THEORY OF COMPUTING, Volume 2 (2006), pp. 147–172
http://theoryofcomputing.org

On Learning Random DNF Formulas Under the Uniform Distribution

Jeffrey C. Jackson Rocco A. Servedio

Received: March 21, 2006; published: September 19, 2006.

Abstract: We study the average-case learnability of DNF formulas in the model of learning from uniformly distributed random examples. We define a natural model of random monotone DNF formulas and give an efficient algorithm which with high probability can learn, for any fixed constant $\gamma > 0$, a random t-term monotone DNF for any $t = O(n^{2-\gamma})$. We also define a model of random non-monotone DNF and give an efficient algorithm which with high probability can learn a random t-term DNF for any $t = O(n^{3/2-\gamma})$. These are the first known algorithms that can learn a broad class of polynomial-size DNF in a reasonable average-case model of learning from random examples.

ACM Classification: I.2.6, F.2.2, G.1.2, G.3

AMS Classification: 68Q32, 68W20, 68W25, 60C05

Key words and phrases: computational learning theory, uniform-distribution learning, PAC learning, DNF formulas, monotone DNF

1 Introduction

1.1 Motivation and background

A *disjunctive normal form* formula, or DNF, is an AND of ORs of Boolean literals. A question that has been open since Valiant's initial paper on computational learning theory [26] is whether or not efficient algorithms exist for learning polynomial size DNF formulas in variants of the PAC (Probably Approximately Correct) learning model introduced by Valiant. Roughly speaking, in these models a learning

DOI: 10.4086/toc.2006.v002a008

algorithm is required to generate a high-accuracy (error rate at most ε) hypothesis with high probability (the algorithm must fail to generate such a hypothesis with probability at most δ); we give a more detailed explanation of our learning scenario in Section 2. The only positive result for learning general DNF formulas in such frameworks to date is the Harmonic Sieve [12]. The Sieve is a membership-query algorithm (i.e. it requires black-box query access to the unknown function f that is being learned) that efficiently PAC learns DNF when the error rate is defined with respect to the uniform distribution over the space of all possible n-bit example strings (and certain related distributions). The approximating hypothesis produced by the Sieve is not itself represented as a DNF; thus, the Sieve is an *improper* learning algorithm.

There has been little progress on polynomial-time algorithms for learning arbitrary DNF since the discovery of the Sieve. There are two obvious relaxations of the uniform distribution membership query model that can be pursued. The first is to learn with respect to arbitrary distributions using membership queries; in this setting, the learning algorithm is given black-box (membership query) access to the unknown function f, and also access to a source of random labeled examples $(x, f(x))$ where each example x is independently drawn from a fixed probability distribution which is arbitrary and not known to the learning algorithm. The learner must generate a high-accuracy hypothesis with respect to this unknown distribution. Given standard cryptographic assumptions, it is known that learning DNF in this framework is essentially as difficult as learning DNF with respect to arbitrary distributions without membership queries [4].

The second obvious relaxation is to learn with respect to the uniform distribution without membership queries. However, there are substantial known obstacles to learning DNF in the model of uniform distribution without membership queries. In particular, no algorithm which can be recast as a Statistical Query algorithm can learn arbitrary polynomial-size DNF under the uniform distribution in $n^{o(\log n)}$ time [8]. (Roughly speaking, a Statistical Query algorithm is an algorithm which is only allowed to obtain statistical estimates of properties of the distribution over labeled example pairs $(x, f(x))$; such an algorithm does not have access to actual labeled examples $(x, f(x))$. See [17] for a detailed description of the Statistical Query model.) Since nearly all non-membership learning algorithms can be recast as Statistical Query algorithms [17], a major conceptual shift seems necessary to obtain an algorithm for efficiently learning arbitrary DNF formulas from uniform examples alone.

An apparently simpler question is whether *monotone* DNF formulas, which contain only un-negated variables, can be learned efficiently. Angluin showed that monotone DNF can be properly learned with respect to arbitrary distributions using membership queries [3]. It has also long been known that with respect to arbitrary distributions without membership queries, monotone DNF are no easier to learn than arbitrary DNF [19]. This leaves the following enticing question (posed in [16, 7, 6]): are monotone DNF efficiently learnable from uniform examples alone?

In 1990, Verbeurgt [27] gave an algorithm that can properly learn any $\mathrm{poly}(n)$-size (arbitrary) DNF from uniform examples in time $n^{O(\log n)}$. More recently, the algorithm of [25] learns any $2^{\sqrt{\log n}}$-term monotone DNF in $\mathrm{poly}(n)$ time. However, despite significant interest in the problem, no algorithm faster than that of [27] is known for learning arbitrary $\mathrm{poly}(n)$-size monotone DNF from uniform examples, and no known hardness result precludes such an algorithm (the Statistical Query result of [8] is at its heart a hardness result for low-degree parity functions, and thus does not apply to monotone DNF).

Since worst-case versions of several DNF learning problems have remained stubbornly open for a

decade or more, it is natural to ask about DNF learning from an average-case perspective, i.e., about learning *random* DNF formulas. In fact, this question has been considered before: Aizenstein and Pitt [1] were the first to ask whether random DNF formulas are efficiently learnable. They proposed a model of random DNF in which each of the t terms is selected independently at random from all possible terms, and gave a membership and equivalence query algorithm which with high probability learns a random DNF generated in this way. (See [1] or [3] for a description of the membership and equivalence query learning framework.) However, as noted in [1], a limitation of this model is that with very high probability all terms will have length $\Omega(n)$. The learning algorithm itself becomes quite simple in this situation. Thus, while this is a "natural" average-case DNF model, from a learning perspective it is not a particularly interesting model. To address this deficiency, they also proposed another natural average-case model which is parameterized by the expected length k of each term as well as the number of independent terms t, but left open the question of whether or not random DNF can be efficiently learned in such a model.

1.2 Our results

We consider an average-case DNF model very similar to the latter Aizenstein and Pitt model, although we simplify slightly by assuming that k represents a fixed term length rather than an expected length. We show that, in the model of learning from uniform random examples only, random monotone DNF are properly and efficiently learnable for many interesting values of k and t. In particular, for $t = O(n^{2-\gamma})$ where $\gamma > 0$, and for $k = \log t$, our algorithm can achieve any error rate $\varepsilon > 0$ in $\mathrm{poly}(n, 1/\varepsilon)$ time with high probability (over both the selection of the target DNF and the selection of examples). In addition, we obtain slightly weaker results for arbitrary DNF: our algorithm can properly and efficiently learn random t-term DNF for t such that $t = O(n^{\frac{3}{2}-\gamma})$. This algorithm cannot achieve arbitrarily small error but can achieve error $\varepsilon = o(1)$ for any $t = \omega(1)$. For detailed result statements see Theorems 3.13 and 4.11.

While our results would clearly be stronger if they held for any $t = \mathrm{poly}(n)$ rather than the specific polynomials given, they are a marked advance over the previous state of affairs for DNF learning. (Recall that in the standard worst-case model, $\mathrm{poly}(n)$-time uniform-distribution learning of $t(n)$-term DNF for any $t(n) = \omega(1)$ is an open problem with an associated cash prize [5].)

At this point a word or two is in order to clarify the relationship between the random DNF model we consider and the models of random CNF formulas that are often studied in the context of the Boolean satisfiability problem. In the study of random k-CNF formulas, k is often taken to be a fixed constant such as 3. In contrast with the satisfiability problem, in the learning arena taking k to be a fixed constant such as 3 is not an interesting choice, since it is well known that k-CNF (or equivalently, DNF formulas in which every term is of length at most k) can be easily learned with respect to any distribution in time $n^{O(k)}$ [26]. Intuitively, the "interesting" values of k are different for the satisfiability problem and the learning problem because in the satisfiability problem the interesting cases occur when there are only a small number of satisfying assignments, whereas in the learning framework the interesting cases occur when the target DNFs are roughly balanced between satisfying and unsatisfying assignments. (From a learning perspective balanced functions are generally more interesting than unbalanced functions, since a constant function is trivially a good approximator to a highly unbalanced function.) Thus, for the learning problem, taking $k = \log t$ is a natural choice when learning with respect to the uniform

distribution. (We actually allow a somewhat more general choice of k, as is described in detail in the paper.)

Our results shed some light on which cases are *not* hard to learn in the worst-case uniform distribution model. While "hard" cases were previously known for arbitrary DNF [5], our findings may be particularly helpful in guiding future research on monotone DNF. In particular, our algorithm learns any monotone $t = O(n^{2-\gamma})$-term DNF which (i) is near-balanced, (ii) has every term uniquely satisfied with reasonably high probability, (iii) has every pair of terms jointly satisfied with much smaller probability, and (iv) has no variable appearing in significantly more than a $1/\sqrt{t}$ fraction of the t terms (this is made precise in Lemma 3.9). So in order to be "hard," a monotone DNF must violate one or more of these criteria.

