Approximation by DNF: Examples and Counterexamples

Ryan O'Donnell
Carnegie Mellon University
odonnell@cs.cmu.edu

Karl Wimmer
Carnegie Mellon University
kwimmer@andrew.cmu.edu

May 1, 2007

Abstract

Say that $f : \{0,1\}^n \to \{0,1\}$ ϵ-approximates $g : \{0,1\}^n \to \{0,1\}$ if the functions disagree on at most an ϵ fraction of points. This paper contains two results about approximation by DNF and other small-depth circuits:

(1) For every constant $0 < \epsilon < 1/2$ there is a DNF of size $2^{O(\sqrt{n})}$ that ϵ-approximates the Majority function on n bits, and this is optimal up to the constant in the exponent.

(2) There is a monotone function $\mathcal{F} : \{0,1\}^n \to \{0,1\}$ with total influence (AKA average sensitivity) $\mathbb{I}(\mathcal{F}) \leq O(\log n)$ such that any DNF or CNF that .01-approximates $\mathcal{F}$ requires size $2^{\Omega(n/\log n)}$ and such that *any* unbounded fan-in AND-OR-NOT circuit that .01-approximates $\mathcal{F}$ requires size $\Omega(n/\log n)$. This disproves a conjecture of Benjamini, Kalai, and Schramm (appearing in [BKS99, Kal00, KS05]).

1 Introduction

1.1 Definitions

This paper is concerned with approximating boolean functions $f : \{0,1\}^n \to \{0,1\}$ by DNF formulas of small size. Let us first give the requisite definitions.

Circuits: We will consider single-output **circuits** composed of unbounded fan-in AND and OR gates over the input **literals** (inputs and negated inputs). The **size** of a circuit is the number of AND and OR gates it contains, and the **depth** of the circuit is the number of AND and OR gates on the longest path from an input bit to the output gate. We will also make the not completely standard definition that the **width** of a circuit is the maximum, over all AND and OR gates, of the number of *literals* feeding into the gate.

We will only be concerned with constant-depth circuits in this paper, and we will be particularly interested in depth 2. We assume circuits of depth 2 are always given by an OR of ANDs of literals, in which case they are **DNFs**, or by an AND of ORs of literals, in which case they are **CNFs**. The ORs in a DNF are called its **terms** and the ANDs in a CNF are called its **clauses**.

Finally, we will often identify a circuit over n input bits with the boolean function $\{0,1\}^n \to \{0,1\}$ that it computes.

Approximation: Given two functions $f, g : \{0,1\}^n \to \{0,1\}$, we will say that f **ϵ-approximates** g, or f is an **ϵ-approximator** for g, if the fraction of inputs in $\{0,1\}^n$ on which they disagree is at most ϵ. We will also write this as

$$\Pr_{\boldsymbol{x}}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})] \leq \epsilon,$$

with the convention that boldface letters are random variables, and that they are drawn from the uniform distribution on $\{0,1\}^n$ unless otherwise specified.

We will later need the following well known observation, showing that small-size circuits are well approximated by small-width circuits:

Observation 1.1 *If C is a circuit of size s, then for every $\epsilon > 0$ there is a "simplification" C' of C that ϵ-approximates C and has width at most $\log(s/\epsilon)$.*[1] *By "simplification" we mean that C' is obtained from C by replacing some of its gates with constants, so that C' has size and depth no more than C, and C' is a DNF (respectively, CNF) if C is.*

Proof: Consider any gate in C that is connected to at least $\log(s/\epsilon)$ input literals. If such a gate is an AND gate, replace it with a 0; and if it is an OR gate, replace it with a 1. This gives us C', which clearly has width at most $\log(s/\epsilon)$. Now on a uniformly random input, the probability that a particular replacement affects C's computation is at most $2^{-\log(s/\epsilon)} = \epsilon/s$. Since C has at most s gates, the probability that *any* replacement affects its computation is at most ϵ, by the union bound. Thus C' is an ϵ-approximator for C. □

[1] In this paper log denotes $\log_2$.

1.2 Approximation by DNF

DNF formulas are one of the simplest and most natural representation classes for boolean functions. Although every function can be computed by a DNF, some functions on n bits may require DNFs of size $\Omega(2^n)$. The natural question we pursue in this paper is whether this size can be significantly reduced for a given function if we are only required to ϵ-approximate it, for some small constant ϵ. Positive results along these lines would have interesting applications in several research areas, including computational learning theory and the the study of threshold phenomena in random graphs; these topics will be discussed in Sections 1.3 and 1.4, respectively. However there do not seem to be many results on either upper or lower bounds for approximation by DNF in the literature.

A notable conjecture in this area was made 8 years ago by Benjamini, Kalai, and Schramm [BKS99] (published again in [Kal00, KS05]). To describe this conjecture, which we call the BKS Conjecture, we need to recall the notion of **total influence** [KKL88, LMN93]:

Definition 1.2 *Given a function $f : \{0,1\}^n \to \{0,1\}$, the* influence of the ith coordinate *on f is*

$$\mathrm{Inf}_i(f) = \mathop{\mathbf{Pr}}_{\boldsymbol{x}}[f(\boldsymbol{x}) \neq f(\sigma_i \boldsymbol{x})],$$

where $\sigma_i \boldsymbol{x}$ denotes $\boldsymbol{x}$ with its ith bit flipped. The total influence *(or* average sensitivity*) of f is*

$$\mathbb{I}(f) = \sum_{i=1}^{n} \mathrm{Inf}_i(f) = \mathop{\mathbf{E}}_{\boldsymbol{x}}\left[\#\{y \sim \boldsymbol{x} : f(y) \neq f(\boldsymbol{x})\}\right],$$

where the notation $y \sim x$ means that the Hamming distance between y and x is 1.

The total influence is an important measure of the complexity of a function, used frequently in learning theory, threshold phenomena, and complexity theory. One important result to note is that constant-depth circuits of small size have small total influence:

Theorem 1.3 *Let $f : \{0,1\}^n \to \{0,1\}$ be computed by a circuit of depth d and size s. Then $\mathbb{I}(f) \leq O(\log^{d-1} s)$.*

This was first proved by Boppana [Bop97], tightening an argument made by Linial, Mansour, and Nisan [LMN93] based on Håstad's Switching Lemma [Hås86]. Note that the $d = 2$ case of this theorem is quite easy, building on the simple result that $\mathbb{I}(f) \leq 2w$ for any f computable by a DNF of width w.

We can now state Benjamini, Kalai, and Schramm's conjecture, which essentially asserts a converse to Theorem 1.3 for monotone functions:

BKS Conjecture: *For every $\epsilon > 0$ there is a constant $K = K(\epsilon) < \infty$ such that the following holds: Every monotone $f : \{0,1\}^n \to \{0,1\}$ can be ϵ-approximated by a depth-d circuit of size at most*

$$\exp\left((K \cdot \mathbb{I}(f))^{1/(d-1)}\right),$$

for some $d \geq 2$.

(Recall that f is **monotone** if $x \leq y \Rightarrow f(x) \leq f(y)$.) Observation 1.1 implies that the BKS Conjecture could also add the condition that width is at most $(K \cdot \mathbb{I}(f))^{1/(d-1)}$ without loss of generality.

If this conjecture were true it would be an important characterization of monotone functions with small total influence; if it were further true with d fixed to 2 it would yield very interesting positive results for approximation by DNF (or CNF).

1.3 Approximating Majority by DNF

Suppose the BKS Conjecture were true even with d fixed to 2. This would imply that for every constant $\epsilon > 0$, every monotone function f could be ϵ-approximated by a DNF or CNF of size $\exp(O(\mathbb{I}(f)))$. Using Observation 1.1, we could further make the width of the approximator $O(\mathbb{I}(f))$. One reason to hope that this is true is that it *is* true, even for non-monotone functions, if one allows a more powerful class of depth-2 circuits:

Definition 1.4 *A* TOP *("threshold of parities" [Jac95]) is a depth-*2 *circuit with Parity gates at the lower level and a Majority gate on top.*

Proposition 1.5 *For all $\epsilon > 0$, every boolean function f is ϵ-approximated by a TOP of width $O(\mathbb{I}(f)/\epsilon)$.*

This proposition was shown in [KKL88, LMN93] by relating the total influence of a function to its Fourier spectrum.

TOP circuits arise frequently as the hypothesis class in many uniform-distribution learning algorithms. Examples include Linial, Mansour, and Nisan's algorithm for learning depth-d size-s circuits [LMN93], Jackson's Harmonic Sieve algorithm for learning polynomial-size DNFs [Jac95], Bshouty and Tamon's algorithm for learning monotone functions [BT96], and O'Donnell and Servedio's algorithm for learning monotone polynomial-size decision trees [OS06]. (Incidentally, except for Jackson's algorithm, all of these proceed by proving upper bounds on total influence.) An open question in learning theory is whether these algorithms (especially Jackson's DNF algorithm) can be made to use the simpler class of DNFs as their hypothesis class.

This suggests the idea of trying to approximate TOPs by DNFs. By Proposition 1.5, approximating TOPs by DNFs could also be considered a way of attacking the BKS Conjecture. Now the Parities in a TOP could be converted to DNFs or CNFs of no greater width. But how to approximate the Majority by a small DNF or CNF is an interesting question. We solve the problem of ϵ-approximating Majority by DNFs in Sections 2 and 3. Unfortunately, the size necessary is too large to give good approximations of TOPs.

