On the Average Tractability of Binary Integer Programming and the Curious Transition to Perfect Generalization in Learning Majority Functions

Shao C. Fang and Santosh S. Venkatesh
Department of Electrical Engineering
University of Pennsylvania
Philadelphia, PA 19104
fang@ee.upenn.edu and venkatesh@ee.upenn.edu

Abstract

Learning binary weights for Majority functions is equivalent to binary integer programming and is hence NP-complete in the strong sense. Nonetheless, in this communication, a very simple learning algorithm, dubbed Majority Rule, is shown to have very curious properties which relate to the average case tractability of the problem. The algorithm has linear time complexity and is hence very efficient. Among its curious features is a very strange transition to perfect generalization as a function of sample size. For very small sample sizes $O(n/\log n)$, where n is the dimensionality, the algorithm is consistent on the examples; for sample sizes in the range $\Omega(n/\log n)$ to $O(n \log n)$, however, the algorithm fails to even load the set of examples, i.e., it is *inconsistent*; when sample sizes are in excess of $\Omega(n \log n)$, however, the algorithm abruptly loads the examples again, and as this is slightly in excess of the VC-dimension ($\Theta(n)$) of the problem, this implies generalization has been achieved in learning an unknown Majority function from examples. We dub this idiosyncratic behavior ***asymptotic consistency***. A consequence is that almost all instances of binary integer programming with $\Omega(n \log n)$ (or $O(n/\log n)$) inequalities to be satisfied are tractable.

ACM COLT '93 /7/93/CA, USA
© 1993 ACM 0-89791-611-5/93/0007/0310...$1.50

1 INTRODUCTION

1.1 Majority Functions and Binary Integer Programming

Binary integer programming is known to be NP-complete in the strong sense (cf. Garey and Johnson [3, p. 245]) and this result relates to the difficulty of learning Majority functions (or weights for a binary Peceptron). For simplicity, write $\mathbb{B} = \{-1, 1\}$. Now consider the following learning problem: we are given a binary Perceptron or McCulloch-Pitts neuron characterized by a vector of binary weights $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{B}^n$. The inputs to the neuron are literals $\mathbf{u} = (u_1, \ldots, u_n) \in \mathbb{B}^n$. The output of the binary neuron is the Majority function

$$h_{\mathbf{w}}(\mathbf{u}) = \operatorname{sgn} \langle \mathbf{w}, \mathbf{u} \rangle = \operatorname{sgn} \sum_{i=1}^{n} w_i u_i.$$

The functions to be learned are exactly the set of functions $\{h_{\mathbf{w}} : \mathbf{w} \in \mathbb{B}^n\}$, i.e., Majority functions of a set of literals. Equivalently, each binary weight vector $\mathbf{w} \in \mathbb{B}^n$ determines a positive half-space of vertices

$$\mathbb{B}^n_+(\mathbf{w}) = \{\mathbf{u} \in \mathbb{B}^n : \langle \mathbf{w}, \mathbf{u} \rangle \geq 0\}.$$

The hypothesis space is hence the set of 2^n positive half-spaces $\{\mathbb{B}^n_+(\mathbf{w})\}$ corresponding to vertices $\mathbf{w} \in \mathbb{B}^n$. Given an arbitrary majority function, the goal is to learn the corresponding vector of binary weights $\mathbf{w}^s \in \mathbb{B}^n$—the *solution vector*—from examples of the function drawn at random from the vertices $\mathbb{B}^n$ of the cube. In particular, we are given a set of m labeled examples $\{(\mathbf{u}^1, l(\mathbf{u}^1)), \ldots, (\mathbf{u}^m, l(\mathbf{u}^m))\}$ labeled according to the positive half-space of the unknown solution vector $\mathbf{w}^s \in \mathbb{B}^n$: for $\alpha = 1, \ldots, m$,

$$l(\mathbf{u}^\alpha) = \begin{cases} -1, & \text{if } \langle \mathbf{w}^s, \mathbf{u}^\alpha \rangle < 0, \\ +1, & \text{if } \langle \mathbf{w}^s, \mathbf{u}^\alpha \rangle \geq 0. \end{cases}$$