Our algorithms work in two stages: they first identify pairs of variables which co-occur in some term of the target DNF, and then use these pairs to reconstruct terms via a specialized clique-finding algorithm. (This is why our results do not extend to random DNF with more than $n^{2-\gamma}$ terms; for such formulas the variable co-occurrence graph is with high probability dense or even complete, so we cannot reconstruct terms from co-occurrence information.) For monotone DNF we can with high probability determine for every pair of variables whether or not the pair co-occurs in some term. For non-monotone DNF, with high probability we can identify most pairs of variables which co-occur in some term; as we show, this enables us to learn to fairly (but not arbitrarily) high accuracy.

We give preliminaries in Section 2. Sections 3 and 4 contain our results for monotone and non-monotone DNF respectively. Section 5 concludes.

A preliminary version of this work appeared in the proceedings of RANDOM 2005 [15]. The current version of the paper gives a more thorough exposition and includes many proofs that were omitted from the conference version due to space limitations.

2 Preliminaries

We first describe our models of random monotone and non-monotone DNF. Let $\mathcal{M}_n^{t,k}$ be the probability distribution over monotone t-term DNF induced by the following random process: each term is independently and uniformly chosen at random from all $\binom{n}{k}$ monotone ANDs of size exactly k over variables $v_1, \ldots, v_n$. For non-monotone DNF, we write $\mathcal{D}_n^{t,k}$ to denote the following distribution over t-term DNF: first a monotone DNF is selected from $\mathcal{M}_n^{t,k}$, and then each occurrence of each variable in each term is independently negated with probability $1/2$. (Equivalently, a draw from $\mathcal{D}_n^{t,k}$ is done by independently selecting t terms from the set of all terms of length exactly k.)

Given a Boolean function $\phi : \{0,1\}^n \to \{0,1\}$, we write $\Pr[\phi]$ to denote $\Pr_{x \sim U_n}[\phi(x) = 1]$, where U_n denotes the uniform distribution over $\{0,1\}^n$. We write log to denote $\log_2$ and ln to denote natural log.

2.1 Tail bounds

We use the following:

Chernoff bound (see [2, Theorem A.12]): Let $B(t,p)$ denote the binomial distribution with parameter

p, i.e. a draw from $B(t,p)$ is a sum of t independent p-biased 0/1 Bernoulli trials. Then for $\beta > 1$,

$$\Pr_{S\sim B(t,p)}[S \geq \beta pt] \leq \left(e^{\beta-1}\beta^{-\beta}\right)^{pt} < (e/\beta)^{\beta pt} \quad .$$

The following bound will also be useful:

McDiarmid bound [24]: *Let $X_1,\ldots X_m$ be independent random variables taking values in a set Ω. Let $F : \Omega^m \to \mathbb{R}$ be such that for all $i \in [m]$ we have*

$$|F(x_1,\ldots,x_m) - F(x_1,\ldots,x_{i-1},x_i',x_{i+1},\ldots,x_m)| \leq c_i$$

for all $x_1,\ldots,x_m$ and x_i' in Ω. Let $\mu = \mathbf{E}[F(X_1,\ldots,X_m)]$. Then for all $\tau > 0$,

$$\Pr[|F(X_1,\ldots,X_m) - \mu| > \tau] < \exp(-\tau^2/(c_1^2+\cdots+c_m^2)) \quad .$$

2.2 The learning model

In the uniform distribution learning model which we consider, the learner is given a source of labeled examples $(x, f(x))$ where each x is uniformly drawn from $\{0,1\}^n$ and f is the unknown function to be learned. The goal of the learner is to efficiently construct a hypothesis h which with high probability (over the choice of labeled examples used for learning) has low error relative to f under the uniform distribution, i.e. $\Pr_{x\sim U_n}[h(x) \neq f(x)] \leq \varepsilon$ with probability $1-\delta$. This model has been intensively studied in learning theory, see e.g. [11, 10, 13, 21, 22, 25, 27]. In our average case framework, the target function f will be drawn randomly from either $\mathcal{M}_n^{t,k}$ or $\mathcal{D}_n^{t,k}$, and (as in [14]) our goal is to construct a low-error hypothesis h for f with high probability over both the random examples used for learning and the random draw of f.

3 Learning random monotone DNF

3.1 Interesting parameter settings

Consider a random draw of f from $\mathcal{M}_n^{t,k}$. It is intuitively clear that if t is too large relative to k then a random $f \in \mathcal{M}_n^{t,k}$ will likely have $\Pr[f] \approx 1$; similarly if t is too small relative to k then a random $f \in \mathcal{M}_n^{t,k}$ will likely have $\Pr[f] \approx 0$. Such cases are not very interesting from a learning perspective since a trivial algorithm can learn to high accuracy. We are thus led to the following definition:

Definition 3.1. A pair of values (k,t) is said to be *monotone α-interesting* if

$$\alpha \leq \mathbf{E}_{f\in\mathcal{M}_n^{t,k}}[\Pr[f]] \leq 1-\alpha \quad .$$

Throughout the paper we will assume that $0 < \alpha \leq .09$ is a fixed constant independent of n and that $t \leq p(n)$, where $p(\cdot)$ is a fixed polynomial (and we will also make assumptions about the degree of p). The following lemma gives necessary conditions for (k,t) to be monotone α-interesting. (As Lemma 3.2 indicates, we may always think of k as being roughly $\log t$.)

Lemma 3.2. *For n sufficiently large, if (k,t) is monotone α-interesting then $\alpha 2^k \leq t \leq 2^{k+1} \ln \frac{2}{\alpha}$.*

Proof. One side is easy: if $t < \alpha 2^k$ then each of the t terms of f is satisfied by a uniform random example with probability at most α/t, and consequently $\Pr[f(x) = 1] \leq \alpha$. Note that by our assumptions on t and α we thus have that $k = O(\log n)$ for any monotone α-interesting pair (k,t).

We now show that if $t > 2^{k+1} \log \frac{2}{\alpha}$, then

$$\mathbf{E}_{f \in \mathcal{M}_n^{t,k}}[\Pr[f]] > 1 - \alpha \ .$$

Let us write $|x|$ to denote $x_1 + \cdots + x_n$ for $x \in \{0,1\}^n$. It is easy to see that $\Pr[f(x) = 1]$, viewed as a random variable over the choice of $f \in \mathcal{M}_n^{t,k}$, depends only on the value of $|x|$. We have

$$\mathbf{E}_{f \in \mathcal{M}_n^{t,k}}[\Pr[f]] = \sum_{r=0}^{n} \mathbf{E}_{f \in \mathcal{M}_n^{t,k}}[\Pr[f(x) = 1 \mid |x| = r] \cdot \Pr[|x| = r] \ .$$

A standard tail bound on the binomial distribution (which can be derived, e.g., from the results in [24]) implies that

$$\Pr_{x \in U_n}\left[|x| \leq n/2 - \sqrt{n \log(2/\alpha)}\right] < \alpha/2 \ .$$

Thus it suffices to show that for any x with $|x| \geq n/2 - \sqrt{n \log(2/\alpha)}$, we have

$$\Pr_{f \in \mathcal{M}_n^{t,k}}[f(x) = 1] \geq 1 - \alpha/2 \ .$$

Fix an $x \in \{0,1\}^n$ with $|x| = w \geq n/2 - \sqrt{n \log(2/\alpha)}$. Let T_1 be a random monotone term of length k. We have

$$\Pr_{T_1}[T_1(x) = 1] = \frac{w(w-1)\cdots(w-k+1)}{n(n-1)\cdots(n-k+1)} \geq \frac{1}{2^{k+1}}$$

where the inequality holds for sufficiently large n using the fact that $k = O(\log n)$ and $\alpha = \Theta(1)$. Since the terms of f are chosen independently, this implies that

$$\Pr_f[f(x) = 0] \leq \left(1 - \frac{1}{2^{k+1}}\right)^t \leq \exp\left(\frac{-t}{2^{k+1}}\right) \ .$$

If $t/2^{k+1} > \ln \frac{2}{\alpha}$ then this bound is at most $\alpha/2$. □

DNF expressions with either a constant number of terms or a constant number of variables per term have long been known to be efficiently learnable [26] (this holds for non-monotone as well as monotone DNF, and in fact holds for learning with respect to arbitrary distributions, not only uniform). So we will assume throughout that both t and k are $\omega(1)$; many of our probability bounds are only meaningful given such an assumption, and some of our lemmas explicitly depend on t and/or k being larger than a certain (small) constant. While this assumption is sufficient for our purposes, we note briefly that in fact a stronger assumption can be made concerning t. If t grows very slowly relative to n, say, $t = O(n^{1/4})$, then with high probability a random f drawn from $\mathcal{M}_n^{t,k}$ will have the property that every variable in f appears in exactly one term. Such a read-once DNF, even if it is non-monotone, is learnable with respect to the uniform distribution [18]. Thus, we can actually think of t as growing reasonably quickly with n.

3.2 Properties of random monotone DNF

Throughout the rest of Section 3 we assume that $\alpha > 0$ is fixed and (k,t) is a monotone α-interesting pair where $t = O(n^{2-\gamma})$ for some $\gamma > 0$. In this section we develop some useful lemmas regarding $\mathcal{M}_n^{t,k}$.

We first prove the following lemma, which will be useful in subsequent proofs. This lemma does not require that f be drawn from $\mathcal{M}_n^{t,k}$.