The question of computing Majority by small circuits has a long and interesting history. Significant work has gone into computing Majority with small circuits of various sorts [PPZ92, AKS83, HMP06, Bop86, Val84]. Some of this work involves the subproblem of constructing small circuits for "approximate-Majority" — i.e., circuits that correctly compute Majority whenever the number of 1's in the input is at least a 2/3 fraction or at most a 1/3 fraction. Note that this notion of approximation is not at all the same as our notion. Constructions of constant-depth circuits for this "approximate-Majority" have had important consequences for complexity theory [Ajt83, Ajt93, Vio05]. It seems, however, that no paper has previously investigated the existence of small constant-depth circuits for Majority that are ϵ-approximators in our sense.

Our result on this topic is the following, following from the main results proved in Sections 2 and 3:

Theorem 1.6 *For every constant $0 < \epsilon < 1/2$, the Majority function on n bits can be ϵ-approximated by a DNF of size $\exp(O(\sqrt{n}))$, and this is best possible up to the constant in the exponent.*

Note that the following fact is well known:

Proposition 1.7 *Every monotone function $f : \{0,1\}^n \to \{0,1\}$ satisfies $\mathbb{I}(f) \le \mathbb{I}(\mathrm{Maj}_n) = (\sqrt{2/\pi} + o(1))\sqrt{n}$.*

Thus Theorem 1.6 shows that the BKS Conjecture with d fixed to 2 is true for the Majority function.

Our proof of the upper bound in Theorem 1.6 is by the probabilistic method; we essentially use the random DNF construction of Talagrand [Tal96]. Our proof of the lower bound in Theorem 1.6 uses the Kruskal-Katona Theorem to show that even ϵ-approximators for Majority must have total influence $\Omega(\sqrt{n})$; the lower bound then follows from Theorem 1.3:

Theorem 1.8 *Suppose $f : \{0,1\}^n \to \{0,1\}$ is a $(1/2 - \delta)$-approximator for Maj_n, for constant $\delta > 0$. Then any depth-d circuit computing f requires size $\exp(\Omega(n^{1/(2d-2)}))$.*

For a discussion of why switching lemmas do not seem to provide any help in proving lower bounds on ϵ-approximating DNFs, please see Appendix B.

1.4 Threshold phenomena and the BKS Conjecture

One of the main motivations behind the BKS Conjecture is to provide general conditions under which a monotone function has large total influence. Benjamini, Kalai, and Schramm made their conjecture in the context of problems about threshold phenomena and noise sensitivity in random graphs. There, proving lower bounds on total influence is important, as the total influence relates to certain "critical exponents" in percolation problems, and it also captures the sharpness of "thresholds" for graph properties.

To understand the connection to threshold phenomena, consider the Erdős-Rényi random graph model on v vertices, and write $n = \binom{v}{2}$. Now a boolean string in $\{0,1\}^n$ can be identified with a graph, and a boolean function $f : \{0,1\}^n \to \{0,1\}$ can be identified with a collection of graphs. We say that f is a **graph property** if it closed under permutations of the v vertices. Suppose f is a nontrivial *monotone* graph property (i.e., f is a monotone function that is not constantly 0 or 1). Then as we increase the edge probability p from 0 to 1, the probability that a random graph from the p-biased distribution on $\{0,1\}^n$ satisfies f increases continuously from 0 to 1. Hence there will be a critical exponent p^* where the probability of a random graph satisfying f is $1/2$. It is of great interest to understand how rapidly the probability of satisfying p jumps from near 0 to near 1 in the interval around p^*. The Russo-Margulis Lemma [Mar74, Rus78] shows that $\frac{\partial}{\partial p}\mathbf{E}[f] = 4p(1-p)\mathbb{I}^{(p)}(f)$, for an appropriate p-biased definition of total influence. It follows that graph properties having "sharp" thresholds corresponds to them having large total influence.

A celebrated theorem of Friedgut [Fri99] provides a version of the depth-2 BKS Conjecture in the context of graph properties with $p^* = o(1)$:

Friedgut's Theorem *There is a function $K = K(C, \epsilon) < \infty$ such that the following holds: If f is a monotone graph property with critical probability $p^* = o(1)$ and $\mathbb{I}^{(p^*)}(f) \le C$, then f can be ϵ-approximated (with respect to the p^*-biased distribution on $\{0,1\}^n$) by a DNF of width $K(C, \epsilon)$. In particular, one can take $K(C, \epsilon) = O(C/\epsilon)$.*

This result has been used with great success to show that various natural graph properties — and also random k-SAT problems — have sharp thresholds (see, e.g., [Fri05]); one proves this essentially by showing that the property cannot be well approximated by a small-width DNF.

The relationship between sharp thresholds and large total influence continues to hold in the context of general monotone boolean functions (i.e., not necessarily graph properties). Indeed, there has been significant interest in trying to extend Friedgut's Theorem to the general, no-symmetry case. The BKS Conjecture is one proposal for such an extension (in the case of $p^* = 1/2$). It is weaker than the Friedgut Theorem in that it allows for approximating circuits of depth greater than

2. However the BKS Conjecture's size/width bound for $d = 2$ is very strong, essentially matching Friedgut's Theorem — it states that in the $d = 2$ case, K may be taken to have a linear dependence on $\mathbb{I}(f)$.

Some partial progress has been made towards proving Friedgut's Theorem in the case of general monotone boolean functions. In an appendix to Friedgut's paper, Bourgain [Bou99] showed that every boolean function with $\mathbb{I}(f) \leq C$ has a Fourier coefficient $\hat{f}(S)$ with $|S| \leq O(C)$ and $|\hat{f}(S)| \geq \exp(-O(C^2))$; he used this to show that when f is monotone and $p^* = o(1)$, there is a term of width $O(C)$ that has $\exp(-O(C^2))$-correlation with f. Friedgut himself later showed [Fri98] that his theorem can be extended to general functions, even non-monotone ones (assuming p^* is bounded away from 0 and 1), at the expense of taking $K(C, \epsilon) = \exp(O(C/\epsilon))$.

However it turns out that these generalizations cannot be taken too far — our main result in Section 4 is that the BKS Conjecture is false. Specifically, we show:

Theorem 1.9 *There is a monotone function $\mathcal{F} : \{0,1\}^n \to \{0,1\}$ with total influence $\mathbb{I}(\mathcal{F}) \leq O(\log n)$ satisfying the following: Any DNF or CNF that $.01$-approximates $\mathcal{F}$ requires width $\Omega(n/\log n)$ and hence size $2^{\Omega(n/\log n)}$; and,* any *circuit that $.01$-approximates $\mathcal{F}$ requires size $\Omega(n/\log n)$.*

This rules out the BKS Conjecture. In particular, it shows that Friedgut's Theorem cannot be proved for general monotone functions (in the $p^* = 1/2$ case) unless one takes $K(C, .01) \geq \exp(\Omega(C))$. We remark that the function $\mathcal{F}$ used in the theorem is is computed by a polynomial-size, depth-3 circuit.

2 Approximating Majority

In this section we give a construction of a DNF of size $2^{O(\sqrt{n}/\epsilon)}$ that ϵ-approximates Majority on n bits. In the next section we will show this result is optimal up to the constant in the exponent.

Our construction is by the probabilistic method, inspired by the random DNF construction of Talagrand [Tal96]:

Theorem 2.1 *For all $\epsilon \geq 1/\sqrt{n}$, there is a DNF of width $w = \frac{1}{\epsilon}\sqrt{n}$ and size $(\ln 2)2^w$ which is an $O(\epsilon)$-approximator for* Maj_n.

Proof: Let $\boldsymbol{D}$ be a randomly chosen DNF with $(\ln 2)2^w$ terms, where each term is chosen by picking w variables independently with replacement. It suffices to show that

$$\mathop{\mathbf{E}}_{\boldsymbol{D}}[\mathop{\mathbf{Pr}}_{\boldsymbol{x}}[\boldsymbol{D}(\boldsymbol{x}) \neq \mathrm{Maj}(\boldsymbol{x})]] \leq O(\epsilon), \tag{1}$$

because then a particular D must exist which has $\mathbf{Pr}[D(\boldsymbol{x}) \neq \mathrm{Maj}(\boldsymbol{x})] \leq O(\epsilon)$. Showing (1) is equivalent to showing

$$\mathop{\mathbf{E}}_{\boldsymbol{x}}[\mathop{\mathbf{Pr}}_{\boldsymbol{D}}[\boldsymbol{D}(\boldsymbol{x}) \neq \mathrm{Maj}(\boldsymbol{x})]] \leq O(\epsilon). \tag{2}$$

Given a string $\boldsymbol{x} \in \{0,1\}^n$, let us write the fraction of 1's in the string as $\frac{1}{2} + \frac{1}{2}(\boldsymbol{t}/\sqrt{n})$. Note that the distribution on $\boldsymbol{t} \in [-\sqrt{n}, \sqrt{n}]$ is close to being normal, by the Central Limit Theorem.

