V_{10}| \geq (\epsilon/4) \cdot 2^n$. So Theorem 3 applies to F and G, and we have

$$\|\mu(V_{01}) - \mu(V_{10})\| \geq \beta^{4\log(1/\epsilon)}$$

Let $\lambda = |V_{10}|/(|V_{10}| + |V_{11}|)$, $\lambda' = |V_{01}|/(|V_{01}| + |V_{11}|)$. If $|V_{10}| \geq |V_{01}|$ then $\lambda \geq \lambda'$ and

$$\lambda - \lambda' \leq \frac{|V_{10}| - |V_{01}|}{|V_{01}| + |V_{11}|} \leq \frac{|V_{10}| - |V_{01}|}{|V_{01}|} \leq \frac{(\epsilon^2/32\sqrt{n})\beta^{4\log(1/\epsilon)}}{\epsilon/4} = \frac{\epsilon}{8\sqrt{n}}\beta^{4\log(1/\epsilon)}.$$

If $|V_{01}| \geq |V_{10}|$ we have the same upper bound on $\lambda' - \lambda \geq 0$.

$$\begin{aligned} \mu(pos(F)) &= (1-\lambda) \cdot \mu(V_{11}) + \lambda\mu(V_{10}) \\ \mu(pos(G)) &= (1-\lambda') \cdot \mu(V_{11}) + \lambda'\mu(V_{01}) \\ &= (1-\lambda) \cdot \mu(V_{11}) + \lambda\mu(V_{01}) + (\lambda - \lambda')(\mu(V_{11}) - \mu(V_{01})). \end{aligned}$$

Hence (note that $\lambda \geq \frac{\epsilon}{4}$):

$$\begin{aligned} \|\mu(pos(F)) - \mu(pos(G))\| &\geq \lambda\|(\mu(V_{10}) - \mu(V_{01}))\| - (\lambda - \lambda')\|\mu(V_{11}) - \mu(V_{01})\| \\ &\geq \tfrac{\epsilon}{4}\|\mu(V_{10}) - \mu(V_{01})\| - (\lambda - \lambda')\sqrt{n} \\ &\geq \tfrac{\epsilon}{4}\beta^{4\log(1/\epsilon)} - \tfrac{\epsilon}{8}\beta^{4\log(1/\epsilon)}. \end{aligned}$$

The statement of the Lemma follows — there exists $i \in \{1, \ldots, n\}$ such that the i-th component of $\mu(pos(F))$ differs from the i-th component of $\mu(pos(G))$ by at least the above quantity divided by $\sqrt{n}$. □

Theorem 4 *Let F be an arbitrary boolean perceptron, and suppose that we have access to a source of observations of the form $((\mathbf{v})_i, F(\mathbf{v}))$, where $\mathbf{v} \sim U(V)$ where we may select the value of $i \in \{1, \ldots, n\}$ for each observation. Then (ignoring issues of computational efficiency) it is possible to find, with probability $1 - \delta$, a boolean function G such that $\Pr_{\mathbf{v}\sim U(V)}(F(\mathbf{v}) \neq G(\mathbf{v})) \leq \epsilon$, and the number of observations required is*

$$\log(1/\delta) \cdot (n/\epsilon)^{O(\log(n/\epsilon)\log(1/\epsilon))}.$$

Proof. We use the procedure illustrated in Figure 1. Note that symbols denoting various quantities are introduced in Figure 1.

Choose $N = |S_0|$ is to ensure that with probability $1 - \frac{1}{2}\delta$, if $\hat{p}_{F,0} < \frac{3}{4}\epsilon$ then $p_{F,0} \leq \epsilon$. As a result, the function G output in line 3 that has no satisfying assignments, has error at most ϵ. We show as follows that $N = O((1/\epsilon)\log(1/\delta))$ is large enough.

Recall Hoeffding's inequality: Let $Y_1, \ldots, Y_N$ be Bernoulli trials with probability p of success. Let $T = Y_1 + \ldots + Y_N$ denote the total number of successes. Then for $\gamma \in [0, 1]$,

$$\Pr(|T - pN| > \gamma N) \leq 2e^{-2N\gamma^2}.$$

1. *Draw a sample* S_0 *of observations, where* $|S_0| = \Theta((1/\epsilon)\log(1/\delta))$.
2. *Let* $\hat{p}_{F,0}$ *be the fraction of examples in* S_0 *which satisfy* F *(we do not look at any component of the input vectors).*
3. *If* $\hat{p}_{F,0} < \frac{3}{4}\epsilon$ *then output* G *where* $G(\mathbf{v}) = 0$ *for all* $\mathbf{v} \in \{0,1\}^n$.
4. *Else*
   (a) *For* $1 \le i \le n$, *Draw a sample* S_i *of observations, where* $|S_i| = (\log(1/\delta)(n/\epsilon)^{O(\log(n/\epsilon)\log(1/\epsilon))})$. *Look at the* i*-th component of each input* $\mathbf{v}$ *in* S_i.
   (b) *For* $0 \le i \le n$ *let* $\hat{p}_{F,i}$ *be the fraction of all examples with* $(\mathbf{v})_i = 1$ *in* S_i, *that are positive (satisfy* F*).*
   (c) *For every satisfiable boolean function* G *let* $p_{G,i} = \Pr(Y_{G,i} = 1)$ *(for* $0 \le i \le n$*).*
   (d) *Let* $\hat{c}(G) = \max_{0\le i\le n}(|\hat{p}_{F,i} - p_{G,i}|)$.
   (e) *Output a boolean function from* $\arg\min_G(\hat{c}(G))$.