Lemma 3.3. *Any monotone DNF f with $t \geq 2$ terms each of size k has $\Pr[\overline{f}] \geq \alpha^3/4$.*

Proof. We write $T_1, T_2, \ldots, T_t$ to denote the terms of f. We have

$$\begin{aligned}\Pr[\overline{f}] &= \Pr[\overline{T_1} \wedge \overline{T_2} \wedge \cdots \wedge \overline{T_t}] = \Pr[\overline{T_1} \mid \overline{T_2} \wedge \cdots \overline{T_t}] \Pr[\overline{T_2} \mid \overline{T_3} \wedge \cdots \overline{T_t}] \cdots \Pr[\overline{T_{t-1}} \mid \overline{T_t}] \Pr[\overline{T_t}] \\ &\geq \prod_{i=1}^{t} \Pr[\overline{T_i}] \qquad (3.1) \\ &= \left(1 - \frac{1}{2^k}\right)^t \geq \left(1 - \frac{1}{2^k}\right)^{2^{k+1}\ln(2/\alpha)} \geq \left(\frac{1}{4}\right)^{2\ln\frac{2}{\alpha}} = \left(\frac{\alpha}{2}\right)^{2\ln 4} \geq \frac{\alpha^3}{4} \ .\end{aligned}$$

The first inequality (3.1) holds since $\Pr[f(x) = 1 \mid g(x) = 1] \geq \Pr[f(x) = 1]$ for any monotone Boolean functions f, g on $\{0,1\}^n$ (see e.g. Corollary 7, p. 149 of [9]). The second inequality holds by Lemma 3.2. The third inequality holds since $(1 - 1/x)^x \geq 1/4$ for all $x \geq 2$, and the fourth follows from the restriction $\alpha \leq .09$. □

Let f^i denote the projected function obtained from f by first removing term T_i from the monotone DNF for f and then restricting all of the variables which were present in term T_i to 1. For $\ell \neq i$ we write T_ℓ^i to denote the term obtained by setting all variables in T_i to 1 in T_ℓ, i.e. T_ℓ^i is the term in f^i corresponding to T_ℓ. Note that if $T_\ell^i \not\equiv T_\ell$ then T_ℓ^i is smaller than T_ℓ.

The following lemma shows that each variable appears in a limited number of terms and that therefore not too many terms T_ℓ^i in f^i are smaller than their corresponding terms T_ℓ in f. In this and later lemmas, "n sufficiently large" means that n is larger than a constant which depends on α but not on k or t.

Lemma 3.4. *Let*

$$\delta_{\text{many}} := n\left(\frac{ekt^{3/2}\log t}{n2^{k-1}\alpha^2}\right)^{2^{k-1}\alpha^2/(\sqrt{t}\log t)} \ .$$

For n sufficiently large, with probability at least $1 - \delta_{\text{many}}$ over the draw of f from $\mathcal{M}_n^{t,k}$, both of the following conditions hold:

- *Every variable v_j, $1 \leq j \leq n$, appears in at most $2^{k-1}\alpha^2/(\sqrt{t}\log t)$ terms of f; and*
- *For all $1 \leq i \leq t$ at most $k2^{k-1}\alpha^2/(\sqrt{t}\log t)$ terms T_ℓ^i with $\ell \neq i$ in the projection f^i are smaller than the corresponding terms T_ℓ in f.*

Note that since (k,t) is a monotone α-interesting pair and $t = O(n^{2-\gamma})$ for some fixed $\gamma > 0$, for sufficiently large n this probability bound is non-trivial.

Proof of Lemma 3.4. We first prove that with high probability every variable appears in a limited number of terms. Fix any variable v_j. For each term T_ℓ we have that v_j occurs in T_ℓ with probability k/n. Since the terms are chosen independently, the number of occurrences of v_j is binomially distributed according to $B(t,p)$ with $p = k/n$. Taking $\beta = n2^{k-1}\alpha^2/(kt^{3/2}\log t)$ in the Chernoff bound (which is greater than 1 for sufficiently large n), the probability that v_j appears in $\beta pt = 2^{k-1}\alpha^2/(\sqrt{t}\log t)$ or more terms is at most

$$\left(\frac{ekt^{3/2}\log t}{n2^{k-1}\alpha^2}\right)^{2^{k-1}\alpha^2/(\sqrt{t}\log t)} .$$

The lemma follows by the union bound over the n variables v_j.

For the bound on the number of terms in f^i smaller than those in f, simply note that if every variable appears in at most $2^{k-1}\alpha^2/(\sqrt{t}\log t)$ terms then, since there are k variables in term T_i, at most $k2^{k-1}\alpha^2/(\sqrt{t}\log t)$ terms T^i_ℓ with $\ell \neq i$ in f^i are smaller than the corresponding terms T_ℓ in f. □

The next lemma shows that there is probably little overlap between any pair of terms in f:

Lemma 3.5. *Let $\delta_{\text{shared}} := t^2(\frac{k^2}{n})^{\log\log t}$. With probability at least $1-\delta_{\text{shared}}$ over the random draw of f from $\mathcal{M}_n^{t,k}$, for all $1 \le i,j \le t$ no set of $\log\log t$ or more variables belongs to two distinct terms T_i and T_j in f.*

Proof. We are interested in upper bounding the probability p_i that $\log\log t$ or more of the variables in a fixed term T_i belonging to f also appear in some other term T_ℓ of f, for any $\ell \neq i$. First, a simple counting argument shows that the probability that a fixed set of $\log\log t$ variables appears in a set of k variables randomly chosen from among n variables is at most $(k/n)^{\log\log t}$. Since there are $\binom{k}{\log\log t}$ ways to choose a fixed set of $\log\log t$ variables from term T_i, we have

$$p_i \le \binom{k}{\log\log t}\left(\frac{k}{n}\right)^{\log\log t}(t-1) .$$

The lemma follows by the union bound over the t probabilities p_i. □

Using the preceding lemmas, we can show that for f drawn from $\mathcal{M}_n^{t,k}$, with high probability each term is "uniquely satisfied" by a noticeable fraction of assignments. More precisely, we have:

Lemma 3.6. *Let $\delta_{\text{usat}} := \delta_{\text{many}} + \delta_{\text{shared}}$. For n sufficiently large and $k \ge 5$, with probability at least $1-\delta_{\text{usat}}$ over the random draw of f from $\mathcal{M}_n^{t,k}$, f is such that for all $i = 1,\ldots,t$ we have*

$$\Pr_x[T_i \text{ is satisfied by } x \text{ but no other } T_j \text{ is satisfied by } x] \ge \frac{\alpha^3}{2^{k+3}} .$$

Proof. Given an f drawn according to $\mathcal{M}_n^{t,k}$ and given any term T_i in f, we are interested in the probability over uniformly drawn instances that T_i is satisfied and T_ℓ is not satisfied for all $\ell \neq i$. Let $\overline{T_{\ell\neq i}}$ represent the formula that is satisfied by an assignment x if and only if all of the T_ℓ with $\ell \neq i$ are not satisfied by x. We want a lower bound on

$$\Pr[T_i \wedge \overline{T_{\ell\neq i}}] = \Pr[\overline{T_{\ell\neq i}} \mid T_i]\cdot\Pr[T_i] .$$

Since $\Pr[T_i] = 1/2^k$, what remains is to show that with very high probability over random draw of f, $\Pr[\overline{T_{\ell \neq i}} \mid T_i]$ is bounded below by $\alpha^3/8$ for all T_i. That is, we need to show that $\Pr[\overline{f^i}] \geq \alpha^3/8$ with very high probability.

We have that all of the following statements hold with probability at least $1 - \delta_{\text{usat}}$ for every $1 \leq i \leq n$ for a random f from $\mathcal{M}_n^{t,k}$:

1. $\Pr[\overline{f^i}] \geq \prod_{\ell:\ell \neq i} \Pr[\overline{T_\ell^i}]$: this follows from Equation (3.1) in the proof of Lemma 3.3.

2. $\prod_{\ell:T_\ell^i \equiv T_\ell} \Pr[\overline{T_\ell^i}] > \alpha^3/4$. This holds because the terms in this product are a subset of the terms in Equation (3.1) (in the proof of Lemma 3.3).

3. At most $k2^{k-1}\alpha^2/(\sqrt{t}\log t)$ terms T_ℓ with $\ell \neq i$ are smaller in f^i than they are in f (by Lemma 3.4).

4. No term in f^i has fewer than $k - \log\log t$ variables (by Lemma 3.5).

These conditions together imply that

$$\Pr[\overline{f^i}] \geq \left(\frac{\alpha^3}{4}\right)\left(\left(1 - \frac{\log t}{2^k}\right)^{2^k/(\log t)}\right)^{k\alpha^2/2\sqrt{t}}.$$

Note that $k\alpha^2/2\sqrt{t} \leq 1/2$ for all $k \geq 5$, since for such k we have $k^2\alpha^4 \leq \alpha k^2 < \alpha 2^k \leq t$. Thus, since $\left(1 - \frac{1}{x}\right)^x \geq 1/4$ for all $x \geq 2$, we have that $\Pr[\overline{f^i}] \geq \alpha^3/8$. □

On the other hand, we can upper bound the probability that two terms of a random DNF f will be satisfied simultaneously:

Lemma 3.7. *With probability at least* $1 - \delta_{\text{shared}}$ *over the random draw of* f *from* $\mathcal{M}_n^{t,k}$, *for all* $1 \leq i < j \leq t$ *we have* $\Pr[T_i \wedge T_j] \leq \frac{\log t}{2^{2k}}$.