We will assume throughout that the examples are chosen independently from the uniform distribution on $\mathbb{B}^n$. In particular, this will imply that our results hold for almost all selections of examples from the vertices of the cube. An easy further consequence is that drawing labeled examples uniformly from $\mathbb{B}^n$ is equivalent to prescribing only positive examples drawn uniformly from the positive half-space $\mathbb{B}^n_+$ of the solution vector $\mathbf{w}^s \in \mathbb{B}^n$.[1] Thus, we consider a random m-set of examples $U = \{\mathbf{u}^1, \ldots, \mathbf{u}^m\}$ drawn independently from the uniform distribution of points in some fixed positive half-space $\mathbb{B}^n_+$. Our goal is to find a binary weight vector $\mathbf{w} \in \mathbb{B}^n$ such that the inner product $\langle \mathbf{w}, \mathbf{u}^\alpha \rangle$ is positive:

$$\sum_{i=1}^{n} w_i u_i^\alpha \geq 0, \qquad \alpha = 1, \ldots, m. \tag{1}$$

If the number m of examples is large enough, with high probability there is a *unique* solution to (1) given by the underlying solution vector $\mathbf{w}^s$ according to which the examples are drawn. Thus, loading the examples would be equivalent to learning the underlying Majority function *exactly*; or, to put it in another way, with a sufficient number of examples *perfect* generalization can be achieved. How many examples are required? Baum and Lyuu [1] show that only about $2n$ examples are needed: in particular, if at least $2n$ examples are drawn from the uniform distribution on $\mathbb{B}^n$ and classified according to a linearly separable target function $h_{\mathbf{w}^s}$, the probability that there exists any other target function consistent with the examples is exponentially small. Thus, an exponential search through the vertices of the cube to find a weight vector $\mathbf{w}$ consistent on the examples will result in identification of the target Majority function with high confidence if m exceeds $2n$.

On the face of it, it does not appear likely that a polynomial time algorithm can achieve learning of Majority functions with such a low sample complexity. Consider each of the m input patterns $\mathbf{u}^\alpha = (u_1^\alpha, \ldots, u_n^\alpha) \in \mathbb{B}^n$ as a n-dimensional column vector and construct a $n \times m$ matrix $\mathbf{M}$ out of the m-set of training patterns U. Then the problem of finding a binary solution weight vector $\mathbf{w}$ to map all input patterns to $+1$ is equivalent to solving the system of inequalities

$$\mathbf{M}^t \mathbf{w} \geq \mathbf{0}$$

with the components of the n-dimensional weight vector $\mathbf{w}$ constrained to $\{-1, 1\}$. This is a binary integer programming problem, known to be NP-complete in the strong sense [3, p. 245]. So we might anticipate that there is an intractable worst case for any (deterministic) algorithm. In this communication, however, we describe the very simple Majority Rule algorithm which learns Majority functions with high confidence while achieving the desideratum of low time and sample complexities.

The algorithm has several curious features of note:

- It has linear time complexity so that it produces putative solutions very rapidly.
- The algorithm is *not* consistent on the examples, and as the sample complexity m varies it exhibits three distinct domains of behavior. When m is less than $O(n/\log n)$ the algorithm is consistent on the examples; when m lies in a range $\Omega(n/\log n)$ and $O(n \log n)$ the algorithm is inconsistent on the examples (with probability one!); finally, when m exceeds $\Omega(n \log n)$ the algorithm, in very fickle fashion, abruptly becomes consistent on the examples (again with probability one). We call this idiosyncratic behavior ***asymptotic consistency***.[2] Because the examples are drawn uniformly from $\mathbb{B}^n$, this implies that almost all instances of the binary integer programming problem are tractable when m is less than $O(n/\log n)$ or when m exceeds $\Omega(n \log n)$.
- The algorithm exactly learns the underlying Majority function (i.e., the binary solution weight vector $\mathbf{w}^s \in \mathbb{B}^n$) with high confidence when the sample complexity m exceeds $\pi n \log n$. Thus the algorithm needs sample complexities for perfect learning only slightly in excess (within a logarithmic factor) of the minimum needed to identify the function.

In the sequel, we describe the algorithm 'and sketch proofs of the main results.

1.2 The Majority Rule Algorithm

Let $U = \{\mathbf{u}^1, \ldots, \mathbf{u}^m\}$ be some m-set of patterns in a positive half-space $\mathbb{B}^n_+$. The Majority Rule algorithm (cf. Venkatesh [4]) prescribes weights as follows:

For $i = 1, \ldots, n$, let

$$U_i^+ = \{\mathbf{u}^\alpha \in U : u_i^\alpha = +1\},$$
$$U_i^- = \{\mathbf{u}^\alpha \in U : u_i^\alpha = -1\}.$$

Set

$$w_i^{MR} = \begin{cases} +1, & \text{if } |U_i^+| \geq |U_i^-|, \\ -1, & \text{if } |U_i^+| < |U_i^-|. \end{cases}$$

In other words, $w_i^{MR} = +1$ if the patterns whose ith component is $+1$ are in the majority, and $w_i^{MR} = -1$ otherwise. The motivation is readily seen from (1): each summand $w_i^{MR} u_i^\alpha$ is more likely to be positive so that the whole sum $\sum_i w_i^{MR} u_i^\alpha$ is also more likely to be positive. Note that we can also write

$$\mathbf{w}^{MR} = \operatorname{sgn} \sum_{\alpha=1}^{m} \mathbf{u}^\alpha,$$

[1] To be precise, this statement holds as stated only for *odd* n. For *even* n there is a probability $O(n^{-1/2})$ of an example landing on the hyperplane corresponding to a solution vector. Such boundary effects will be negligible for large n.