We have that $\mathrm{Maj}(\boldsymbol{x}) = 1$ iff $\boldsymbol{t} > 0$, and furthermore, by construction, $\mathbf{Pr}_{\boldsymbol{D}}[\boldsymbol{D}(\boldsymbol{x}) = 1]$ only depends on $\boldsymbol{t}$. Indeed,

$$\mathbf{Pr}[\boldsymbol{D}(\boldsymbol{x}) = 1] = 1 - \Big(1 - (\tfrac{1}{2} + \tfrac{1}{2}(\boldsymbol{t}/\sqrt{n}))^w\Big)^{(\ln 2)2^w} = 1 - \Big(1 - 2^{-w}(1 + \boldsymbol{t}/\sqrt{n})^w\Big)^{(\ln 2)2^w}.$$

(We chose the size to be $(\ln 2)2^w$ so that this quantity would go to $1/2$ as $\boldsymbol{t}$ goes to 0.) So to show (2), it suffices to show that

$$\mathop{\mathbf{E}}_{\boldsymbol{x}}\left[\left(1-2^{-w}(1+\boldsymbol{t}/\sqrt{n})^w\right)^{(\ln 2)2^w} \;\middle|\; \boldsymbol{t}>0\right] \;\leq\; O(\epsilon), \tag{3}$$

$$\text{and} \qquad \mathop{\mathbf{E}}_{\boldsymbol{x}}\left[1-\left(1-2^{-w}(1+\boldsymbol{t}/\sqrt{n})^w\right)^{(\ln 2)2^w} \;\middle|\; \boldsymbol{t}<0\right] \;\leq\; O(\epsilon). \tag{4}$$

For (3) we use $(1-y)^r \leq \exp(-yr)$ (for $0 \leq y \leq 1$, $r > 0$) to get

$$\left(1-2^{-w}(1+\boldsymbol{t}/\sqrt{n})^w\right)^{(\ln 2)2^w} \leq (1/2)^{(1+\boldsymbol{t}/\sqrt{n})^w} \leq (1/2)^{1+\boldsymbol{t}/\epsilon},$$

where we also are using $w = \frac{1}{\epsilon}\sqrt{n}$ and the assumption $\boldsymbol{t} > 0$. For (4) we use $(1-y)^r \geq 1 - yr$ to get

$$1-\left(1-2^{-w}(1+\boldsymbol{t}/\sqrt{n})^w\right)^{(\ln 2)2^w} \leq (\ln 2)(1+\boldsymbol{t}/\sqrt{n})^w \leq (\ln 2)\exp(w\boldsymbol{t}/\sqrt{n}) = (\ln 2)\exp(\boldsymbol{t}/\epsilon),$$

where we used the assumption $\boldsymbol{t} < 0$, and $(1-y)^r \leq \exp(-yr)$ again. Hence to prove (3) and (4), it remains to show

$$\mathop{\mathbf{E}}_{\boldsymbol{x}}[\exp(-|\boldsymbol{t}|/\epsilon)] \leq O(\epsilon). \tag{5}$$

It is relatively easy to check that this holds when $\boldsymbol{t}$ is truly normally distributed. With its actual binomial distribution, we use the following fact: for each $i = 0, 1, 2, \ldots,$

$$\mathbf{Pr}\Big[|\boldsymbol{t}| \in [2^i\epsilon, 2^{i+1}\epsilon]\Big] = \mathbf{Pr}\Big[|N(0,1)| \in [2^i\epsilon, 2^{i+1}\epsilon]\Big] \pm O(1/\sqrt{n}),$$

which follows from the Berry-Esseen Central Limit Theorem. We have $\mathbf{Pr}[|N(0,1)| \in [2^i\epsilon, 2^{i+1}\epsilon]] \leq O(2^i\epsilon)$, and so the additive $O(1/\sqrt{n})$ is negligible since $\epsilon \geq 1/\sqrt{n}$. Using also $\mathbf{Pr}[|\boldsymbol{t}| \in [0,\epsilon]] \leq O(\epsilon)$ (again by Berry-Esseen and $\epsilon \geq 1/\sqrt{n}$), we get:

$$\mathop{\mathbf{E}}_{\boldsymbol{x}}[\exp(-|\boldsymbol{t}|/\epsilon)] \leq O(\epsilon) + \sum_{i=1}^{\infty} \exp(-2^i) \cdot O(2^i\epsilon) \leq O(\epsilon),$$

since $\sum_{i=0}^{\infty} \exp(-2^i)2^i \leq O(1)$, and this proves (5). □

3 A Lower Bound for Majority, via Total Influence

The main result in this section shows that corrupting the Majority function, Maj_n, on even a large fraction of strings cannot decrease its total influence very much:

Theorem 3.1 *Let $f : \{0,1\}^n \to \{0,1\}$ be an ϵ-approximator for Maj_n. Then*

$$\mathbb{I}(f) \geq \begin{cases} (1-O(\epsilon)) \cdot \mathbb{I}(\mathrm{Maj}_n) & \textit{if } \omega(\frac{1}{\sqrt{n}}) \leq \epsilon \leq 1/4, \\ \Omega(\epsilon) \cdot \mathbb{I}(\mathrm{Maj}_n) & \textit{if } 1/4 \leq \epsilon \leq 1/2 - \omega(\frac{1}{\sqrt{n}}). \end{cases}$$

As mentioned in Proposition 1.7, the total influence of Maj_n is $\Theta(\sqrt{n})$. Thus Boppana's relation, Theorem 1.3, implies the following:

Corollary 3.2 *For any constant $\epsilon < 1/2$, every ϵ-approximator for* Maj_n *with depth d requires size at least*

$$\exp\left(\Omega(n^{1/(2d-2)})\right).$$

In particular, any ϵ-approximating DNF for Majority requires size at least $\exp(\Omega(\sqrt{n}))$.

This matches the upper bound we proved in Theorem 2.1, up to the constant in the exponent.

The remainder of this section is devoted to proving Theorem 3.1. In Appendix A we include an alternate proof of the following much weaker statement: If $\epsilon > 0$ is sufficiently small then $\mathbb{I}(f) \geq \Omega(\sqrt{n})$ for all f that ϵ-approximate Majority.

The first basic fact we will need is that we can assume without loss of generality that the approximators f are monotone.

Proposition 3.3 *Let $f : \{0,1\}^n \to \{0,1\}$ be an ϵ-approximator for some monotone function $g : \{0,1\}^n \to \{0,1\}$ (e.g.,* Maj_n*). Then there is* monotone $f' : \{0,1\}^n \to \{0,1\}$ *that ϵ-approximates g and has $\mathbb{I}(f') \leq \mathbb{I}(f)$.*

Proof: Recall the *combinatorial shifting* operators $\kappa_1, \dots, \kappa_n$ introduced by Kleitman [Kle66]; the operator κ_i applied to f yields the function given by

$$(\kappa_i f)(x) = \begin{cases} f(x) & \text{if } f(x) = f(\sigma_i x), \\ x_i & \text{if } f(x) \neq f(\sigma_i x), \end{cases}$$

where $\sigma_i x$ denotes the string x with the ith coordinate flipped. Let $f' = \kappa_1 \kappa_2 \cdots \kappa_n f$; it is well known that this makes f' a monotone function. The fact that $\mathbb{I}(f') \leq \mathbb{I}(f)$ follows because the κ_i operators never increase total influence [BOL90]. Finally, it is easy to see that the κ_i operators can only improve approximation with respect to a monotone function g; this shows that f' is an ϵ-approximation for g. □

The second basic fact we'll need involves the following definition:

Definition 3.4 *Given $f : \{0,1\}^n \to \{0,1\}$, we define $C(f)$ to be the expected number of "correct" bits in a random string $\boldsymbol{x}$; i.e.,*

$$C(f) = \mathop{\mathbf{E}}_{\boldsymbol{x}}[\#\{i : \boldsymbol{x}_i = f(\boldsymbol{x})\}]$$

Lemma 3.5 *Let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function. Then $C(f) = n/2 + \mathbb{I}(f)/2$.*

Proof: Clearly $C(f) = \sum_{i=1}^n \mathbf{Pr}_{\boldsymbol{x}}[\boldsymbol{x}_i = f(\boldsymbol{x})]$. Thus it suffices to show that $\mathbf{Pr}_{\boldsymbol{x}}[\boldsymbol{x}_i = f(\boldsymbol{x})] = 1/2 + \mathrm{Inf}_i(f)/2$ for each i. When $\boldsymbol{x}$ is chosen randomly, there is an $\mathrm{Inf}_i(f)$ chance that $\boldsymbol{x}_i$ is influential for f. In this case, the expected number of correct bits $\boldsymbol{x}_i$ is 1; this is because f is monotone so $f(x)$ agrees with $f(\boldsymbol{x}_i)$. With probability $1 - \mathrm{Inf}_i(f)$ the bit $\boldsymbol{x}_i$ is not influential for f; in this case the expected number of correct bits $\boldsymbol{x}_i$ is 1/2, since on of the two possibilities will agree with f's constant value. Thus the overall probability of a correct $\boldsymbol{x}_i$ is $\mathrm{Inf}_i(f) \cdot 1 + (1 - \mathrm{Inf}_i(f)) \cdot \frac{1}{2} = 1/2 + \mathrm{Inf}_i(f)/2$, as claimed. □

We will now prove Theorem 3.1 under an assumption, namely, that f only disagrees with Maj on the strings where Maj is 0. (Recall that we are assuming f is monotone.) We will show later that we can get rid of this assumption.

Theorem 3.6 *Theorem 3.1 holds if $f \geq \mathrm{Maj}$.*

We will define some functions. Define $\mathrm{M}_0 := \mathrm{Maj}$. For an integer $0 \leq t \leq 2^{n-1}$, M_t will agree with M_{t-1} on all strings except one. That string is the "largest" string in $\mathrm{M}_{t-1}^{-1}(0)$, where "largest" is respect to the ordering when strings are interpreted as binary integers. We will view these functions as a process, where strings are being added to $\mathrm{M}_t^{-1}(1)$ as t increases. We will refer to the unique string x such that $\mathrm{M}_t(x) = 1$ and $\mathrm{M}_{t-1}(x) = 0$ as the string added at time t. For example, if $n = 5$, the first few strings added are 11000, 10100, 10010, 10001, 10000, 01100, etc. We also define $w_{j,t}$ as the "largest" string x with $|x| = j$ such that $\mathrm{M}_t(x) = 0$. The string $w_{j,t}$ is the next string of Hamming weight j whose value becomes 1 as t increases.