Proof. By Lemma 3.5, with probability at least $1 - \delta_{\text{shared}}$ f is such that, for all $1 \leq i < j \leq n$, terms T_i and T_j share at most $\log\log t$ variables. Thus for each pair of terms a specific set of at least $2k - \log\log t$ variables must be simultaneously set to 1 in an instance in order for both terms to be satisfied. □

3.3 Identifying co-occurring variables

We now show how to identify pairs of variables that co-occur in some term of f. First, some notation. Given a monotone DNF f over variables $v_1, \ldots, v_n$, define DNF formulas g_{**}, g_{1*}, g_{*1}, and g_{11} over variables $v_3, \ldots, v_n$ as follows:

- g_{**} is the disjunction of the terms in f that contain neither v_1 nor v_2;
- g_{1*} is the disjunction of the terms in f that contain v_1 but not v_2 (but with v_1 removed from each of these terms);
- g_{*1} is defined similarly as the disjunction of the terms in f that contain v_2 but not v_1 (but with v_2 removed from each of these terms);

- g_{11} is the disjunction of the terms in f that contain both v_1 and v_2 (with both variables removed from each term).

We thus have $f = g_{**} \vee (v_1 g_{1*}) \vee (v_2 g_{*1}) \vee (v_1 v_2 g_{11})$. Note that any of $g_{**}, g_{1*}, g_{*1}, g_{11}$ may be an empty disjunction which is identically false.

We can empirically estimate each of the following using uniform random examples $(x, f(x))$:

$$\begin{aligned} p_{00} &:= \Pr_x[g_{**}] = \Pr_{x \in U_n}[f(x) = 1 \mid x_1 = x_2 = 0] \\ p_{01} &:= \Pr_x[g_{**} \vee g_{*1}] = \Pr_{x \in U_n}[f(x) = 1 \mid x_1 = 0, x_2 = 1] \\ p_{10} &:= \Pr_x[g_{**} \vee g_{1*}] = \Pr_{x \in U_n}[f(x) = 1 \mid x_1 = 1, x_2 = 0] \\ p_{11} &:= \Pr_x[g_{**} \vee g_{*1} \vee g_{1*} \vee g_{11}] = \Pr_{x \in U_n}[f(x) = 1 \mid x_1 = 1, x_2 = 1] \ . \end{aligned}$$

It is clear that g_{11} is nonempty if and only if v_1 and v_2 co-occur in some term of f; thus we would ideally like to obtain $\Pr_{x \in U_n}[g_{11}]$. While we cannot obtain this probability from p_{00}, p_{01}, p_{10}, and p_{11}, the following lemma shows that we can estimate a related quantity:

Lemma 3.8. *Let P denote $p_{11} - p_{10} - p_{01} + p_{00}$. Then $P = \Pr[g_{11} \wedge \overline{g}_{1*} \wedge \overline{g}_{*1} \wedge \overline{g}_{**}] - \Pr[g_{1*} \wedge g_{*1} \wedge \overline{g}_{**}]$.*

Proof. P gets a net contribution of 0 from those x which belong to $g_{*,*}$ (since each such x is added twice and subtracted twice in P). We proceed to analyze the contributions to P from the remaining 8 subsets of the events g_{11}, g_{1*}, and g_{*1}:

- P gets a net contribution of 0 from those x which are in $g_{1*} \wedge \overline{g}_{*1} \wedge \overline{g}_{**}$ since each such x is counted in p_{11} and p_{10} but not in p_{01} or p_{00}. Similarly P gets a net contribution of 0 from those x which are in $g_{*1} \wedge \overline{g}_{1*} \wedge \overline{g}_{**}$.
- P gets a net contribution of $\Pr[g_{11} \wedge \overline{g}_{1*} \wedge \overline{g}_{*1} \wedge \overline{g}_{**}]$ since each such x is counted in p_{11}.
- P gets a net contribution of $-\Pr[g_{1*} \wedge g_{*1} \wedge \overline{g}_{**}]$ since each such x is counted in p_{01}, p_{10}, and p_{11}.

□

More generally, let P_{ij} be defined as P but with v_i, x_i, v_j, and x_j substituted for v_1, x_1, v_2, and x_2, respectively, throughout the definitions of the g's and p's above. The reader familiar with Boolean Fourier analysis will readily recognize that P_{ij} is a scaled (by a factor of -2) version of the second-order Fourier coefficient of f corresponding to the pair of variables (v_i, v_j). (This coefficient is equal to $2\Pr_{x \in U_n}[f(x) = x_i \oplus x_j] - 1$; see [23] for a nice overview of Boolean Fourier analysis in the context of uniform-distribution learning.) The following lemma shows that, for most random choices of f, for all $1 \le i, j \le n$, the value of P_{ij} is a good indicator of whether or not v_i and v_j co-occur in some term of f:

Lemma 3.9. *For n sufficiently large and $t \ge 16$, with probability at least $1 - \delta_{\text{usat}} - \delta_{\text{shared}} - \delta_{\text{many}}$ over the random draw of f from $\mathcal{M}_n^{t,k}$, we have that for all $1 \le i, j \le n$ (i) if v_i and v_j do not co-occur in some term of f then $P_{ij} \le 0$; (ii) if v_i and v_j do co-occur in some term of f then $P_{ij} \ge \alpha^4/16t$.*

Proof. Part (i) holds for any monotone DNF by Lemma 3.8. For (ii), we first note that with probability at least $1-\delta_{\text{usat}}-\delta_{\text{shared}}-\delta_{\text{many}}$, a randomly chosen f has all the following properties:

1. Each term in f is uniquely satisfied with probability at least $\alpha^3/2^{k+3}$ (by Lemma 3.6);
2. Each pair of terms T_i and T_j in f are both satisfied with probability at most $\log t/2^{2k}$ (by Lemma 3.7); and
3. Each variable in f appears in at most $2^{k-1}\alpha^2/(\sqrt{t}\log t)$ terms (by Lemma 3.4).

We call such an f *well-behaved*. For the sequel, assume that f is well-behaved and also assume without loss of generality that $i=1$ and $j=2$. We consider separately the two probabilities

$$\rho_1 = \Pr[g_{11} \wedge \overline{g}_{1*} \wedge \overline{g}_{*1} \wedge \overline{g}_{**}] \qquad \text{and} \qquad \rho_2 = \Pr[g_{1*} \wedge g_{*1} \wedge \overline{g}_{**}]$$

whose difference defines $P_{12} = P$. By property (1) above, $\rho_1 \geq \alpha^3/2^{k+3}$, since each instance x that uniquely satisfies a term T_j in f containing both v_1 and v_2 also satisfies g_{11} while falsifying all of g_{1*}, g_{*1}, and g_{**}. Since (k,t) is monotone α-interesting, this implies that $\rho_1 \geq \alpha^4/8t$. On the other hand, clearly $\rho_2 \leq \Pr[g_{1*} \wedge g_{*1}]$. By property (2) above, for any pair of terms consisting of one term from g_{1*} and the other from g_{*1}, the probability that both terms are satisfied is at most $\log t/2^{2k}$. Since each of g_{1*} and g_{*1} contains at most $2^{k-1}\alpha^2/(\sqrt{t}\log t)$ terms by property (3), by a union bound we have $\rho_2 \leq \alpha^4/(4t\log t)$, and the lemma follows:

$$\rho_1 - \rho_2 \geq \frac{\alpha^4}{8t} - \frac{\alpha^4}{4t\log t} \geq \frac{\alpha^4}{16t}$$

given the assumption that $t \geq 16$. □

Thus, our algorithm for finding all of the co-occurring pairs of a randomly chosen monotone DNF consists of estimating P_{ij} for each of the $n(n-1)/2$ pairs (i,j) so that all of our estimates are—with probability at least $(1-\delta)$—within an additive factor of $\alpha^4/32t$ of their true values. Recalling that each P_{ij} is a scaled version of the second-order Fourier coefficient, by the standard Hoeffding bound a uniform random sample of size $O(t^2\ln(n^2/\delta)/\alpha^8)$ is sufficient to estimate all of the P_{ij}'s to the specified tolerance with overall probability at least $1-\delta$. We thus have the following theorem for monotone α-interesting (k,t) with $t = O(n^{2-\gamma})$:

Theorem 3.10. *For n sufficiently large and any $\delta > 0$, with probability at least $1-\delta_{\text{usat}}-\delta_{\text{shared}}-\delta_{\text{many}}-\delta$ over the choice of f from $\mathcal{M}_n^{t,k}$ and the choice of random examples, our algorithm runs in $O(n^2t^2\log(n/\delta))$ time and identifies exactly those pairs (v_i,v_j) which co-occur in some term of f.*

3.4 Forming a hypothesis from pairs of co-occurring variables

Here we show how to construct an accurate DNF hypothesis for a random f drawn from $\mathcal{M}_n^{t,k}$.