[2] While learning theory predicts that any consistent algorithm will generalize given a set of examples in excess of the VC-dimension, practical time constraints on the algorithms force many heuristics in use to be inconsistent. Examples of inconsistent, nonetheless popular, learning algorithms abound, and include such generic classes as gradient descent algorithms. For the present problem too, the NP-completeness of binary integer programming makes it unlikely that there is any polynomial time consistent algorithm which is guaranteed to work on all instances of the problem.

where the sign operation on a vector is interpreted in natural fashion as the vector obtained by taking the sign of each of the components. Note that Majority Rule is a *local* algorithm; specifically, the ith component of $\mathbf{w}^{MR}$ depends solely on the ith components of the input patterns $\{\mathbf{u}^\alpha\}_{\alpha=1}^m$. The algorithm is also *homogeneous*; that is, the same procedure is employed to determine every component of the weight vector. Locality and homogeneity are clearly desirable features contributing to a low complexity of specification. The algorithm requires $n(m-1)$ additions, and n comparisons, so that it has time complexity linear in nm, the number of bits needed to specify the examples. The space complexity is also small being linear in nm.

Our basic results are encapsulated in the following theorem.[3] We assume that, for each n, the m-set of examples U is drawn randomly and independently from the uniform distribution on some fixed positive half-space $\mathbb{B}_+^n$ corresponding to a solution weight vector $\mathbf{w}^s \in \mathbb{B}^n$. We write $m = m_n$ to explicitly indicate that the number of examples drawn is a function of dimension n. Let $\mathcal{E}(n, m_n)$ denote the event that $\sum_{i=1}^n w_i^{MR} u_i^\alpha > 0$ for all $1 \leq \alpha \leq m$, i.e., $\mathcal{E}(n, m_n)$ is the event that the Majority Rule algorithm is consistent on the set of randomly drawn examples.

Theorem 1 *For any fixed* $0 < \lambda < 1$ *(chosen arbitrarily small), as* $n \to \infty$*:*

(a) If the sequence of sample complexities $\{m_n\}$ *satisfies*

$$m_n \leq (1-\lambda)\frac{n}{\pi \log n},$$

then $\mathbf{P}(\mathcal{E}(n, m_n)) \to 1$.

(b) If the sequence of sample complexities $\{m_n\}$ *satisfies*

$$(1+\lambda)\frac{n}{\pi \log n} \leq m_n \leq (1-\lambda)\pi n \log n,$$

then $\mathbf{P}(\mathcal{E}(n, m_n)) \to 0$.

(c) If the sequence of sample complexities $\{m_n\}$ *satisfies*

$$m_n \geq (1+\lambda)\pi n \log n,$$

then $\mathbf{P}(\mathcal{E}(n, m_n)) \to 1$.

In particular, there is a sharp threshold beha. at a sample complexity of $n/\pi \log n$ (the *lower threshold*), and again at $\pi n \log n$ (the *upper threshold*). In the "small-sample-complexity" region $n/\pi \log n < m < \pi n \log n$, there are a plethora of possible solutions to the set of inequalities (1) and the algorithm successfully finds one. Note that there is no generalization in this regime as the sample complexity is still below the information theoretic minimum (of the order of $2n$) needed to identify the solution vector uniquely. In the "moderate-sample-complexity" region $m > n/\pi \log n$, however, the number of possible solutions dwindles sharply and the feeble minded algorithm fails to find any solution even though at least one exists. Clearly, the algorithm does not generalize in this regime.

Having failed to generate a solution when sample complexities exceed $n/\pi \log n$, one would perhaps anticipate that increasing m any further would only worsen the situation as the constraint can only increase with increased sample complexity. However, as we see in part (c) of the theorem, most surprisingly, the algorithm vindicates itself triumphantly when m exceeds $\pi n \log n$, i.e., when m is only slightly more than linear in n. As this exceeds $2n$, it follows that the algorithm in this domain has generalized effectively and perfectly, with probability one.

Part (c) of the theorem motivates us to define the notion of *asymptotic consistency*: we say that the Majority Rule algorithm is *asymptotically consistent with learning sample complexity* $\pi n \log n$.

In the following we sketch a proof of the main result.

2 PROOF SKETCH

2.1 Dependencies

The basic components of the analysis involve the use of a probabilistic sieve coupled with careful asymptotic calculations. The major complicating factor is that statistical dependencies, albeit somewhat weak, abound in the problem, and a considerable portion of the effort goes into quelling these dependencies with a firm hand.