We will show that if f is a monotone function such that f disagrees with Maj on exactly $t \leq \epsilon(2^n)$ strings, then $\mathbb{I}(f) \geq \mathbb{I}(\mathrm{M}_t)$. Fix t, and let $\mathrm{M} := \mathrm{M}_t$.

For any function g: $\{0,1\}^n \to \{0,1\}$, let $X_j(g)$ be the set of strings such that $|x| = j$, $g(x) = 1$ and $\mathrm{Maj}(x) = 0$. Define $X(g)$ as the vector $(|X_0(g)|, |X_1(g)|, \ldots, |X_{(n-1)/2}(g)|)$. Note that since we are assuming that f differs from Maj on t strings of Hamming weight at most $(n-1)/2$, the sum of the entries of $X(\mathrm{M})$ equals the sum of the entries of $X(f)$, or equivalently, the sum of the entries of $X(\mathrm{M}) - X(f)$ is 0.

Claim 3.7 *The vector $X(\mathrm{M})-X(f)$ has all its nonnegative entries preceding all its negative entries.*

Proposition 3.8 *Claim 3.7 implies Theorem 3.6*

Proof: Suppose that the claim is true. Assuming $f \geq \mathrm{Maj}$, we have $C(\mathrm{M})-C(f) = \sum_{i=0}^{(n-1)/2} (X_i(\mathrm{M})-X_i(f))(-n+2i)$. The sum of the entries of $X(\mathrm{M}) - X(f)$ is 0, as M disagrees with Maj on t strings and f disagrees with Maj on at most t strings. In the weighted sum given, the weight on entry i increases with i, so if all the nonnegative entries of $X(\mathrm{M}) - X(f)$ come first (getting lower weights), and the sum of the entries of $X(\mathrm{M}) - X(f)$ is 0, then $C(\mathrm{M}) - C(f) \leq 0$. Thus $C(\mathrm{M}) \leq C(f)$, and by Lemma 3.5, $\mathbb{I}(\mathrm{M}) \leq \mathbb{I}(f)$. □

Proposition 3.9 *Claim 3.7 is true if $f \geq \mathrm{Maj}$.*

Proof: Our proof will use the Kruskal-Katona theorem; in order to do this, we require some definitions. For a set A of strings of Hamming weight j, we define the upper shadow of A as $\partial_u A = \{y : |y| = j+1, \text{and } \exists x \in A \text{ such that } x < y\}$. We define the lexicographic order on sets of size j in the following way: Let $S(x)$ for any string x be the length-j vector of indices i such that $x_i = 1$ in increasing order. So $S(1101000) = (1, 2, 4)$. Then $x < y$ if $S(x)_1 < S(y)_1$, or $S(x)_1 = S(y)_1$ and $S(x)_2 < S(y)_2$, or $S(x)_1 = S(y)_1$,$S(x)_2 = S(y)_2$,and $S(x)_3 < S(y)_3$,. . ., or $S(x)_1 = S(y)_1$,$S(x)_2 = S(y)_2$,. . . ,$S(x)_{j-1} = S(y)_{j-1}$, and $S(x)_j < S(y)_j$. The 10 strings of length 5 and Hamming weight 2 in this ordering are $11000, 10100, 10010, 10001, 01100, 01010, 01001, 00110, 00101, 00011$.

We can now state the Kruskal-Katona theorem.

Theorem 3.10 *Suppose A is a set of strings of Hamming weight j, and B is the set of the first $|A|$ strings of Hamming weight j in lexicographic order. Then $|\partial_u A| \geq |\partial_u B|$.*

We will require a few claims.

Claim 3.11 *For any t and j, $X_j(\mathrm{M}_t)$ is a set of least strings in lexicographic order.*

This claim follows from the fact that lexicographic order is a suborder of the order we get when we order all binary strings by comparing them as numbers in decreasing order.

Claim 3.12 *Let u be the string that is added at time $t+1$, and suppose $|u| = j+1$. Then $u \geq w_{j,t}$.*

Proof: Suppose not. Compare u and $w_{j,t}$ as binary numbers. If u is less than $w_{j,t}$, then M_t and M_{t+1} would not disagree on u, since as a number, u would not be the largest string such that $\mathrm{M}_t = 0$. So u must be greater than $w_{j,t}$.

Now suppose the claim is false. Then there is a bit k such that $u_k = 0$ and $(w_{j,t})_k = 1$. If we change the least significant 0 bit of $w_{j,t}$ from 0 to 1, the resulting string has Hamming weight $j+1$ and is still as a number less than u. But M_t is a monotone function, so M_t is 1 on this string. But in our process, we derive $\{\mathrm{M}_t\}$ by always flipping the output of the "largest" string, a contradiction. □

Define $W_j(\mathrm{M}) = X_j(\mathrm{M}) \cup \{w_{j,t}\}$.

Claim 3.13 *For any j, $\partial_u(W_j(\mathrm{M})) \supseteq X_{j+1}(\mathrm{M}_t)$, with equality only if the string to be added at time $t+1$ is $w_{j,t}$.*

Proof: Take any x in $X_{j+1}(\mathrm{M}_t)$. There exists $t' \leq t$ such that x was added at time t'. By Claim 3.12, $x \geq w_{j,t'}$. If $w_{j,t} = w_{j,t'}$, then x is in the upper shadow of $w_{j,t}$, and we are done. Otherwise, it must be the case that $w_{j,t'}$ is in $X_j(\mathrm{M}_t)$, and thus x is in the upper shadow of $X_j(\mathrm{M}_t)$. So it follows that x is in the upper shadow of $W_j(\mathrm{M})$. Equality occurs only if all the strings y such that $y \geq w_{j,t}$ are already added, and thus the string added at time $t+1$ will be $w_{j,t}$. □

Given these claims, we can now finish the proof of Theorem 3.6. The theorem is obvious when $\mathrm{M} = f$, so we assume $\mathrm{M} \neq f$. Suppose that the theorem is false. As the sum of the entries of $X(\mathrm{M}) - X(f)$ is 0, then there exists some $0 \leq j < j' \leq (n-1)/2$ such that $|X_j(f)| > |X_j(\mathrm{M})|$, $|X_{j'}(f)| < |X_{j'}(\mathrm{M})|$, and $|X_i(f)| = |X_i(\mathrm{M})|$ for $j < i < j'$. It is possible that $j = j'-1$ and no such i exists.

As f is monotone, $X_{j+1}(f) \supseteq \partial_u(X_j(f))$, and so

$$|X_{j+1}(f)| \geq |\partial_u(X_j(f))| \tag{6}$$

By the Kruskal-Katona theorem, $|\partial_u(X_j(f))|$ is at least as large as the upper shadow of the first $|X_j(f)|$ strings in lexicographic order. Consider $X_j(\mathrm{M})$. By Claim 3.11, the set of strings $X_j(\mathrm{M})$ is precisely the first $|X_j(\mathrm{M})|$ strings in lexicographic order. So if $|X_j(f)| > |X_j(\mathrm{M})|$, it must be true that the size of the upper shadow of the first $|X_j(f)|$ strings in lexicographic order is at least as large as the upper shadow of the first $|X_j(\mathrm{M})| + 1$ strings in lexicographic order. But the first $|X_j(\mathrm{M})| + 1$ strings in lexicographic order is precisely $W_j(\mathrm{M})$ by definition, so

$$|X_{j+1}(f)| \geq |\partial_u(W_j(\mathrm{M}))| \tag{7}$$

By Claim 3.13, $\partial_u(W_j(\mathrm{M}))$ contains $X_{j+1}(\mathrm{M})$, and thus $|\partial_u(W_j(\mathrm{M}))| \geq |X_{j+1}(\mathrm{M})|$. Along with (7), we have that

$$|X_{j+1}(f)| \geq |X_{j+1}(\mathrm{M})|. \tag{8}$$

We have assumed that $|X_{j+1}(f)| \leq |X_{j+1}(\mathrm{M})|$, so if $|X_{j+1}(f)| < |X_{j+1}(\mathrm{M})|$, we have our contradiction and we are done. We must now handle the case that

$$|X_{j+1}(f)| = |X_{j+1}(\mathrm{M})|. \tag{9}$$

Putting (7) and (9) together yields

$$|X_{j+1}(\mathrm{M})| \geq |\partial_u(W_j(\mathrm{M}))|. \tag{10}$$

By Claim 3.13, $X_{j+1}(\mathrm{M}) \subseteq \partial_u(W_j(\mathrm{M}))$. Adding this to (10) yields

$$X_{j+1}(\mathrm{M}) = \partial_u(W_j(\mathrm{M})). \tag{11}$$

By analyzing the order in which strings are added, for all $j < k \leq (n-3)/2, X_{k+1}(\mathrm{M}) = \partial_u(X_k(\mathrm{M}))$.