Identifying k-cliques. By Theorem 3.10, with high probability we have complete information about which pairs of variables (v_i, v_j) co-occur in some term of f. We thus may consider the graph G with vertices $v_1, \ldots, v_n$ and edges for precisely those pairs of variables (v_i, v_j) which co-occur in some term of f. This graph is a union of t randomly chosen k-cliques from $\{v_1, \ldots, v_n\}$ which correspond to the t terms in f; we will call these the *f-cliques* of G. Ideally, if we could identify exactly the t f-cliques of G, then we could exactly reconstruct f. While we do not know how to accomplish this, we do show how to find a set of k-cliques corresponding to a set of terms whose union closely approximates f.

Specifically, we will show how to efficiently identify (with high probability over the choice of f and random examples of f) a set of k-cliques in G that contains as a subset the set of all of the f-cliques in G. Once these k-cliques have been identified, as we show later it is easy to construct an accurate DNF hypothesis for f.

The following lemma shows that with high probability over the choice of f, each pair (v_i, v_j) co-occurs in at most a constant number of terms:

Lemma 3.11. *Let $\delta_C := (\frac{tk^2}{n^2})^C$ (δ_C is a function of C as well as of t, k, and n) and fix $1 \le i < j \le n$. For any $C \ge 0$ and all sufficiently large n, we have*

$$\Pr_{f \in \mathcal{M}_n^{t,k}}[\textit{some pair of variables } (v_i, v_j) \textit{ co-occur in more than } C \textit{ terms of } f] \le \delta_C \ .$$

Proof. For any fixed $r \in \{1, \ldots, t\}$ we have that

$$\Pr[v_i \text{ and } v_j \text{ co-occur in term } T_r] = \frac{k(k-1)}{n(n-1)} \le \frac{k^2}{n^2} \ .$$

Since these events are independent for all r, the probability that there is any collection of C terms such that v_i and v_j co-occur in all C of these terms is at most

$$\binom{t}{C} \cdot \left(\frac{k^2}{n^2}\right)^C \le \left(\frac{tk^2}{n^2}\right)^C \ .$$

□

By Lemma 3.11 we know that, for any given pair (v_i, v_j) of variables, with probability at least $1 - \delta_C$ there are at most Ck other variables v_ℓ such that (v_i, v_j, v_ℓ) all co-occur in some term of f. Suppose that we can efficiently (with high probability) identify the set S_{ij} of all such variables v_ℓ. Then we can perform an exhaustive search over all $(k-2)$-element subsets S' of S_{ij} in at most $\binom{Ck}{k} \le (eC)^k = n^{O(\log C)}$ time, and can identify all of the sets S' such that $S' \cup \{v_i, v_j\}$ is a clique of size k in G that includes both v_i and v_j. Repeating this over all pairs of variables (v_i, v_j), we can with high probability identify a set G_k of k-cliques in G such that G_k contains all of the f-cliques.

Thus, to identify G_k, it remains only to show that for every pair of variables v_i and v_j, we can determine the set S_{ij} of those variables v_ℓ that co-occur in at least one term with both v_i and v_j. Assume that f is such that all pairs of variables co-occur in at most C terms, and let T be a set of variables of cardinality at most C having the following properties:

- In the projection $f_{T\leftarrow 0}$ of f in which all of the variables of T are fixed to 0, v_i and v_j do not co-occur in any term; and
- For every set $T' \subset T$ such that $|T'| = |T| - 1$, v_i and v_j do co-occur in $f_{T'\leftarrow 0}$.

Then T is clearly a subset of S_{ij}. Furthermore, if we can identify all such sets T, then their union will be S_{ij}. There are only $O(n^C)$ possible sets to consider, so our problem now reduces to the following: given a set T of at most C variables, determine whether v_i and v_j co-occur in $f_{T\leftarrow 0}$.

The proof of Lemma 3.9 shows that f is well-behaved with probability at least $1 - \delta_{\text{usat}} - \delta_{\text{shared}} - \delta_{\text{many}}$ over the choice of f. Furthermore, if f is well-behaved then it is easy to see that for every $|T| \leq C$, $f_{T\leftarrow 0}$ is also well-behaved, since $f_{T\leftarrow 0}$ is just f with $O(\sqrt{t})$ terms removed (by Lemma 3.4). That is, removing terms from f can only make it more likely that the remaining terms are uniquely satisfied, does not change the bound on the probability of a pair of remaining terms being satisfied, and can only decrease the bound on the number of remaining terms in which a remaining variable can appear. Furthermore, Lemma 3.8 holds for any monotone DNF f. Therefore, if f is well-behaved then the proof of Lemma 3.9 also shows that for every $|T| \leq C$, the P_{ij}'s of $f_{T\leftarrow 0}$ can be used to identify the co-occurring pairs of variables within $f_{T\leftarrow 0}$. It remains to show that we can efficiently simulate a uniform example oracle for $f_{T\leftarrow 0}$ so that these P_{ij}'s can be accurately estimated.

In fact, for a given set T, we can simulate a uniform example oracle for $f_{T\leftarrow 0}$ by filtering the examples from the uniform oracle for f so that only examples setting the variables in T to 0 are accepted. Since $|T| \leq C$, the filter accepts with constant probability at least $1/2^C$. A Chernoff argument shows that if all P_{ij}'s are estimated using a single sample of size $2^{C+12}t^2 \ln(2(C+2)n^C/\delta)/\alpha^8$ (filtered appropriately when needed) then all of the estimates will have the desired accuracy with probability at least $1-\delta$.

In somewhat more detail, the G_k-finding algorithm can be written as:

- Given: α, γ, C, δ
- (Note that f is well-behaved and has the "each pair occurs in at most C terms" property with probability at least $1 - \delta_{\text{usat}} - \delta_{\text{shared}} - \delta_{\text{many}} - \delta_C$. So assume this of f for the remainder of the algorithm.)
- Draw set S of $O(t^2 \log^2(n/\delta))$ examples of f
- For $1 \leq i < j \leq n$ (fewer than n^2 times)
  - Estimate P_{ij} over S ($O(|S|)$ time)
  - Add (v_i, v_j) to the set of co-occurring pairs if estimated P_{ij} exceeds the threshold $\alpha^4/(32t)$
- For each $|T| \leq C$ (at most n^C times)
  - For each co-occurring pair (v_i, v_j) disjoint from T (less than tk^2 times)
    (1) Estimate P_{ij} over $(T \leftarrow 0)$-filtered S ($O(|S|)$ time)
    (2) For each subset $T' \subset T$, $|T'| = |T| - 1$ (at most C times)
      (i) Estimate P_{ij} over $(T' \leftarrow 0)$-filtered S ($O(|S|)$ time)

        * Add T to S_{ij} if it passes all threshold tests, i.e. the estimate from (1) is at least $\alpha^4/(32t)$ and each estimate from (2)(i) is at most $\alpha^4/(32t)$ ($O(C)$ time)

- For each co-occurring pair (v_i, v_j) (less than tk^2 times)
  - For each $(k-2)$-size subset U of S_{ij} ($n^{O(\log C)}$ times)
    * Test if the union of (v_i, v_j) and U is a clique ($O(k^2)$ time)

The time bound for this algorithm is then

$$O(|S| + n^2|S| + n^C tk^2 C|S| + tk^2 n^{O(\log C)} k^2)$$

which is dominated by the third term if $C \geq 2$.

The sample complexity $|S|$ is derived as follows. We need a sample large enough to

- succeed for all n^2 tests for co-occurring pairs (over the full sample), and
- succeed for all $n^C(C+1)$ tests over filtered examples.

The total number of tests if $C \geq 2$ is bounded by $O((C+2)n^C)$. Recalling that our estimates need to be accurate to within an additive factor of $\alpha^4/32t$, we see that if all tests are run over samples of size $m = 2^{11}t^2 \ln(2(C+2)n^C/\delta)/\alpha^8$ then, by Hoeffding and the union bound, all tests succeed with probability at least $1 - \delta/2$.

We want $|S|$ large enough so that all $(C+1)n^C$ filtered samples will be of size m with probability $1 - \delta/2$. If a filter accepts with probability p over a sample of size $2m/p$, then the probability that fewer than m examples are accepted is at most $e^{-m/4}$ by Chernoff. Using the m given in the previous paragraph and the union bound, it can be seen that choosing $|S| = 2m/p$ gives us the desired probability of success over all tests.

Thus, since we are using $p = 1/2^C$ in the filtering, the final time bound of the algorithm becomes (for arbitrary $C \geq 2$) $O((2n)^C t^3 k^2 \log(Cn^C/\delta))$. This gives us the following:

Theorem 3.12. *For n sufficiently large, any $\delta > 0$, and any fixed $C \geq 2$, with probability at least $1 - \delta_{\text{usat}} - \delta_{\text{shared}} - \delta_{\text{many}} - \delta_C - \delta$ over the random draw of f from $\mathcal{M}_n^{t,k}$ and the choice of random examples, a set G_k containing all of the f-cliques of G can be identified in time $n^{O(C)} t^3 k^2 \log(n/\delta)$.*

The main learning result for monotone DNF From G_k we construct in the obvious way a list $T'_1, \ldots, T'_N$ (with $N = |G_k| = O(n^C)$) of length-k monotone terms that contains all t true terms $T_1, \ldots, T_t$ of f. Now note that the target function f is simply an OR of some subset of these N "variables" $T_1, \ldots, T_N$, so the standard elimination algorithm for PAC learning disjunctions (under any distribution) can be used to PAC learn the target function. The algorithm requires $O((1/\varepsilon)\log(1/\delta) + N/\varepsilon)$ examples and runs in time which is linear in its sample size; see e.g. Chapters 1 and 2 of [20].