If components of the examples are chosen randomly from a sequence of symmetric Bernoulli trials without regard to linear separability, w_i^{MR} and w_j^{MR}, $i \neq j$, are *independent* of each other as seen from the algorithm description. The training sample set is constituted in this fashion when we investigate an algorithm's (random) capacity [4]. However, when $\{\mathbf{u}^\alpha\}_{\alpha=1}^m$ is restricted to be a linearly separable set, different components of $\mathbf{w}^{MR}$ are no longer independent; i.e., if $\mathbf{w}^s$ denotes the solution weight vector which classifies all input patterns to $+1$, the events $\{w_i^{MR} = w_i^s\}$ and $\{w_j^{MR} = w_j^s\}$, $i \neq j$, are *dependent*. The dependence across components complicates analysis of the algorithm. However, we will see that the dependence is *weak*. In fact, the components are independent *asymptotically*.

Assign equal probability to points in $\mathbb{B}_+^n$ corresponding to $\mathbf{w}^s$ and randomly pick a point $\mathbf{u}$ from them. Let the random variable X denote the number of components $\mathbf{u}$ and $\mathbf{w}^s$ have in common; i.e.,

$$X = \left|\{i | u_i = w_i^s, 1 \leq i \leq n\}\right|.$$

Therefore X takes value ν, $\frac{n}{2} \leq \nu \leq n$, with probability

[3] All logarithms are to base e.

$\frac{\binom{n}{\nu}}{2^{n-1}}$. Then

$$\begin{aligned}\mathbf{E}X &= \sum_{\nu=\frac{n}{2}}^{n} \nu \cdot \frac{\binom{n}{\nu}}{2^{n-1}} \\ &= \sum_{a=0}^{\frac{n}{2}} (\frac{n}{2}+a) \cdot \frac{\binom{n}{\frac{n}{2}+a}}{2^{n-1}} \\ &= \frac{n}{2} + \nu_o,\end{aligned}$$

where

$$\nu_o = \sum_{a=0}^{\frac{n}{2}} a \cdot \frac{\binom{n}{\frac{n}{2}+a}}{2^{n-1}}.$$

Without loss of generality, assume $n = 2k$.

Claim 1: The estimate

$$\mathbf{E}\,X = \frac{n}{2} + \nu_o = \frac{n}{2} + \Theta(\sqrt{n})$$

holds. In particular, as $n \to \infty$, $\nu_o \sim \sqrt{n}/\sqrt{2\pi}$.

The proof is long and replete with algebraic detail. We will be content to give a plausibility argument which the proof actually rigorizes. Note that $\mathbf{E}\,X$ would be exactly $n/2$ if $\mathbf{u}$ were to be chosen uniformly from the entire $\mathbb{B}^n$. Choosing $\mathbf{u}$ uniformly from the positive half-space $\mathbb{B}^n_+$, as expected, increases the expected number of components $\mathbf{u}$ has in common with $\mathbf{w}^s$. However, most choices of $\mathbf{u}$ will still be around the hyperplane orthogonal to $\mathbf{w}^s$ by the sphere hardening effect. The central term of the binomial hence yields the added factor $\Theta(\sqrt{n})$.

Notice that each of the n components of $\mathbf{u}$ has the probability $\frac{\mathbf{E}X}{n}$ of having the *same* sign as the same component of the solution vector $\mathbf{w}^s$, and, on the other hand, the two components have the probability $1 - \frac{\mathbf{E}X}{n}$ of having the *opposite* sign. Let

$$p \equiv \mathbf{P}\{u_i^\alpha = w_i^s\} = \frac{\mathbf{E}\,X}{n} = \frac{1}{2} + \frac{1}{\sqrt{2\pi n}} + o\left(\frac{1}{\sqrt{n}}\right),$$

$$q \equiv \mathbf{P}\{u_i^\alpha \neq w_i^s\} = 1 - p = \frac{1}{2} - \frac{1}{\sqrt{2\pi n}} + o\left(\frac{1}{\sqrt{n}}\right).$$

Since $\mathbf{u}^\alpha, \alpha = 1, \ldots, m$, are picked randomly and independently from $\mathbb{B}^n_+$, the application of the Majority Rule algorithm to $u_i^\alpha, \alpha = 1, \ldots, m$, is equivalent to a *Bernoullian random walk* of m steps with p and q as its probabilities of success (i.e., $\mathrm{S}_m = \mathrm{S}_{m-1} + 1$) and failure (i.e., $\mathrm{S}_m = \mathrm{S}_{m-1}$), respectively, where successive positions of the random walk are denoted by $\mathrm{S}_1, \mathrm{S}_2, \ldots$. Using the large deviation extension of the De Moivre-Laplace central limit theorem (cf. Feller [2]), we have[4]