Figure 1: Rule for selecting low-error perceptron

Put $\gamma = \frac{1}{4}\epsilon$ to ensure that with high probability

$$|\hat{p}_{F,0} - p_{F,0}| < \frac{1}{4}\epsilon. \tag{5}$$

$N = |S_0|$ must then satisfy $2e^{-2N(\epsilon/4)^2} \le \frac{1}{2}\delta$, which is satisfied by $N = O(\epsilon^{-1}\log(\delta^{-1}))$.

Equation 5 ensures that if $\hat{p}_{F,0} \ge \frac{3}{4}\epsilon$ then $p_{F,0} \ge \frac{1}{2}\epsilon$. So line 4 is (with probability $1 - \frac{1}{2}\delta$) used only when $p_{F,0} \ge \frac{1}{2}\epsilon$ (and Lemma 2 is applicable). As in the proof of Theorem 3 and Lemma 2 let $\beta = ((4/\epsilon)\cdot n^{(5+\lfloor\log(2n/\epsilon)\rfloor)}\cdot(2+\lfloor\log(2n/\epsilon)\rfloor)!)^{-1}$.

We choose the size of each S_i large enough to ensure that with probability $1 - \delta/4$ each S_i contains at least N' examples $(\mathbf{v}, F(\mathbf{v}))$ with $(\mathbf{v})_i = 1$, where N' is large enough to ensure that

$$\text{with probability } 1-\delta/4, \text{ for } 1 \le i \le n,\ |\hat{p}_{F,i} - p_{F,i}| < \Big(\frac{\epsilon^2}{64\sqrt{n}}\Big)\beta^{4\log(1/\epsilon)}. \tag{6}$$

The above can be ensured by taking a union bound if we have

$$\text{for } 1 \le i \le n, \text{ with probability } 1-\delta/4n,\ |\hat{p}_{F,i} - p_{F,i}| < \Big(\frac{\epsilon^2}{64\sqrt{n}}\Big)\beta^{4\log(1/\epsilon)}.$$

By Hoeffding's inequality it is sufficient for N' to satisfy $2\exp(-2N'(\epsilon^2/64\sqrt{n})\beta^{4\log(1/\epsilon)}) < \delta/4n$, which is satisfied by $N' = O((n/\epsilon^2)\log(n/\delta)/\beta^{4\log(1/\epsilon)})$.

Put $|S_i| = 4N'$. A standard Chernoff bound (see for example (1), p. 361) tells us that if T is the number of successes in N Bernoulli trials with probability p of success,

$$\Pr(T < \frac{1}{2}Np) \leq \exp(-Np/8).$$

$|S_i| = 4N'$, so the expected number of examples with $(\mathbf{v})_i = 1$ is $2N'$ (since $\Pr((\mathbf{v})_i = 1) = \frac{1}{2}$), and the probability that we fail to obtain N' of these examples is $O(\exp(-N'(\epsilon/2)/8)) = O(\delta/n)$. For $N' = O((n/\epsilon^2)\log(n/\delta)/\beta^{4\log(1/\epsilon)})$ this failure probability can be made as low as $\delta/4n$, so that with probability at least $1 - \frac{1}{4}\delta$, for $1 \leq i \leq n$, S_i contains at least N' examples with $(\mathbf{v})_i = 1$.

Equation 6 implies

$$\text{with probability } 1 - \delta/4, \text{ for all } G, \ |\hat{c}_F(G) - c_F(G)| < \Big(\frac{\epsilon^2}{64\sqrt{n}}\Big)\beta^{4\log(1/\epsilon)}.$$

Then by Lemma 2 (and noting that $c_F(F) = 0$), $\hat{c}_F(F) < \hat{c}_F(G)$ for all boolean functions G that disagree with F on a fraction at least ϵ of inputs.

The total sample size is $O(n \cdot N')$ which is $O((n^2/\epsilon^2)\log(n/\delta)/\beta)$, which is $\log(1/\delta) \cdot (n/\epsilon)^{O(\log(n/\epsilon)\log(1/\epsilon))}$. □

3.1 Conclusions and Open Problems

The problem of PAC-learning a boolean perceptron from empirical estimates of its Chow parameters has been raised in various papers in computational learning theory. We have so far just shown a bound on the asymptotic growth rate of sample-size required (the problem of how to best select the right hypothesis given sufficient data having not been addressed), and that bound is still super-polynomial. We suspect that the true growth rate is polynomially bounded, as a function of n/ϵ.

Our results show that an algorithm can minimise over the set of all boolean functions; we do not have to restrict to boolean perceptrons. This shows how the usage of a set of statistics, as opposed to empirical risk minimisation, can automatically avoid over-fitting. But there is the possibility that there should exist a better bound on the distance between the average satisfying assignment of two functions if both, and not just one, of them are perceptrons.