Now since $|X_j(f)| > |X_j(\mathrm{M})|$ and the entries of $X(\mathrm{M}) - X(f)$ sum to 0, there exists some $j' > j$ such that $|X_{j'}(f)| = |X_{j'}(\mathrm{M})|$ and $|X_{j'+1}(f)| < |X_{j'+1}(\mathrm{M})|$. We get

$$|X_{j'+1}(f)| \geq |\partial_u X_{j'}(f)| \geq |\partial_u X_{j'}(\mathrm{M})| = |X_{j'+1}(\mathrm{M})|, \tag{12}$$

where the first inequality follows from monotonicity, and the middle inequality follows from the Kruskal-Katona theorem. This yields our desired contradiction. □

We now analyze the functions M_t. We will select a few of the M_t to ease our calculations. As t increases, eventually we add a string consisting of a block of 1's followed by a block of 0's. At this point, we get $\mathrm{M}_t = \mathrm{Maj} \vee (\bigwedge_{i=1}^k x_i)$ for some k. Define t_0 such that $\mathrm{M}_{t_0} = \mathrm{Maj} \vee x_1$. All strings with $x_1 = 1$ are 1, so the "largest" 0 strings in M_t where $t > t_0$ have x_2 1. At some time t_1, we will add the string that is all 0's except for a 1 in the kth bit. Then $\mathrm{M}_{t_k} = \mathrm{Maj} \vee (\bigvee_{i=1}^k x_i)$. Also note that $\mathrm{M}_{2^{n-1}}$ is the constant function 1. Also note that since the string added at time t had $\mathrm{Maj} = 0$ before time t, the expected number of correct bits goes down. So for all $t > 0$, $C(\mathrm{M}_t) \leq C(\mathrm{M}_{t-1}$ and by Lemma 3.5, $\mathbb{I}(\mathrm{M}_t) \leq \mathbb{I}(\mathrm{M}_{t-1})$.

Define $g_k := \mathrm{Maj} \vee x_1 x_2 \ldots x_k$ and $h_k := \mathrm{Maj} \vee x_1 \vee x_2 \vee \ldots \vee x_k$. We will first find the probability that each of these functions differ from Maj, then compute their average sensitivities. We will do this assuming $k = o(\log n)$.

The following claim will be useful in the analysis the functions g_k and h_k.

Claim 3.14 *For $k = o(\log n)$, $2^{-n}\binom{n}{(n+1)/2} = 2^{-n}\binom{n}{(n-1)/2} \sim 2^{-n-k}\binom{n-k}{\frac{n}{2} \pm O(k)} \sim \frac{1}{\sqrt{2\pi n}}$.*

Proof: This follows from the Central Limit Theorem. □

We will start by showing how well g_k and h_k approximates Maj. First note that $g_k \geq \mathrm{Maj}$, so $\mathbf{Pr}_{\boldsymbol{x}}[g_k(\boldsymbol{x}) \neq \mathrm{Maj}(\boldsymbol{x})] = \mathbf{Pr}_{\boldsymbol{x}}[g_k(\boldsymbol{x}) = 1 \wedge \mathrm{Maj}(\boldsymbol{x}) = 0] = \mathbf{Pr}_{\boldsymbol{x}}[\mathrm{Maj}(\boldsymbol{x}) = 0 | g_k(\boldsymbol{x}) = 1](2^{-k})$. Suppose $\boldsymbol{x}_1\boldsymbol{x}_2 \ldots \boldsymbol{x}_k$ is 1. Then $\boldsymbol{x}_1 = \boldsymbol{x}_2 = \ldots = \boldsymbol{x}_k = 1$, so $\mathrm{Maj}(\boldsymbol{x})$ is 0 only if at most $(n-1)/2$ of the remaining $n-k$ bits are 0. However, if $k = o(\log n)$, the probability that this happens is close to 1/2 [Karl: **I guess this doesn't follow from our claim**]. So g_k is a $\Theta(2^{-k-1})$-approximator for Maj. *

For h_k, Again note that $h_k \geq \mathrm{Maj}$. Thus, $\mathbf{Pr}_{\boldsymbol{x}}[h_k(\boldsymbol{x}) \neq \mathrm{Maj}(\boldsymbol{x})] = \mathbf{Pr}_{\boldsymbol{x}}[h_k(\boldsymbol{x}) = 1 \wedge \mathrm{Maj}(\boldsymbol{x}) = 0] = \mathbf{Pr}_{\boldsymbol{x}}[\mathrm{Maj}(\boldsymbol{x}) = 0] - \mathbf{Pr}_{\boldsymbol{x}}[\mathrm{Maj}(\boldsymbol{x}) = 0 \wedge h_k(\boldsymbol{x}) = 0] = (1/2 - \mathbf{Pr}_{\boldsymbol{x}}[\mathrm{Maj}(\boldsymbol{x}) = 0 | h_k(\boldsymbol{x}) = 0])(2^{-k})$. Suppose $\boldsymbol{x}_1 = \boldsymbol{x}_2 = \ldots = \boldsymbol{x}_k = 0$. The probability that $\mathrm{Maj}(\boldsymbol{x})$ is 0 is the probability that at most $(n-k-1)/2$ of the remaining $n-k$ bits are 0. But since $k = o(\log n)$, the probability that his happens is close to 1/2 [Karl: **Or this either**]. So h_k is a $(1/2 - \Theta(2^{-k-1}))$-approximator for Maj. *

We now analyze the total influence of g_k and h_k. For any function g, let $s(g,x) = |\{y \sim x : g(x) \neq g(y)\}|$. In the case of Maj, we have that

$$\mathbb{I}(\mathrm{Maj}) = \mathop{\mathbf{E}}_{\boldsymbol{x}}[s(\mathrm{Maj},\boldsymbol{x})] = (\frac{n+1}{2}) \mathop{\mathbf{Pr}}_{\boldsymbol{x}}[|\boldsymbol{x}| = (n \pm 1)/2]$$

By definition, $\mathbb{I}(g_k) = \mathbf{E}_{\boldsymbol{x}}[s(g_k,\boldsymbol{x})]$. We will assume $s(g_k,\boldsymbol{x})$ is nonzero only for strings $\boldsymbol{x}$ where $\boldsymbol{x}_1\boldsymbol{x}_2\ldots\boldsymbol{x}_k$ is 0, so we get $\mathbb{I}(g_k) \geq (1-2^{-k})\,\mathbf{E}_{\boldsymbol{x}}[s(g_k,\boldsymbol{x})|\boldsymbol{x}_1\boldsymbol{x}_2\ldots\boldsymbol{x}_k\ =0]$. The assignment to $\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_k$ satisfying that the AND of them is 0 that minimizes the expectation is the one that minimizes the probability of the string being sensitive, which is the assignment where all the bits are 0. Thus,

$$\begin{aligned}
\mathbb{I}(g_k) &\geq (1-2^{-k}) \mathop{\mathbf{E}}_{\boldsymbol{x}}[s(g_k,\boldsymbol{x})|\boldsymbol{x}_1 = \boldsymbol{x}_2 = \ldots = \boldsymbol{x}_k = 0] \\
&= (1-2^{-k})(\frac{n+1}{2}) \mathop{\mathbf{Pr}}_{\boldsymbol{x}}[\boldsymbol{x} \text{ has exactly } (n \pm 1)/2 + k \ 1'\text{s in indices greater than } k] \\
&= (1-2^{-k})(\frac{n+1}{2})2^{-(n-k)}(\binom{n-k}{(n+1)/2+k} + \binom{n-k}{(n-1)/2+k}) \\
&\sim (1-2^{-k})(\frac{n+1}{2})2^{-n}(\binom{n}{(n+1)/2} + \binom{n}{(n-1)/2}) \text{ [using Claim 3.14]} \\
&= (1-2^{-k})(\frac{n+1}{2}) \mathop{\mathbf{Pr}}_{\boldsymbol{x}}[|\boldsymbol{x}| = (n \pm 1)/2] \\
&= (1-2^{-k})\mathbb{I}(\mathrm{Maj}).
\end{aligned}$$

So we can take $\mathbb{I}(g_k) \geq (1-2^{-k})(1-o(1))\mathbb{I}(\mathrm{Maj})$. A similar calculation shows that $\mathbb{I}(h_k) = \mathbf{E}_{\boldsymbol{x}}[s(h_k,\boldsymbol{x})] \geq 2^{-k}\,\mathbf{E}_{\boldsymbol{x}}[s(h_k,\boldsymbol{x})|\boldsymbol{x}_1 = \boldsymbol{x}_2 = \ldots = \boldsymbol{x}_k = 0]$, by only considering strings where $\boldsymbol{x}_1 = \boldsymbol{x}_2 = \ldots = \boldsymbol{x}_k = 0$. Then we get

$$\begin{aligned}
\mathbb{I}(h_k) &\geq 2^{-k}(\frac{n+1}{2}) \mathop{\mathbf{Pr}}_{\boldsymbol{x}}[\boldsymbol{x} \text{ has exactly } (n \pm 1)/2 + k \ 1'\text{s in indices greater than } k] \\
&\sim 2^{-k}(\frac{n+1}{2}) \mathop{\mathbf{Pr}}_{\boldsymbol{x}}[|\boldsymbol{x}| = (n \pm 1)/2] \text{ [using Claim 3.14]} \\
&= 2^{-k}\mathbb{I}(\mathrm{Maj}).
\end{aligned}$$

So we can take $\mathbb{I}(h_k) \geq (2^{-k})(1-o(1))\mathbb{I}(\mathrm{Maj})$. It follows now from the above and 3.7 that if $f \geq \mathrm{Maj}$ and f is ϵ-close to Maj, then (1) if $\epsilon < 1/4$, then $\mathbb{I}(f) \geq (1-\Omega(\epsilon))\mathbb{I}(\mathrm{Maj})$, and (2) if $1/4 \geq \epsilon < 1/2$, then $\mathbb{I}(f) \geq \Omega(1/2-\epsilon)\mathbb{I}(\mathrm{Maj})$.