$$\begin{aligned}\mathbf{P}\{w_i^{MR} \neq w_i^s\} &= \mathbf{P}\left\{\mathrm{S}_m < \frac{m}{2}\right\} \\ &= \mathbf{P}\left\{\frac{\mathrm{S}_m - mp}{\sqrt{mpq}} < \frac{\frac{m}{2} - mp}{\sqrt{mpq}}\right\} \\ &\sim \Phi\left(-\sqrt{\frac{m \cdot \frac{\nu_o^2}{n^2}}{\frac{1}{4} - \frac{\nu_o^2}{n^2}}}\right) \\ &\sim \frac{1}{\sqrt{\frac{m \cdot \frac{\nu_o^2}{n^2}}{\frac{1}{4} - \frac{\nu_o^2}{n^2}}}} \cdot \phi\left(\sqrt{\frac{m \cdot \frac{\nu_o^2}{n^2}}{\frac{1}{4} - \frac{\nu_o^2}{n^2}}}\right) \\ &\sim \frac{\sqrt{n}}{2\sqrt{m}} e^{-\frac{m}{\pi n}} \qquad (n \to \infty).\end{aligned}$$

The use of the large deviation theorem above requires $n \ll m \ll n^{\frac{4}{3}}$; e.g., $m = \pi n \log n$.

2.2 Asymptotic Consistency

We will sketch proofs of parts (b) and (c) of the theorem. The techniques needed to prove the rest of the theorem are similar.

Let Y denote the Hamming distance between $\mathbf{w}^{MR}$ and $\mathbf{w}^s$, i.e., $Y = |\{i : w_i^{MR} \neq w_i^s\}|$.

Claim 2: If $n \to \infty$ and $m \to \infty$ in such a way that $n = o(m)$ and $m = o(n^{\frac{4}{3}})$, then

$$\mathbf{E}\,Y \sim \frac{n^{\frac{3}{2}}}{2\sqrt{m}} e^{-\frac{m}{\pi n}}. \tag{2}$$

Proof: When $n \ll m \ll n^{\frac{4}{3}}$, we have

$$\mathbf{P}\{w_i^{MR} \neq w_i^s\} \sim \frac{\sqrt{n}}{2\sqrt{m}} e^{-\frac{m}{\pi n}}$$

for *any* $i = 1, \ldots, n$; i.e., the events $\{w_i^{MR} \neq w_i^s, 1 \leq i \leq n\}$ are *exchangeable*. Therefore

$$\begin{aligned}\mathbf{E}\,Y &= \sum_{i=1}^{n} \mathbf{P}\{w_i^{MR} \neq w_i^s\} = n \cdot \mathbf{P}\{w_i^{MR} \neq w_i^s\} \\ &\sim \frac{n^{\frac{3}{2}}}{2\sqrt{m}} e^{-\frac{m}{\pi n}}.\end{aligned}$$

The claim follows. ∎

Claim 3:

$$\mathbf{P}\{Y > 0\} \leq \mathbf{E}\,Y. \tag{3}$$

[4] As usual, $\phi(x) = e^{-x^2/2}/\sqrt{2\pi}$ is the standard Gaussian density, and $\Phi(x) = \int_{-\infty}^{x} \phi(y)\,dy$ is the Gaussian distribution function.

PROOF: Since Y takes values from $\{0,1,2,\dots\}$, we have

$$\mathbf{P}\{Y > 0\} = \mathbf{P}\{Y \geq 1\} \leq \sum_{i=1}^{\infty} \mathbf{P}\{Y \geq i\} = \mathbf{E}\,Y.$$

The claim is proved. ∎

With the help of the two results above, we can show

Claim 4: For any $0 < \lambda < 1$, if the sample complexity m_n increases with n such that $m_n \geq (1+\lambda)\pi n \log n$, then the probability that all the examples are correctly labeled by the weight vector generated by the algorithm tends to 1 as $n \to \infty$.

PROOF: If we fix any $\epsilon > 0$, and set $\frac{n^{\frac{3}{2}}}{2\sqrt{m}} e^{-\frac{m}{\pi n}} = \epsilon$. Then by successive approximation, we have

$$m = \pi n \log n \left(1 + \frac{\log\frac{1}{2\epsilon} - \frac{1}{2}\log \pi - O(\log\log n)}{\log n}\right).$$

In other words, $\mathbf{P}\{Y > 0\} \leq \mathbf{E}\,Y < \epsilon$ if we choose

$$m > \pi n \log n \left(1 + \frac{\log\frac{1}{2\epsilon} - \frac{1}{2}\log \pi - O(\log\log n)}{\log n}\right).$$

Since ϵ is arbitrary, it follows that, as $n \to \infty$, $\mathbf{P}\{\mathbf{w}^{MR} = \mathbf{w}^s\} \to 1$ if $m \geq (1+\lambda)\pi n \log n$ for any $\lambda > 0$. ∎

This establishes part (c) of the theorem. We now proceed to show that if $m < \pi n \log n$ then there is no generalization as the algorithm is inconsistent.