There may be a practical advantage to minimising over all boolean functions, in that if the minimisation is being done by local search, it may reduce problems with local optima. However, in principle one can just minimise over the set of all boolean perceptrons. The algorithm uses the values $p_{G,i}$ for boolean functions G, and for boolean perceptrons, computing these quantities exactly is $\sharp P$-hard since it is the 0/1 knapsack problem (11). However, sufficiently good approximations to these quantities could be found by generating a polynomial-size collection of inputs from $U(V)$ and using the empirical values.

Håstad (15) has shown that some boolean perceptrons need weights of size around $2^{(n\log n)/2-n}$ to be represented exactly. For $n = \lfloor\log(\epsilon^{-1})\rfloor$ (n being the dimension of the domain), an approximation with error less than ϵ must be exact. This implies we may need to learn a weight of size more than polynomial in ϵ, in order to recover a weights-based parametrization — weights may be as high as $(1/\epsilon)^{\log\log(1/\epsilon)}$. This eliminates one natural-looking way of obtaining the desired polynomial growth rate in ϵ^{-1} (namely, look for a perceptron whose coefficients are polynomially bounded as a function of the dimension and the quality of the approximation).

4 Acknowledgements

I would like to thank the referees for their corrections and comments.

References

[1] M. Anthony and P.L. Bartlett. (1999). *Neural Network Learning: Theoretical Foundations.* Cambridge University Press, Cambridge.

[2] M. Anthony, G. Brightwell and J. Shawe-Taylor. (1995). On Specifying Boolean Functions by Labelled Examples. *Discrete Applied Mathematics*, 61, pp. 1-25.

[3] S. Ben-David and E. Dichterman. (1998). Learning with Restricted Focus of Attention. *Journal of Computer and System Sciences*, 56(3), pp. 277-298.

[4] S. Ben-David and E. Dichterman. (1994). Learnability with restricted focus of attention guarantees noise-tolerance. *5th International Workshop on Algorithmic Learning Theory*, pp. 248-259.

[5] A. Birkendorf, E. Dichterman, J. Jackson, N. Klasner and H.U. Simon. (1998). On restricted-focus-of-attention learnability of Boolean functions. *Machine Learning*, 30, pp. 89-123.

[6] A. Blum, A. Frieze, R. Kannan and S. Vempala (1998). A Polynomial-time Algorithm for Learning Noisy Linear Threshold Functions. *Algorithmica*, 22, pp. 35-52.

[7] A. Blumer, A. Ehrenfeucht, D. Haussler and M.K. Warmuth. (1989). Learnability and the Vapnik-Chervonenkis Dimension. *Journal of the ACM*, 36, pp. 929-965.

[8] J. Bruck. (1990). Harmonic analysis of polynomial threshold functions. *SIAM Journal of Discrete Mathematics*, 3(2), pp. 168-177.

[9] C.K. Chow. (1961). On the characterization of threshold functions. *Proc. Symp. on Switching Circuit Theory and Logical Design*, pp. 34-38.

[10] E. Dichterman. (1998). Learning with Limited Visibility. *CDAM Research Reports Series, LSE-CDAM-98-01*, 44pp.

[11] M. E. Dyer, A. M. Frieze, R. Kannan, A. Kapoor, L. Perkovic and U. Vazirani. (1993). A mildly exponential time algorithm for approximating the number of solutions to a multidimensional knapsack problem. *Combinatorics, Probability and Computing*, 2, pp. 271-284.

[12] T. Eiter, T. Ibaraki and K. Makino. (1998). Decision Lists and Related Boolean Functions. *Institut Für Informatik JLU Giessen (IFIG) Research Reports 9804.*

[13] P.W. Goldberg. (2001). Learning Fixed-dimension Linear Thresholds from Fragmented Data. *Information and Computation*, 171, pp. 98-122.

[14] J. Hadamard. (1893). Résolution d'une question relative aux déterminants, *Bull. Sci. Math.*, 2, pp. 240-246.

[15] J. Håstad. (1994). On the size of weights for threshold gates. *SIAM Journal on Discrete Math*, 7(3), pp. 484-92.

[16] P. Kaszerman. (1963). A geometric test-synthesis procedure for a threshold device. *Information and Control*, 6(4), pp. 381-398.

[17] N. Littlestone. (1988). Learning Quickly When Irrelevant Attributes Abound: A New Linear-threshold Algorithm. *Machine Learning*, 2, pp. 285-318.

[18] R. L. Rivest. (1996). Learning Decision Lists. *Machine Learning*, 2, pp. 229-246.

[19] F. Rosenblatt. (1962). *Principles of Neurodynamics*. Spartan Books, New York, 1962.

[20] R.O. Winder. (1969). Threshold Gate Approximations Based on Chow Parameters. *IEEE Transactions on Computers*, 18, pp. 372-5.

[21] R.O. Winder. (1971). Chow Parameters in Threshold Logic. *Journal of the ACM*, 18(2), pp. 265-89.