In Proposition 3.9 as well as most of this section, we have assumed that if $f \geq \mathrm{Maj}$. Here we remove this assumption. First note that by symmetry of 1 and 0 (and the Kruskal-Katona theorem on an appropriate order), we could have proved everything in this section in a very similar way assuming that if Maj is 0 then f is 0.

Theorem 3.15 *Proposition 3.9 implies Theorem 3.1.*

Proof: Suppose that f is ϵ-close to Maj. Set $\epsilon = \epsilon_0 + \epsilon_1$, where ϵ_0 is the fraction of strings where f is 1 and Maj is 0, and ϵ_1 is the fraction of strings where f is 0 and Maj is 1. We could think of building f as a process, starting with Maj, and flipping the output of one string at a time. First we flip the ϵ_0 fraction of strings where Maj is 0, then the ϵ_1 fraction of strings where Maj is 1. To minimize total influence, we will add examples as in the process M_t, but on each side. We will assume that we can do this; if we couldn't, it would only help us.

Note that each string that disagrees with Maj contributes a negative amount in total influence, and the contribution gets more negative as the Hamming weight gets further from $n/2$. Suppose instead that we had a function g such that there is an ϵ fraction of strings where g is 1 and Maj is 0, so $g \geq$ Maj. Then from our theorem, $g = \mathrm{M}_{\epsilon 2^n}$. Consider the set of strings such that $g = 1, f = 0$, and Maj $= 0$. There are $\epsilon_1 2^n$ such strings. Note that these strings are a set of consecutive strings in the order comparing strings as binary numbers, where the Hamming weight is at most $(n-1)/2$. But it is true that the expected number of 1's in a set of consecutive strings in this order is maximized when the first strings are taken, causing the least possible decrease in total influence. This is the decrease we will get if we choose to make f and Maj disagree on $\epsilon_1 2^n$ strings where Maj $= 1$, so we can do no worse by having f disagree with Maj only when Maj $= 0$. $\square$

4 Falsifying the BKS Conjecture

In this section our goal is to falsify the BKS Conjecture. In particular, we will have to prove a lower bound for ϵ-approximating a monotone function by DNF and CNF. Note that the technique we used in Section 3 — lower-bounding the total influence of an approximator and then using Theorem 1.3 — is useless here. This is because the BKS Conjecture was made as a converse to Theorem 1.3!

Since we have difficulty enough showing size lower bounds for ϵ-approximating DNF, we should hope that our lower bounds for higher depths follow for an easy reason. This suggests looking for a counterexample among monotone functions with total influence $\ll \log^2 n$, since for such functions we will only have to prove sublinear size lower bounds for ϵ-approximating circuits of depth $d \geq 3$.

The function we will use to falsify the BKS Conjecture will be based on the *Tribes* functions. These were originally introduced by Ben-Or and Linial [BOL90]; we will use slightly different parameters than they did, to simplify notation.

Given $b \in \mathbb{N}$, write $I = \{1, 2, \ldots, 2^b\}$, $J = \{1, 2, \ldots, b\}$, and $n = b2^b$. We define the Tribes function $\mathrm{Tribes}_n : \{0,1\}^n \to \{0,1\}$ as follows. Given an input $x \in \{0,1\}^n$, we index its bits as $x_{i,j}$, for $i \in I$ and $j \in J$. We also write $y_i = \bigwedge_{j \in J} x_{i,j}$. $\mathrm{Tribes}_n(x)$ is then defined to be $\bigvee_{i \in I} y_i$.

In other words, Tribes_n is given by the monotone read-once DNF of width b and size 2^b+1. We have

$$\Pr_{\boldsymbol{x}}[\mathrm{Tribes}_n(\boldsymbol{x}) = 1] = 1 - (1 - 2^{-b})^{2^b} \approx 1 - 1/e,$$

so $\mathbf{Pr}_{\boldsymbol{x}}[\mathrm{Tribes}_n(\boldsymbol{x}) = 1]$ is uniformly bounded away from 0 and 1.

We also define the monotone complement of Tribes_n:

$$\mathrm{Tribes}_n^\dagger(x) = \overline{\mathrm{Tribes}_n(\overline{x}_{1,1}, \overline{x}_{1,2}, \ldots, \overline{x}_{2^b,b})}.$$

The function $\mathrm{Tribes}_n^\dagger(x)$ is given by the monotone read-once CNF of width b and size $2^b + 1$. It has $\mathbf{Pr}[\mathrm{Tribes}_n^\dagger(\boldsymbol{x}) = 1] \approx 1/e$. Ben-Or and Linial showed that $\mathbb{I}(\mathrm{Tribes}_n) = \Theta(\log n)$, and the same holds for $\mathrm{Tribes}_n^\dagger$ by boolean duality.

Suppose we attempt to approximate Tribes_n with some CNF C. We view C as being an AND of ORs, where each OR's wires may pass through a NOT gate before being wired to an input gate $x_{i,j}$.

Now further suppose we introduce additional "input gates" y_i, where each y_i is always equal to $\bigwedge_{j \in J} x_{i,j}$, and we allow the circuit C to access the y_i gates if it wants. Our main lemma uses the fact that Tribes_n depends only on the y_i's to show that C can be taken to only depend on the y_i's as well:

Lemma 4.1 *Suppose* Tribes_n *is* ϵ*-approximated by a CNF* C *of size* s *and width* w *over the variables* $(x_{i,j})_{i \in I, j \in J}$*. Then there is another CNF* C' *of size at most* s *and width at most* w only over the variables $(y_i)_{i \in I}$ *that also* ϵ*-approximates* Tribes_n.

Proof: Given C over the input gates $x_{i,j}$, imagine that every wire going to an input gate $x_{i,j}$ is instead rewired to access $x_{i,j} \vee y_i$. Call the resulting circuit C_1. We claim that C_1 and C compute the same function of x. The reason is that on any input x where $y_i = 0$, the rewiring to $x_{i,j} \vee y_i$ has no effect; and, on any input x where $y_i = 1$, the rewiring to $x_{i,j} \vee y_i$ converts $x_{i,j}$ to 1, but that still has no effect since $y_i = 1 \Rightarrow x_{i,j} = 1$. Since C was an ϵ-approximator for Tribes_n, we have

$$\mathop{\mathbf{Pr}}_{\boldsymbol{x}}[C_1(\boldsymbol{x}, \boldsymbol{y}) \neq \mathrm{Tribes}_n(\boldsymbol{x})] \leq \epsilon.$$

Now picking $\boldsymbol{x}$ uniformly at random induces the 2^{-b}-biased product distribution on $\boldsymbol{y} \in \{0,1\}^I$. We can get the same distribution on $(\boldsymbol{x}, \boldsymbol{y})$ by picking $\boldsymbol{y}$ first and then picking $\boldsymbol{x}$ conditioned on $\boldsymbol{y}$. I.e., for each $i \in I$: if $\boldsymbol{y}_i = 1$ then all $\boldsymbol{x}_{i,j}$'s are chosen to be 1; if $\boldsymbol{y}_i = 0$ then the substring $\boldsymbol{x}_i \in \{0,1\}^J$ is chosen uniformly from $\{0,1\}^J \setminus \{(1,1,\ldots,1)\}$.

In view of this, and using the fact that Tribes_n depends only on $\boldsymbol{y}$, we have

$$\mathop{\mathbf{E}}_{\boldsymbol{y}}\left[\mathop{\mathbf{Pr}}_{\boldsymbol{x} \mid \boldsymbol{y}}\Big[C_1(\boldsymbol{x}, \boldsymbol{y}) \neq \mathrm{Tribes}_n(\boldsymbol{y})\Big]\right] \leq \epsilon.$$

We next introduce new input gates $(z_{i,j})_{i \in I, j \in J}$ that take on random values, completely independent of the $\boldsymbol{x}_{i,j}$'s and the $\boldsymbol{y}_i$'s. Each substring $(\boldsymbol{z}_{i,j})_{j \in J}$ will be uniform on $\{0,1\}^J \setminus \{(1,1,\ldots,1)\}$; i.e., it will have the same distribution as $(\boldsymbol{x}_i)_{j \in J} \mid \boldsymbol{y}_i = 0$. Now let the circuit C_2 be the same as C_1 except with all accesses to the $x_{i,j}$'s replaced by accesses to the corresponding $z_{i,j}$'s.

We claim that for every string $y \in \{0,1\}^I$, the distributions $C_1(\boldsymbol{x}|y, y)$ and $C_2(\boldsymbol{z}, y)$ are identical. The reason is that for each $i \in I$ such that $y_i = 1$, the $(\boldsymbol{x}_{i,j})_{j \in J}$ and $(\boldsymbol{z}_{i,j})_{j \in J}$ values are irrelevant, since C_1 only accesses $x_{i,j}$ by accessing $x_{i,j} \vee y_i$ and the same is true of C_2 and $z_{i,j}$. On the other hand, for each $i \in I$ such that $y_i = 0$, the $(\boldsymbol{x}_{i,j})_{j \in J}$ and $(\boldsymbol{z}_{i,j})_{j \in J}$ values are identically distributed.