Claim 5: When $m = 1$, as $n \to \infty$, $\mathbf{P}\{w_i^{MR} = w_i^s$ in t specific components $\} = \frac{1}{2^t} + \frac{t}{2^{t-1}}\frac{1}{\sqrt{2\pi n}} + o(\frac{1}{\sqrt{n}})$.

The proof is by induction on t. We omit the messy details.

Now let $f(t,s)$ denote the probability of the event that, when $m = 1$, $w_i^{MR} = w_i^s$ in t specific components and $w_i^{MR} \neq w_i^s$ in s specific components.

Claim 6: As $n \to \infty$,

$$f(t,s) = \frac{1}{2^{t+s}} + \frac{t-s}{2^{t+s-1}}\frac{1}{\sqrt{2\pi n}} + o\left(\frac{1}{\sqrt{n}}\right). \quad (4)$$

PROOF: The proof is by double induction. Fix t and s with $(t,s) \neq (0,0)$.

BASE. We have $f(t,0) = \frac{1}{2^t} + \frac{t}{2^{t-1}}\frac{1}{\sqrt{2\pi n}} + o(\frac{1}{\sqrt{n}})$ for $t \geq 1$, and $f(0,1) = \frac{1}{2} - \frac{1}{\sqrt{2\pi n}} + o(\frac{1}{\sqrt{n}})$.

INDUCTION HYPOTHESIS. Assume the claim holds for $f(t,s)$ and $f(t+1,s)$, and $(t,s) \neq (0,0)$. Then

$$\begin{aligned} f(t,s+1) &= f(t,s) - f(t+1,s) \\ &= (\frac{1}{2^{t+s}} + \frac{t-s}{2^{t+s-1}}\frac{1}{\sqrt{2\pi n}}) \\ &- (\frac{1}{2^{t+s+1}} + \frac{t+1-s}{2^{t+s}}\frac{1}{\sqrt{2\pi n}}) + o\left(\frac{1}{\sqrt{n}}\right) \\ &= \frac{1}{2^{t+s+1}} + \frac{t-s-1}{2^{t+s}}\frac{1}{\sqrt{2\pi n}} + o\left(\frac{1}{\sqrt{n}}\right). \end{aligned}$$

Therefore the claim also holds for $f(t,s+1)$. So (4) holds for all fixed t and s as $n \to \infty$. ∎

Denote $p = \frac{1}{2} + \frac{1}{\sqrt{2\pi n}}$ and $q = \frac{1}{2} - \frac{1}{\sqrt{2\pi n}}$. Also let $\mathbf{P}_{(t,s)}$ denote the probability that $w_i^{MR} = w_i^s$ in t components and $w_i^{MR} \neq w_i^s$ in s components.

Claim 7: When $m = 1$, for fixed t and s, as $n \to \infty$,

$$\begin{aligned} \mathbf{P}_{(t,s)} &= \binom{t+s}{t} p^t q^s + o\left(\frac{1}{\sqrt{n}}\right) \\ &= \binom{t+s}{t}\left[\frac{1}{2^{t+s}} + \frac{t-s}{2^{t+s-1}}\frac{1}{\sqrt{2\pi n}}\right] + o\left(\frac{1}{\sqrt{n}}\right) \\ &= \binom{t+s}{t} f(t,s). \end{aligned}$$

PROOF: Note that

$$\begin{aligned} p^t q^s &= \left[\frac{1}{2} + \frac{1}{\sqrt{2\pi n}}\right]^t \left[\frac{1}{2} - \frac{1}{\sqrt{2\pi n}}\right]^s \\ &= \left[\frac{1}{2^t} + \frac{t}{2^{t-1}}\frac{1}{\sqrt{2\pi n}} + o\left(\frac{1}{\sqrt{n}}\right)\right] \\ &\cdot \left[\frac{1}{2^s} - \frac{s}{2^{s-1}}\frac{1}{\sqrt{2\pi n}} + o\left(\frac{1}{\sqrt{n}}\right)\right] \\ &= \frac{1}{2^{t+s}} + \frac{t-s}{2^{t+s-1}}\frac{1}{\sqrt{2\pi n}} + o\left(\frac{1}{\sqrt{n}}\right). \end{aligned}$$

The binomial coefficient factor follows from the fact that the matches and mismatches of the components of the Majority Rule weight vector to the corresponding solution vector components are *exchangeable* events. ∎

We have shown above that when $m = 1$, any t events from the set $\{w_i^{MR} \neq w_i^s\}_{i=1}^n$ are asymptotically independent as $n \to \infty$. When $m > 1$, since the samples are drawn independently of each other, the asymptotic independence across the components of $\mathbf{w}^{MR}$ is preserved. Therefore

Claim 8: For any $m \geq 1$, any t events from the set $\{w_i^{MR} \neq w_i^s,\ 1 \leq i \leq n\}$ are asymptotically independent as $n \to \infty$.