Call the above described entire learning algorithm A. In summary, we have proved the following:

Theorem 3.13. *Fix $\gamma, \alpha > 0$ and $C \geq 2$. Let (k,t) be a monotone α-interesting pair. For any $\varepsilon > 0$, $\delta > 0$, and $t = O(n^{2-\gamma})$, algorithm A will with probability at least $1 - \delta_{\text{usat}} - \delta_{\text{shared}} - \delta_{\text{many}} - \delta_C - \delta$ (over the random choice of DNF from $\mathcal{M}_n^{t,k}$ and the randomness of the example oracle) produce a hypothesis h that ε-approximates the target with respect to the uniform distribution. Algorithm A runs in time polynomial in n, $\log(1/\delta)$, and $1/\varepsilon$.*

4 Non-monotone DNF

4.1 Interesting parameter settings

As with $\mathcal{M}_n^{t,k}$ we are interested in pairs (k,t) for which $\mathbf{E}_{f\in\mathcal{D}_n^{t,k}}[\Pr[f]]$ is between α and $1-\alpha$:

Definition 4.1. For $\alpha > 0$, the pair (k,t) is said to be *α-interesting* if $\alpha \le \mathbf{E}_{f\in\mathcal{D}_n^{t,k}}[\Pr[f]] \le 1-\alpha$.

For any fixed $x \in \{0,1\}^n$ we have

$$\Pr_{f\in\mathcal{D}_n^{t,k}}[f(x)=0] = \left(1-\frac{1}{2^k}\right)^t \quad , \qquad \text{and thus} \qquad \mathbf{E}_{f\in\mathcal{D}_n^{t,k}}[\Pr[f]] = 1-\left(1-\frac{1}{2^k}\right)^t$$

by linearity of expectation; this formula will be useful later.

Throughout the rest of Section 4 we assume that $\alpha > 0$ is fixed and (k,t) is an α-interesting pair where $t = O(n^{3/2-\gamma})$ for some $\gamma > 0$.

4.2 Properties of random DNF

In this section we develop analogues of Lemmas 3.6 and 3.7 for $\mathcal{D}_n^{t,k}$. The $\mathcal{D}_n^{t,k}$ analogue of Lemma 3.7 follows directly from the proof of Lemma 3.7, and we have:

Lemma 4.2. *With probability at least $1-\delta_{\text{shared}}$ over the random draw of f from $\mathcal{D}_n^{t,k}$, for all $1 \le i < j \le n$, $\Pr[T_i \wedge T_j] \le \frac{\log t}{2^{2k}}$.*

In the following lemma we use McDiarmid's bound to prove a $\mathcal{D}_n^{t,k}$ version of Lemma 3.6:

Lemma 4.3. *Let*

$$\delta'_{\text{usat}} := t\left((t-1)\left(\frac{k^2}{n}\right)^{\log\log t} + \exp\left(\frac{-\alpha^2 t}{16\ln^2(2/\alpha)\log^2 t}\right)\right) \quad .$$

With probability at least $1-\delta'_{\text{usat}}$, a random f drawn from $\mathcal{D}_n^{t,k}$ is such that for each $i = 1,\ldots,t$, we have

$$P_i \equiv \Pr_x[T_i \text{ is satisfied by } x \text{ but no other } T_j \text{ is satisfied by } x] \ge \frac{\alpha}{2^{k+1}} \quad .$$

Proof. We show that $P_1 \ge \alpha/2^{k+1}$ with probability at least $1-\delta'_{\text{usat}}/t$; the lemma follows by a union bound. We first show that $\mathbf{E}_{f\in\mathcal{D}_n^{t,k}}[P_1] \ge \alpha/2^k$. For any fixed $x \in T_1$, we have

$$\Pr[\overline{T}_2(x) \wedge \cdots\cdots \wedge \overline{T}_t(x)] = (1-2^{-k})^{t-1} > (1-2^{-k})^t \ge \alpha$$

where the last inequality holds since (k,t) is α-interesting. Since a 2^{-k} fraction of all $x \in \{0,1\}^n$ belong to T_1, by linearity of expectation we have $\mathbf{E}_{f\in\mathcal{D}_n^{t,k}}[P_1] \ge \alpha/2^k$.

Now we show that with high probability the deviation of P_1 from its expected value is low. Given any fixed length-k term T_1, let Ω denote the set of all length-k terms T which satisfy $\Pr[T_1 \wedge T] \le (\log t)/2^{2k}$. By reasoning as in the proof of Lemma 4.2, with probability at least $1-(t-1)(\frac{k^2}{n})^{\log\log t}$

each of $T_2,\ldots,T_t$ belongs to Ω, so we henceforth assume that this is in fact the case, i.e. we condition on the event $\{T_2,\ldots,T_t\}\subset\Omega$. Note that under this conditioning we have that each of $T_2,\ldots,T_t$ is selected uniformly and independently from Ω. Note also that this conditioning can change the value of P_1 (a probability) by at most $(t-1)(\frac{k^2}{n})^{\log\log t} < \frac{\alpha}{2^{k+2}}$, so under this conditioning we have $\mathbf{E}[P_1] \geq \frac{3}{4}\cdot\frac{\alpha}{2^k}$.

We now use McDiarmid's inequality where the random variables are the randomly selected terms $T_2,\ldots,T_t$ from Ω and $F(T_2,\ldots,T_t)$ denotes P_1, i.e.

$$F(T_2,\ldots,T_t) = \Pr_x[T_1 \text{ is satisfied by } x \text{ but no } T_j \text{ with } j\geq 2 \text{ is satisfied by } x] \ .$$

Since each T_j belongs to Ω, we have

$$|F(T_2,\ldots,T_t) - F(T_2,\ldots,T_{j-1},T_j',T_{j+1},\ldots,T_t)| \leq c_i = \frac{\log t}{2^{2k}}$$

for all $j=2,\ldots,t$. Taking $\tau = \frac{1}{4}\cdot\frac{\alpha}{2^k}$, McDiarmid's inequality implies that $\Pr\left[P_1 < \frac{\alpha}{2^{k+1}}\right]$ is at most

$$\exp\left(\frac{-\alpha^2/(16\cdot 2^{2k})}{(t-1)(\frac{\log t}{2^{2k}})^2}\right) = \exp\left(\frac{-\alpha^2 2^{2k}}{16(t-1)\log^2 t}\right) \leq \exp\left(\frac{-\alpha^2 2^{2k}}{16t\log^2 t}\right) \leq \exp\left(\frac{-\alpha^2 t}{16\ln^2(2/\alpha)\log^2 t}\right)$$

where the last inequality holds since (k,t) is α-interesting. Combining all the failure probabilities, the lemma is proved. □

4.3 Identifying (most pairs of) co-occurring variables

Recall that in Section 3.3 we partitioned the terms of our monotone DNF into four disjoint groups depending on what subset of $\{v_1,v_2\}$ was present in each term. In the non-monotone case, we will partition the terms of f into nine disjoint groups depending on whether each of v_1,v_2 is unnegated, negated, or absent:

$$f = g_{**} \vee (v_1 g_{1*}) \vee (\overline{v_1} g_{0*}) \vee (v_2 g_{*1}) \vee (v_1 v_2 g_{11}) \vee (\overline{v_1} v_2 g_{01}) \vee (\overline{v_2} g_{*0}) \vee (v_1 \overline{v_2} g_{10}) \vee (\overline{v_1}\,\overline{v_2} g_{00})$$

Thus g_{**} contains those terms of f which contain neither v_1 nor v_2 in any form; g_{0*} contains the terms of f which contain $\overline{v_1}$ but not v_2 in any form (with $\overline{v_1}$ removed from each term); g_{*1} contains the terms of f which contain v_2 but not v_1 in any form (with v_2 removed from each term); and so on. Each $g_{\cdot,\cdot}$ is thus a DNF (possibly empty) over literals formed from $v_3,\ldots,v_n$.

For all four possible values of $(a,b)\in(0,1)^2$, we can empirically estimate

$$p_{ab} \ := \ \Pr_x[g_{**}\vee g_{a*}\vee g_{*b}\vee g_{ab}] = \Pr_x[f(x)=1 \mid x_1=a, x_2=b] \ .$$

It is easy to see that $\Pr[g_{11}]$ is either 0 or else at least $4/2^k$ depending on whether g_{11} is empty or not. Ideally we would like to be able to accurately estimate each of $\Pr[g_{00}]$, $\Pr[g_{01}]$, $\Pr[g_{10}]$, and $\Pr[g_{11}]$; if we could do this then we would have complete information about which pairs of literals involving variables v_1 and v_2 co-occur in terms of f. Unfortunately, the probabilities $\Pr[g_{00}]$, $\Pr[g_{01}]$, $\Pr[g_{10}]$, and $\Pr[g_{11}]$ cannot in general be obtained from p_{00}, p_{01}, p_{10}, and p_{11}. However, we will show that we can efficiently obtain some partial information which enables us to learn to fairly high accuracy.