In light of this, we conclude

$$\mathop{\mathbf{E}}_{\boldsymbol{y}}\left[\mathop{\mathbf{Pr}}_{\boldsymbol{z}}\Big[C_2(\boldsymbol{z}, \boldsymbol{y}) \neq \mathrm{Tribes}_n(\boldsymbol{y})\Big]\right] \leq \epsilon,$$

which can be switched to

$$\mathop{\mathbf{E}}_{\boldsymbol{z}}\left[\mathop{\mathbf{Pr}}_{\boldsymbol{y}}\Big[C_2(\boldsymbol{z}, \boldsymbol{y}) \neq \mathrm{Tribes}_n(\boldsymbol{y})\Big]\right] \leq \epsilon.$$

Since $\boldsymbol{z}$ and $\boldsymbol{y}$ are independent, we can conclude there must be a particular setting z^* such that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{y}}\Big[C_2(z^*, \boldsymbol{y}) \neq \mathrm{Tribes}_n(\boldsymbol{y})\Big] \leq \epsilon.$$

We may now take C' to be the circuit only over the $\boldsymbol{y}$ gates gotten by fixing the input z^* for C_2. It is easy to check that C' has width at most w and size at most s. □

We can now use Lemma 4.1 to show that Tribes_n has no good CNF approximator of width much smaller than $n/\log n$:

Theorem 4.2 *Any CNF that* $.2$*-approximates* Tribes_n *must have width at least* $(1/3)2^b = \Omega(n/\log n)$*.*

Proof: Let C be a CNF of width w that .2-approximates Tribes_n over the variables $(x_{i,j})_{i\in I, j\in J}$. Using Lemma 4.1, convert it to a CNF C' over the variables $(y_i)_{i\in I}$ that .2-approximates Tribes_n. We may assume that no term in C' includes both y_i and $\overline{y}_i$ for some i. We now consider two cases.

Case 1: Every term in C' includes at least one negated y_i. In this case, C' is 1 whenever $y = (0, 0, \ldots, 0)$. But Tribes_n is 0 when $y = (0, 0, \ldots, 0)$. Since this occurs with probability $(1 - 2^{-b})^{2^b} \geq 1/4 > .2$, we have a contradiction.

Case 2: C' has at least one term in which all y_i's are unnegated. Suppose this term has width w. Since y_i is true only with probability 2^{-b}, this term is true with probability at most $w2^{-b}$, by the union bound. And whenever this term is false, C' is false. Hence $\mathbf{Pr}[C' = 0] \geq 1 - w2^{-b}$. Since $\mathbf{Pr}[\mathrm{Tribes}_n = 0] \leq 1/e$ and C' is a .2-approximator for Tribes_n, we must have $1 - w2^{-b} \leq 1/e + .2 \Rightarrow w2^{-b} \geq 1/3$, completing the proof. □

By symmetry of 0 and 1, we infer:

Corollary 4.3 *Any DNF that* $.2$*-approximates* $\mathrm{Tribes}_n^\dagger$ *must have width at least* $\Omega(n/\log n)$*.*

As an aside, we can now show that the idea of approximating TOPs by DNFs discussed in Section 1.3 cannot work. Since $\mathrm{Tribes}_n^\dagger$ is computable by a polynomial-size CNF, Jackson's Harmonic Sieve learning algorithm [Jac95] can produce a polynomial-size $O(\log n)$-width TOP ϵ-approximator for it, for any constant $\epsilon > 0$. But one can never convert this to even a .2-approximating DNF of size smaller than $2^{\Omega(n/\log n)}$, by Corollary 4.3 combined with Observation 1.1.

We now define the function that contradicts the BKS Conjecture:

Definition 4.4 *Let* n *be of the form* $b2^{b+1}$*. We define* $\mathcal{F}_n : \{0,1\}^n \to \{0,1\}$ *to be the OR of* $\mathrm{Tribes}_{n/2}$ *and* $\mathrm{Tribes}_{n/2}^\dagger$*, on disjoint sets of bits.*

Proposition 4.5 $\mathcal{F}_n$ *is a monotone function computable by a depth-*3 *read-once formula, and* $\mathbb{I}(\mathcal{F}) = O(\log n)$*.*

The fact that $\mathbb{I}(\mathcal{F}_n) = O(\log n)$ holds because $\mathbb{I}(\mathcal{F}_n) \leq \mathbb{I}(\mathrm{Tribes}_{n/2}) + \mathbb{I}(\mathrm{Tribes}_{n/2}^\dagger) = O(\log n) + O(\log n)$.

Theorem 4.6 *Any depth-*2 *circuit that* $.04$*-approximates* $\mathcal{F}_n$ *must have size at least* $2^{\Omega(n/\log n)}$*.*

Proof: Suppose D is a DNF of size s that .04-approximates $\mathcal{F}_n$. By Observation 1.1, we can replace it with a DNF D' of width at most $\log(100s)$ which $.04 + 1/100 = .05$-approximates $\mathcal{F}_n$.

Consider choosing $\boldsymbol{x} \in \{0,1\}^{n/2}$ uniformly at random from the set of strings that make $\mathrm{Tribes}_{n/2}$ false, and also choosing $\boldsymbol{y} \in \{0,1\}^{n/2}$ independently and uniformly at random. Since at least $1/4$ of all strings make $\mathrm{Tribes}_{n/2}$ false (close to $1/e$, in fact), this distribution is uniform on some subset of $\{0,1\}^n$ of fractional size at least $1/4$. Since D' errs in computing $\mathcal{F}_n$ on at most a .05 fraction of strings, we conclude that

$$\mathbf{Pr}[D'(\boldsymbol{x}, \boldsymbol{y}) \neq \mathcal{F}_n(\boldsymbol{x}, \boldsymbol{y})] \leq 4 \cdot .05 = .2.$$

Note that $\mathcal{F}_n(\boldsymbol{x}, \boldsymbol{y})$ is always just $\text{Tribes}^{\dagger}_{n/2}(\boldsymbol{y})$. We conclude that there must be a particular setting of bits $x^* \in \{0,1\}^{n/2}$ such that

$$\mathbf{Pr}[D'(x^*, \boldsymbol{y}) \neq \text{Tribes}^{\dagger}_{n/2}(\boldsymbol{y})] \leq .2.$$

Hence we have a DNF $D'' = D'(x^*, \cdot)$ over $\{0,1\}^{n/2}$ of width at most $\log(100s)$ that .2-approximates $\text{Tribes}^{\dagger}_{n/2}$. By Corollary 4.3, we conclude that $\log(100s) \geq \Omega(n/\log n)$. Hence the original DNF D has size at least $2^{\Omega(n/\log n)}$.

A very similar argument, restricting to the inputs to $\mathcal{F}_n$ where the $\text{Tribes}^{\dagger}_{n/2}$ part is 0 and then using Theorem 4.2 shows that any CNF that is a .04-approximator for $\mathcal{F}_n$ must have size at least $2^{\Omega(n/\log n)}$. This completes the proof. □

Theorem 4.6 already implies that the BKS Conjecture cannot hold with d always equal to 2. To completely falsify the conjecture, we need the following additional observations:

Proposition 4.7 *Any function $f : \{0,1\}^n \to \{0,1\}$ that .02-approximates $\mathcal{F}_n$ must depend on at least $\Omega(n)$ input bits.*

Proof: It is very well known (see [DF06] for a written proof) that there is an explicit $\epsilon > 0$ (and certainly $\epsilon = .1$ is achievable) such that any function $g : \{0,1\}^n \to \{0,1\}$ that ϵ-approximates $\text{Tribes}_{n/2}$ must depend on at least $\Omega(n)$ of its input bits. Now an argument very similar to the one used in the proof of Theorem 4.6 shows that if f is a .02-approximator for $\mathcal{F}_n$, then some restriction of f must be a δ-approximator for $\text{Tribes}_{n/2}$ with $\delta \leq 4 \cdot .02 < .1$. Since this restriction must depend on at least $\Omega(n/2)$ input bits, we conclude that f must also depend on at least this many input bits. □

Proposition 4.8 *Any circuit that .01-approximates $\mathcal{F}_n$ must have size at least $\Omega(n/\log n)$.*

Proof: Suppose the circuit C has size s and is a .01-approximator for $\mathcal{F}_n$. By Observation 1.1, there is another circuit C' of size at most s and width at most $\log(100s)$ that .01-approximates C; this C' is thus a $.01 + .01 = .02$-approximator for $\mathcal{F}_n$. But C' depends on at most size × width $= s\log(100s)$ literals. Hence $s\log(100s) \geq \Omega(n)$ by Proposition 4.7 and so $s \geq \Omega(n/\log n)$. □

Finally, we've established:

Theorem 4.9 *The BKS Conjecture is false.*

Proof: We use the function $\mathcal{F}_n$, which is monotone and has $\mathbb{I}(\mathcal{F}_n) = O(\log n)$. The BKS Conjecture implies that there is some universal constant $K = K(.01) < \infty$ such that the following holds: There is a circuit C that .01-approximates $\mathcal{F}_n$ and has depth d and size s, for some d and s satisfying

$$s \leq \exp\left((K \cdot \mathbb{I}(\mathcal{F}_n))^{1/(d-1)}\right) = \exp\left(O(\log^{1/(d-1)} n)\right).$$

Now d can't be 2, since this would imply $s \leq \text{poly}(n)$, and we know from Theorem 4.6 that there is no circuit .01-approximating $\mathcal{F}_n$ of depth 2 and size $2^{o(n/\log n)}$. But $d \geq 3$ is also impossible, since this would imply $s \leq \exp(\sqrt{\log n})$, and we know from Proposition 4.8 that there is no circuit .01-approximating $\mathcal{F}_n$ of size $o(n/\log n)$. □

References

[Ajt83] M. Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24:1–48, 1983.

[Ajt93] M. Ajtai. Approximate counting with uniform constant-depth circuits. In *Advances in Computational Complexity Theory*, pages 1–20. Amer. Math. Soc., Providence, RI, 1993.

[AKS83] M. Ajtai, J. Komlós, and E. Szemerédi. Sorting in $c \log n$ parallel steps. *Combinatorica*, 3:1–19, 1983.