Let $\mathbf{P}_t = \mathbf{P}\{w_i^{MR} \neq w_i^s$ in t specific components$\}$. Then using the exchangeability of the events in $\{w_i^{MR} \neq$

$w_i^s,\ 1 \le i \le n\}$, we have

$$\begin{aligned}\mathbf{P}\{\mathbf{w}^{MR} \neq \mathbf{w}^s\} &= \mathbf{P}\left\{\bigcup_{i=1}^{n} w_i^{MR} \neq w_i^s\right\} \\ &= \sum_{t=1}^{n}(-1)^{t-1}\binom{n}{t}\mathbf{P}_t.\end{aligned}$$

Fix $T < n$. By the asymptotic independence of the T components, we have, as $n \to \infty$,

$$\begin{aligned}\sum_{t=1}^{T}(-1)^{t-1}\binom{n}{t}\mathbf{P}_t &\sim \sum_{t=1}^{T}(-1)^{t-1}\binom{n}{t}\mathbf{P}_1^t \\ &\sim \sum_{t=1}^{T}(-1)^{t-1}\frac{(n\mathbf{P}_1)^t}{t!} \\ &= -\sum_{t=1}^{T}\frac{(-n\mathbf{P}_1)^t}{t!}.\end{aligned}$$

Take T arbitrarily large. We thus have $\mathbf{P}\{\mathbf{w}^{MR} \neq \mathbf{w}^s\} \sim 1 - e^{-n\mathbf{P}_1}$, as $n \to \infty$. Set $e^{-n\mathbf{P}_1} = \epsilon$. Since $\mathbf{P}_1 \sim \frac{\sqrt{n}}{2\sqrt{m}}e^{-\frac{m}{\pi n}}$, using successive approximation, we obtain

$$m = \pi n \log n\left(1 - \frac{\log\sqrt{\pi} + \log(2\log\frac{1}{\epsilon}) + O(\log\log n)}{\log n}\right).$$

Since $\epsilon > 0$ is arbitrary, we have thus shown that as $n \to \infty$, $\mathbf{P}\{\mathbf{w}^{MR} = \mathbf{w}^s\} \to 0$ if $m \le (1-\lambda)\pi n \log n$ for any $\lambda > 0$. In other words, we have shown

Claim 9: For any $0 < \lambda < 1$, if the sample complexity m_n increases with n such that $m_n \le (1-\lambda)\pi n \log n$, then the probability that all the examples are correctly labeled by the weight vector generated by the algorithm tends to 0 as $n \to \infty$.

This completes the proof of one segment of part (b) of the theorem. The rest of the theorem can be proved along similar lines, with care being taken to suppress the dependencies. We conclude the sketch of the proof here.

3 SIMULATIONS

The 0/1 probability lower and upper thresholds are observed in computer simulations of the Majority Rule algorithm on linearly separable training sets for large n, confirming the previous analyses. The details of the simulations are as follows. For each *run*, one binary solution weight vector $\mathbf{w}^s$ is randomly picked from $\mathbb{B}^n$, and it defines $\mathbb{B}^n_+$. Then m points are independently and uniformly selected from $\mathbb{B}^n_+$, and Majority Rule is applied to the m-set $\{\mathbf{u}^\alpha\}_{\alpha=1}^m$ to obtain $\mathbf{w}^{MR}$. We now check to see if $\operatorname{sgn}\langle\mathbf{u}^\alpha, \mathbf{w}^{MR}\rangle = \operatorname{sgn}\langle\mathbf{u}^\alpha, \mathbf{w}^s\rangle$ for *all* $\alpha = 1, \ldots, m$. If so, a *success* is registered. For each pair of n and m values, 1000 runs are performed to obtain the success rate, which we call the *consistency probability*. In the lower threshold simulation, for each n value, we sweep m across a range of integers such that the ratio $\frac{m}{n/(\pi \log n)}$ ranges from 0 to about 15. The n values actually used range from 31 to 2047. Note in Figure 1 that as predicted, the consistency probability is near 1 for $\frac{m}{n/(\pi \log n)} < 1$ and decays abruptly to zero when $\frac{m}{n/(\pi \log n)} > 1$. To simulate the upper threshold, for each n value, we sweep m across a range of integers such that $\frac{m}{\pi n \log n}$ ranges from 0 to about 1.8. The n values actually used range from 127 to 1535. Note again in Figure 2 that as predicted, the consistency probability abruptly becomes 1 again (perfect generalization) when $\frac{m}{\pi n \log n} > 1$.