As before, our approach is to accurately estimate the quantity $P = p_{11}-p_{10}-p_{01}+p_{00}$. We have the following two lemmas:

Lemma 4.4. *If all four of g_{00}, g_{01}, g_{10}, and g_{11} are empty, then P equals*

$$\begin{aligned}&\Pr[g_{1*} \wedge g_{*0} \wedge (\textit{no other } g_{\cdot,\cdot})] + \Pr[g_{0*} \wedge g_{*1} \wedge (\textit{no other } g_{\cdot,\cdot})] \\ &\quad - \Pr[g_{1*} \wedge g_{*1} \wedge (\textit{no other } g_{\cdot,\cdot})] - \Pr[g_{0*} \wedge g_{*0} \wedge (\textit{no other } g_{\cdot,\cdot})] \ . \end{aligned} \tag{4.1}$$

Proof. Since all four of g_{00}, g_{01}, g_{10}, and g_{11} are empty we need only consider the five events g_{**}, g_{*0}, g_{0*}, g_{*1}, and g_{1*}. We now analyze the contribution to P from each possible subset of these 5 events:

- P gets a net contribution of 0 from those x which belong to $g_{*,*}$ (and to any other subset of the remaining four events) since each such x is counted in each of p_{00}, p_{01}, p_{10}, and p_{11}. It remains to consider all 16 subsets of the four events g_{*0}, g_{0*}, g_{*1}, and g_{1*}.

- P gets a net contribution of 0 from those x which are in at least 3 of the four events g_{*0}, g_{0*}, g_{*1}, and g_{1*} since each such x is counted in each of p_{00}, p_{01}, p_{10}, and p_{11}. P also gets a net contribution of 0 from those x which are in exactly one of the four events g_{*0}, g_{0*}, g_{*1}, and g_{1*}. It remains to consider those x which are in exactly two of the four events g_{1*}, g_{0*}, g_{*1}, and g_{*0}.

- P gets a net contribution of 0 from those x which are in g_{1*} and g_{0*} and no other events, since each such x is counted in each of p_{00}, p_{01}, p_{10}, and p_{11}. The same is true for those x which are in g_{*1} and g_{*0} and no other events.

- P gets a net contribution of

  $$-\Pr[g_{1*} \wedge g_{*1} \wedge (\text{no other } g_{\cdot,\cdot} \text{ occurs})]$$

  from those x which are in g_{1*} and g_{*1} and no other event. Similarly, P gets a net contribution of

  $$-\Pr[g_{0*} \wedge g_{*0} \wedge (\text{no other } g_{\cdot,\cdot} \text{ occurs})]$$

  from those x which are in g_{0*} and g_{*0} and no other event. P gets a net contribution of

  $$\Pr[g_{1*} \wedge g_{*0} \wedge (\text{no other } g_{\cdot,\cdot} \text{ occurs})]$$

  from those x which are in g_{1*} and g_{*0} and no other event, and gets a net contribution of

  $$\Pr[g_{0*} \wedge g_{*1} \wedge (\text{no other } g_{\cdot,\cdot} \text{ occurs})]$$

  from those x which are in g_{0*} and g_{*1} and no other event.

□

Lemma 4.5. *If exactly one of g_{00}, g_{01}, g_{10} and g_{11} is nonempty (say g_{11}), then P equals (4.1) plus*

$$\begin{aligned}&\Pr[g_{11} \wedge g_{1*} \wedge g_{*0} \wedge (\textit{no other } g_{\cdot,\cdot})] + \Pr[g_{11} \wedge g_{0*} \wedge g_{*1} \wedge (\textit{no other } g_{\cdot,\cdot})] \\ &\quad - \Pr[g_{11} \wedge g_{1*} \wedge g_{*1} \wedge (\textit{no other } g_{\cdot,\cdot})] - \Pr[g_{11} \wedge g_{0*} \wedge g_{*0} \wedge (\textit{no other } g_{\cdot,\cdot})] \\ &\quad + \Pr[g_{11} \wedge g_{0*} \wedge (\textit{no other } g_{\cdot,\cdot})] + \Pr[g_{11} \wedge g_{*0} \wedge (\textit{no other } g_{\cdot,\cdot})] + \Pr[g_{11} \wedge (\textit{no other } g_{\cdot,\cdot})] \ . \end{aligned}$$

Proof. We suppose that g_{11} is nonempty. We wish to analyze the contribution to P from all 64 subsets of the six events g_{**}, g_{1*}, g_{0*}, g_{*1}, g_{*0}, and g_{11}. From Lemma 4.4 we know this contribution for the 32 subsets which do not include g_{11} is (4.1) so only a few cases remain:

- P gets a net contribution of 0 from those x which are in g_{11} and in g_{**} and in any other subset of events (each such x is counted in each of p_{11}, p_{01}, p_{10}, and p_{00}). Similarly, P gets a contribution of 0 from those x which are in g_{11} and in at least three of g_{1*}, g_{0*}, g_{*1}, and g_{*0}. So it remains only to analyze the contribution from subsets which contain g_{11}, contain at most two of g_{1*}, g_{0*}, g_{*1}, g_{*0}, and contain nothing else.

- An analysis similar to that of Lemma 4.4 shows that P gets a net contribution of

$$\begin{aligned}&\Pr[g_{11} \wedge g_{1*} \wedge g_{*0} \wedge (\text{no other } g_{\cdot,\cdot})] + \Pr[g_{11} \wedge g_{0*} \wedge g_{*1} \wedge (\text{no other } g_{\cdot,\cdot})] \\ &\quad - \Pr[g_{11} \wedge g_{1*} \wedge g_{*1} \wedge (\text{no other } g_{\cdot,\cdot})] - \Pr[g_{11} \wedge g_{0*} \wedge g_{*0} \wedge (\text{no other } g_{\cdot,\cdot})]\end{aligned}$$

  from those x which are in g_{11}, in exactly two of $\{g_{1*}, g_{0*}, g_{*1}, g_{*0}\}$, and in no other events. So it remains only to consider subsets which contain g_{11} and at most one of g_{1*}, g_{0*}, g_{*1}, g_{*0}, and nothing else.

- P gets a contribution of 0 from x which are in g_{11} and g_{1*} and in nothing else; likewise from x which are in g_{11} and g_{*1} and in nothing else. P gets a contribution of

$$\Pr[g_{11} \wedge g_{0*} \wedge (\text{no other } g_{\cdot,\cdot})]$$

  from x which are in g_{11} and g_{0*} and in nothing else, and a contribution of

$$\Pr[g_{11} \wedge g_{*0} \wedge (\text{no other } g_{\cdot,\cdot})]$$

  from x which are in g_{11} and g_{*0} and in nothing else.

- P gets a net contribution of $\Pr[g_{11} \wedge (\text{ no other } g_{\cdot,\cdot})]$ from those x which are in g_{11} and in no other event.

□

Using the above two lemmas we can show that the value of P is a good indicator for distinguishing between all four of g_{00}, g_{01}, g_{10}, and g_{11} being empty versus exactly one of them being nonempty:

Lemma 4.6. *For n sufficiently large and $t \geq 4$, with probability at least $1 - \delta'_{\text{usat}} - \delta_{\text{shared}} - \delta_{\text{many}}$ over a random draw of f from $\mathcal{D}_n^{t,k}$, we have that: (i) if v_1 and v_2 do not co-occur in any term of f then $P \leq \alpha^2/8t$; (ii) if v_1 and v_2 do co-occur in some term of f and exactly one of g_{00}, g_{01}, g_{10}, and g_{11} is nonempty, then $P \geq 3\alpha^2/16t$.*

Proof. With probability at least $1 - \delta'_{\text{usat}} - \delta_{\text{shared}} - \delta_{\text{many}}$ a randomly chosen f from $\mathcal{D}_n^{t,k}$ will have all of the following properties:

1. Each term in f is uniquely satisfied with probability at least $\alpha/2^{k+1}$ (by Lemma 4.3);

2. Each variable in f appears in at most $2^{k-1}\alpha^2/(\sqrt{t}\log t)$ terms (by Lemma 3.4); and

3. Each pair of terms T_i and T_j in f are both satisfied with probability at most $\log t/2^{2k}$ (by Lemma 4.2).

For the sequel assume that we have such an f. We first prove (i) by showing that P—as represented by (4.1) of Lemma 4.4—is at most $\alpha^4/(t\log t)$. By property 3 above, for any pair of terms consisting of one term from g_{1*} and the other from g_{*0}, the probability that both terms are satisfied is at most $\log t/2^{2k}$. Since each of g_{1*} and g_{*0} contains at most $2^{k-1}\alpha^2/(\sqrt{t}\log t)$ terms by property 2, a union bound gives

$$\Pr[g_{1*} \wedge g_{*0} \wedge (\text{no other } g_{\cdot,\cdot})] \leq \Pr[g_{1*} \wedge g_{*0}] \leq \frac{\alpha^4}{4t\log t} \quad .$$

A similar bound holds for $\Pr[g_{0*} \wedge g_{*1} \wedge (\text{no other } g_{\cdot,\cdot})]$, which is the only other positive summand in (4.1), so P is certainly at most $\alpha^4/(t\log t)$. This is at most $\alpha^2/8t$ since $\alpha \leq 1/2$ and $t \geq 4$.