[Bea93] P. Beame. A switching lemma primer. Unpublished, 1993.

[BKS99] I. Benjamini, G. Kalai, and O. Schramm. Noise sensitivity of boolean functions and applications to percolation. *Inst. Hautes Études Sci. Publ. Math.*, 90:5–43, 1999.

[BOL90] M. Ben-Or and N. Linial. Collective coin flipping. In S. Micali, editor, *Randomness and Computation.* Academic Press, New York, 1990.

[Bop86] R. Boppana. Threshold functions and bounded depth monotone circuits. *J. Comp. Sys. Sci.*, 32(2):222–229, April 1986.

[Bop97] R. Boppana. The average sensitivity of bounded-depth circuits. *Inf. Process. Lett.*, 63(5):257–261, 1997.

[Bou99] J. Bourgain. 1999. Appendix to [Fri99].

[BT96] N. Bshouty and C. Tamon. On the Fourier spectrum of monotone functions. *Journal of the ACM*, 43(4):747–770, 1996.

[DF06] I. Dinur and E. Friedgut, 2006. Lecture notes, available at `http://www.cs.huji.ac.il/~analyt/scribes/L11.pdf`.

[Fri98] E. Friedgut. Boolean functions with low average sensitivity depend on few coordinates. *Combinatorica*, 18(1):474–483, 1998.

[Fri99] E. Friedgut. Sharp thresholds of graph properties, and the k-SAT problem. *J. American Math. Soc.*, 12(4):1017–1054, 1999.

[Fri05] E. Friedgut. Hunting for sharp thresholds. *Random Struct. & Algorithms*, 26(1-2):37–51, 2005.

[FSS84] M. Furst, J. Saxe, and M. Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[Hås86] J. Håstad. *Computational Limitations for Small Depth Circuits.* MIT Press, Cambridge, MA, 1986.

[HMP06] S. Hoory, A. Magen, and T. Pitassi. Monotone circuits for the majority function. In *RANDOM*, 2006.

[Jac95] J. Jackson. *The Harmonic sieve: a novel application of Fourier analysis to machine learning theory and practice.* PhD thesis, Carnegie Mellon University, August 1995.

[Kal00] G. Kalai. Combinatorics with a geometric flavor: some examples, 2000. GAFA Special Volume 10, Birkhauser Verlag, Basel, 2000.

[KKL88] J. Kahn, G. Kalai, and N. Linial. The influence of variables on boolean functions. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science*, pages 68–80, 1988.

[Kle66] D. Kleitman. Families of non-disjoint sets. *J. Comb. Theory*, 1:153–155, 1966.

[KS05] G. Kalai and S. Safra. *Threshold phenomena and influence.* In *Computational Complexity and Statistical Physics*, Oxford University Press, 2005.

[LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993.

[Mar74] G. Margulis. Probabilistic characteristics of graphs with large connectivity. *Prob. Peredachi Inform.*, 10:101–108, 1974.

[OS06] R. O'Donnell and R. Servedio. Learning monotone decision trees in polynomial time. *SIAM J. Comp.*, 2006. (To appear.).

[PPZ92] M. Paterson, N. Pippenger, and U. Zwick. Optimal carry save networks. In Mike S. Paterson, editor, *Boolean function complexity*, volume 169 of *London Mathematical Society Lecture Note Series*, pages 174–201. Cambridge University Press, Cambridge, 1992.

[Rus78] L. Russo. On the critical percolation probabilities. *Z. Wahrsch. Werw. Gebiete*, 43:39–48, 1978.

[Tal96] M. Talagrand. How much are increasing sets positively correlated? *Combinatorica*, 16(2):243–258, 1996.

[Val84] L. Valiant. Short monotone formulae for the majority function. *J. Algorithms*, 5(3):363–366, September 1984.

[Vio05] E. Viola. On probabilistic time versus alternating time. *ECCC*, (137), 2005.

[Yao83] A. C. Yao. Lower bounds by probabilistic arguments. In *FOCS*, pages 420–428, Tucson, Arizona, 7–9 November 1983. IEEE.

A A Weak Version of Theorem 3.1

Theorem A.1 *If $\epsilon < \frac{1}{2\pi}$, then $\mathbb{I}(f) \geq \Omega(\sqrt{n})$ for all f that ϵ-approximate* Maj_n.

Proof: Suppose f is an ϵ-approximator for Maj_n. By Proposition 3.3, we may assume f is monotone.

It is well known [KKL88] that if f is monotone then $\mathrm{Inf}_i(f) = 2\hat{f}(\{i\})$. Here we are writing $\hat{g}(S) = \mathbf{E}_{\boldsymbol{x}}[g(x) \cdot \prod_{i \in S}(-1)^{\boldsymbol{x}_j}]$ for the $S \subseteq [n]$ Fourier coefficient of g. Let α denote the vector $[\hat{f}(\{1\}), \ldots, \hat{f}(\{n\})] \in \mathbb{R}^n$ and β the vector $[\widehat{\mathrm{Maj}_n}(\{1\}), \ldots, \widehat{\mathrm{Maj}_n}(\{n\})]$. We have $\beta = (c, c, \ldots, c)$, where $c = \frac{1}{2}\mathrm{Inf}_1(f) = 1/\sqrt{2\pi n} + o(1/\sqrt{n})$ (see Proposition 1.7). Now

$$\|\alpha - \beta\|^2 = \sum_{i=1}^{n} (\hat{f}(\{i\}) - \widehat{\mathrm{Maj}_n}(\{i\}))^2 \leq \sum_{S \subseteq [n]} (\hat{f}(S) - \widehat{\mathrm{Maj}_n}(S))^2 = \mathbf{E}_{\boldsymbol{x}}[(f(\boldsymbol{x}) - \mathrm{Maj}_n(\boldsymbol{x}))^2] \leq \epsilon,$$

where the second-to-last equality is Parseval's Theorem and the last equality uses the fact that f is an ϵ-approximator for Maj_n. We would now like to lower-bound $\langle \alpha, \beta \rangle$ subject to this constraint $\|\beta - \alpha\| \le \sqrt{\epsilon}$, because

$$\langle \alpha, \beta \rangle = c \cdot \sum_{i=1}^{n} \alpha_i = c \cdot \frac{1}{2} \sum_{i=1}^{n} \mathrm{Inf}_i(f) = (c/2) \cdot \mathbb{I}(f).$$

Now if $\|\beta\| \le \sqrt{\epsilon}$ then we can't give any lower bound, since α could potentially be 0. This is why we require the assumption $\epsilon < \frac{1}{2\pi} < \frac{1}{2\pi} + o(1) = \|\beta\|$. Assuming this, it's clear that subject to $\|\beta - \alpha\| \le \sqrt{\epsilon}$ the minimum value of $\langle \alpha, \beta \rangle$ occurs when α points in the direction of β, and equals $(1 - \sqrt{\epsilon}/\|\beta\|)\beta$. In this case, $\langle \alpha, \beta \rangle = \|\beta\|^2 - \sqrt{\epsilon}\|\beta\|$, and so we get

$$(c/2) \cdot \mathbb{I}(f) = \langle \alpha, \beta \rangle \ge \|\beta\|^2 - \sqrt{\epsilon}\|\beta\| = c^2 n - c\sqrt{\epsilon n}$$

which implies

$$\mathbb{I}(f) \ge 2cn - 2\sqrt{\epsilon n} = \mathbb{I}(\mathrm{Maj}_n) - 2\sqrt{\epsilon n}.$$

Since $\epsilon < \frac{1}{2\pi}$ and $\mathbb{I}(\mathrm{Maj}_n) \ge \sqrt{2\pi n}$, we conclude that $\mathbb{I}(f) \ge \Omega(\sqrt{n})$, as claimed. From this we also see that $\mathbb{I}(f) \ge (1 - O(\sqrt{\epsilon}))\mathbb{I}(\mathrm{Maj}_n)$ for small ϵ. □

B Why Switching Lemmas Don't Seem to Help

It might seem that switching lemmas could give a size lower bound for DNFs that ϵ-approximate Majority; certainly these tools have been extremely useful for proving size lower bounds on constant-depth circuits that *exactly* compute Majority (see [FSS84, Ajt83, Hås86], and [Yao83, Bop86] for the monotone case). However even Håstad's Switching Lemma doesn't seem to help. Briefly, suppose one tries to argue against $O(\sqrt{n})$-width DNF by using random restrictions with $\Theta(\sqrt{n})$ many unset variables. With high probability the imbalance of 0's and 1's in the restriction will be $\Omega(\sqrt{n})$; this will make the restricted Majority function essentially constant, because the imbalance in the unset variables is extremely unlikely to exceed $O(n^{1/4})$. Thus there is no contradiction when the approximating DNF is devastated. One could also consider using the variation due to Beame [Bea93] in which the random restrictions have an equal number of 0's and 1's. Then the restricted Majority function remains a Majority, but the restrictions themselves, even when completed randomly, only constitute a negligible fraction of the probability space on $\{0,1\}^n$. Perhaps the best one can do is use the usual random restrictions to show that width $\Omega(\sqrt{n})$ DNF are required to compute a "$1/2 + \Omega(1/\sqrt{n})$ approximate-Majority" — i.e., to compute Majority correctly on all strings that have at least a $1/2+\Omega(1/\sqrt{n})$ fraction of 1's or at most a $1/2-\Omega(1/\sqrt{n})$ fraction of 1's (Viola [Vio05] appears to have made this observation). However this only rules out one very particular way of being an $\Omega(1)$-approximator in our sense.