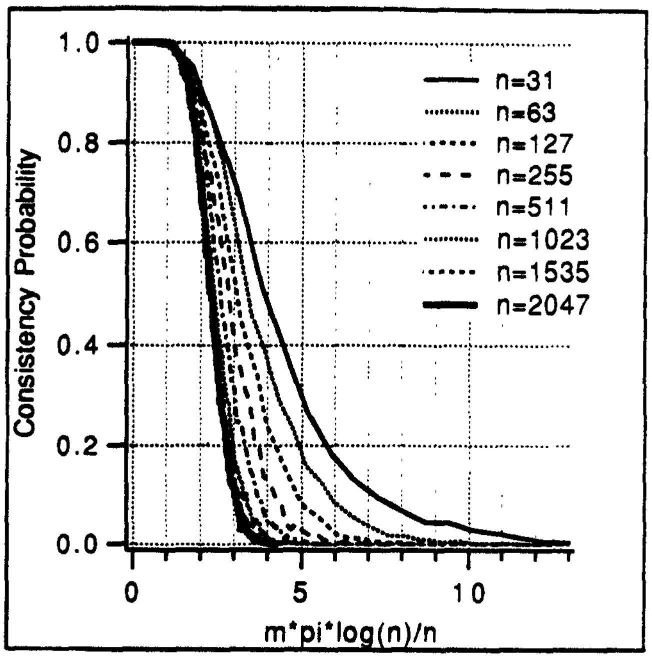

Figure 1: Actual Simulation of Lower Threshold

4 DISCUSSION

Learning binary weights for perceptrons is equivalent to binary integer programming, a known NP-complete problem; that is, it is unlikely that there exists an efficient (deterministic) algorithm which can *always* decide in polynomial time whether a given set of binary patterns in n-space is linearly separable by a hyperplane defined by a binary vector. In other words, we expect an intractable worst-case input for any (deterministic) algorithm targeted to answer the decision question. Our analysis here, however, shows a very optimistic average-case result — *almost all* instances of the problem are

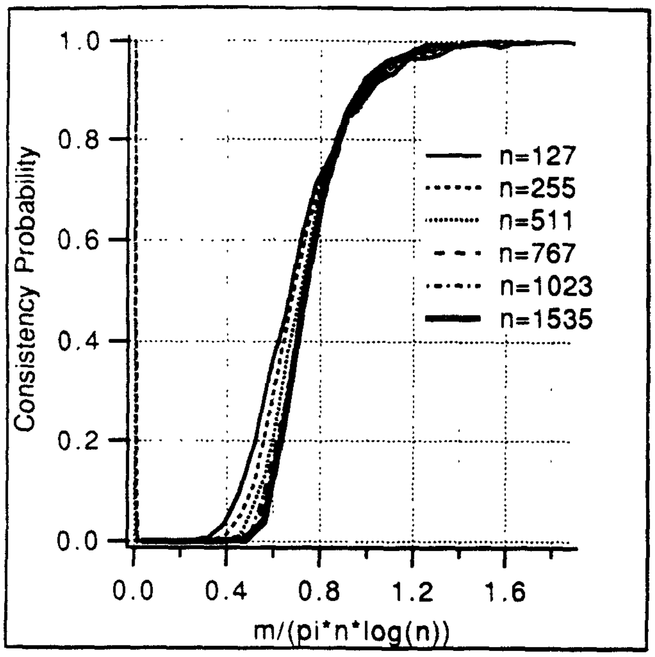

Figure 2: Actual Simulation of Upper Threshold

tractable.

More specifically, our setup imposes a probability distribution on the input space — for a given solution weight vector $\mathbf{w}^s$, we assign equal probability to all patterns located on the positive side of the hyperplane defined by $\mathbf{w}^s$.[5] Therefore, for any m, all m-set of patterns are equally probable and thus for a given number of patterns, all instances of the binary weight learning problem are uniformly distributed. Under this probabilistic framework, parts (a) and (c) of our result indicate that *almost all* instances of the problem with $O(n/\log n)$ or $\Omega(n \log n)$ inequalities to be satisfied are solvable by the Majority Rule algorithm.

Acknowledgement

The research support of the Air Force Office of Scientific Research is gratefully acknowledged.

References

[1] E. B. Baum and Yuh-Dauh Lyuu, "The Transition to Perfect Generalization in Perceptrons," *Neural Comp*, 3, 386-401, 1991.

[2] William Feller. *An Introduction to Probability Theory and Its Applications*, Volume I, Third Edition. Wiley, 1968.

[3] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: W. H. Freeman, 1979.

[4] S. S. Venkatesh, "On Learning Binary Weights for Majority Functions," in *Proc. of the Fourth Workshop on Comp. Learning Theory*, (eds. L. G. Valiant and M. K. Warmuth). San Mateo, California: Morgan Kaufmann, 1991.

[5] Equivalently, we could have assigned equal probability to *all* points in $\mathbb{B}^n$. Then the training set would be a m-set of pairs $\{(\mathbf{u}^\alpha, v^\alpha)\}_{\alpha=1}^{m}$, with $v^\alpha = +1$ for patterns lying on the positive side of the hyperplane and $v^\alpha = -1$ for the others. The patterns would need to be *normalized* before the application of Majority Rule.