We now prove (ii). By an argument similar to the above we have that the first six summands (not including (4.1) in the expression of Lemma 4.5, namely $\Pr[g_{11} \wedge g_{1*} \wedge g_{*0} \wedge (\text{no other } g_{\cdot,\cdot})]$ through $\Pr[g_{11} \wedge g_{*0} \wedge (\text{no other } g_{\cdot,\cdot})]$, are each at most $\alpha^4/(4t\log t)$ in magnitude. Now observe that each instance x that uniquely satisfies a term T_j in f containing both v_1 unnegated and v_2 unnegated must satisfy g_{11} and no other $g_{\cdot,\cdot}$. Thus under the conditions of (ii) the last summand in Lemma 4.5, namely $\Pr[g_{11} \wedge (\text{ no other } g_{\cdot,\cdot})]$, is at least $\alpha/2^{k+1}$ by property 1 above, so we have that (ii) is at least

$$\frac{\alpha}{2^{k+1}} - \frac{5}{2}\frac{\alpha^4}{t\log t} \quad .$$

(Here the $5\alpha^4/(2t\log t)$ comes from the ten summands – four from (4.1) and six from the first six summands of Lemma 4.5 – each of which contributes at most $\alpha^4/(4t\log t)$ in magnitude.) Since (k,t) is α-interesting we have $t/2^k \geq \alpha$, and from this and the constant bounds on α and t it is easily shown that

$$\frac{\alpha}{2^{k+1}} \geq \frac{\alpha^2}{2t} \qquad \text{and} \qquad \frac{5}{2}\frac{\alpha^4}{t\log t} \leq \frac{5\alpha^2}{16t} \quad ,$$

from which the lemma follows. □

It is clear that an analogue of Lemma 4.6 holds for any pair of variables v_i, v_j in place of v_1, v_2. Thus, for each pair of variables v_i, v_j, if we decide whether v_i and v_j co-occur (negated or otherwise) in any term on the basis of whether P_{ij} is large or small, we will err only if two or more of g_{00}, g_{01}, g_{10}, and g_{11} are nonempty.

We now show that for $f \in \mathcal{D}_n^{t,k}$, with very high probability there are not too many pairs of variables (v_i, v_j) which co-occur (with any sign pattern) in at least two terms of f. Note that this immediately bounds the number of pairs (v_i, v_j) which have two or more of the corresponding g_{00}, g_{01}, g_{10}, and g_{11} nonempty.

Lemma 4.7. *Let $d > 0$ and $f \in \mathcal{D}_n^{t,k}$. The probability that more than $(d+1)t^2k^4/n^2$ pairs of variables (v_i, v_j) each co-occur in two or more terms of f is at most* $\exp(-d^2t^3k^4/n^4)$.

Proof. We use McDiarmid's inequality, where the random variables are the terms $T_1,\ldots,T_t$ chosen independently from the set of all possible terms of length k and $F(T_1,\ldots,T_t)$ denotes the number of pairs of variables (v_i,v_j) that co-occur in at least two terms. For each $\ell = 1,\ldots,t$ we have

$$\Pr[T_\ell \text{ contains both } v_1 \text{ and } v_2] \leq \frac{k^2}{n^2} \ ,$$

so by a union bound we have

$$\Pr[f \text{ contains at least two terms which contain both } v_1 \text{ and } v_2 \text{ in any form}] \leq \frac{t^2k^4}{n^4} \ .$$

By linearity of expectation we have $\mu = \mathbf{E}[F] \leq t^2k^4/n^2$. Since each term involves at most k^2 pairs of co-occurring variables, we have

$$|F(T_1,\ldots,T_t) - F(T_1,\ldots,T_{i-1},T_i',T_{i+1},\ldots,T_t)| \leq c_i = k^2 \ .$$

We thus have by McDiarmid's inequality that $\Pr[F \geq t^2k^4/n^2 + \tau] \leq \exp(-\tau^2/(tk^4))$. Taking $\tau = dt^2k^4/n^2$, we have $\Pr[F \geq (d+1)t^2k^4/n^x 2] \leq \exp(-d^2t^3k^4/n^4)$. □

Taking $d = n^2/(t^{5/4}k^4)$ in the above lemma (note that $d > 1$ for n sufficiently large since $t^{5/4} = O(n^{15/8})$), we have $(d+1)t^2k^4/n^2 \leq 2t^{3/4}$ and the failure probability is at most $\exp(-\sqrt{t}/k^4)$ (we henceforth write $\delta_{\text{co-occur}}$ to denote this quantity $\exp(-\sqrt{t}/k^4)$). The results of this section (together with a standard analysis of error in estimating each P_{ij}) thus yield:

Theorem 4.8. *For n sufficiently large and for any $\delta > 0$, with probability at least $1 - \delta_{\text{co-occur}} - \delta'_{\text{usat}} - \delta_{\text{shared}} - \delta_{\text{many}} - \delta$ over the random draw of f from $\mathcal{D}_n^{t,k}$ and the choice of random examples, the above algorithm runs in $O(n^2t^2\log(n/\delta))$ time and outputs a list of pairs of variables (v_i,v_j) such that: (i) if (v_i,v_j) is in the list then v_i and v_j co-occur in some term of f; and (ii) at most $N_0 = 2t^{3/4}$ pairs of variables (v_i,v_j) which do co-occur in f are not on the list.*

4.4 Reconstructing an accurate DNF hypothesis

It remains to construct a good hypothesis for the target DNF from a list of pairwise co-occurrence relationships as provided by Theorem 4.8. As in the monotone case, we consider the graph G with vertices $v_1,\ldots,v_n$ and edges for precisely those pairs of variables (v_i,v_j) which co-occur (with any sign pattern) in some term of f. As before this graph is a union of t randomly chosen k-cliques $S_1,\ldots,S_t$ which correspond to the t terms in f, and as before we would like to find a set of k-cliques in G that contains the t k-cliques corresponding to the terms of f as a subset. However, there are two differences now: the first is that instead of having the true graph G, we instead have access only to a graph G' which is formed from G by deleting some set of at most $N_0 = 2t^{3/4}$ edges. The second difference is that the final hypothesis must take the signs of literals in each term into account. To handle these two differences, we use a different reconstruction procedure than we used for monotone DNF in Section 3.4; this reconstruction procedure only works for $t = O(n^{3/2-\gamma})$ where $\gamma > 0$.

We first show how to identify (with high probability over the choice of f) the set of *all* k-cliques in G'; clearly, the k-cliques corresponding to terms in f are a subset of this set. We then show how to form a DNF hypothesis from the set of all k-cliques in G'.

We now describe an algorithm which, for $t = O(n^{3/2-\gamma})$ with $\gamma > 0$, with high probability runs in polynomial time and identifies all the k-cliques in G' which contain edge (v_1, v_2). Since G' has at most tk^2 edges, running the algorithm at most tk^2 times on all edges in G' will give us with high probability all the k-cliques in G'. The algorithm is:

- Let Δ be the set of vertices v_j such that v_1, v_2, and v_j form a triangle in G'. Run a brute-force algorithm to find all $(k-2)$-cliques in the subgraph induced by Δ (this is the subgraph of G' whose vertices are the vertices of Δ, and whose edges are the edges that are present in G' between vertices of Δ).

It is clear that the algorithm finds every k-clique which contains edge (v_1, v_2). To bound the algorithm's running time, it suffices to give a high probability bound on the size of Δ in the graph G (clearly Δ only shrinks in passing from G to G'). The following lemma gives such a bound:

Lemma 4.9. *Let G be a random graph as described above. For any $t = O(n^{3/2-\gamma})$ and any $C > 0$ we have that with probability $1 - O\left(\frac{\log^{6C} n}{n^{2\gamma C}}\right)$ the size of Δ in G is at most Ck.*

Proof. In order for v_1, v_2, and v_j to form a triangle in G, it must be the case that either (i) some clique S_i contains $\{1, 2, j\}$; or (ii) there is some pair of cliques S_a, S_b with $2 \notin S_a$ and $\{1, j\} \subset S_a$ and $1 \notin S_b$ and $\{2, j\} \subset S_b$.

For (i), we have from Lemma 3.11 that v_1 and v_2 co-occur in more than C terms with probability at most $(tk^2/n^2)^C$. Since each term in which v_1 and v_2 co-occur contributes at most $k-2$ vertices v_j to condition (i), the probability that more than $C(k-2)$ vertices v_j satisfy condition (i) is at most $(tk^2/n^2)^C = O(1/n^{C/2})$.

For (ii), let A be the set of those indices $a \in \{1, \ldots, t\}$ such that $2 \notin S_a$ and $1 \in S_a$, and let S_A be $\cup_{a \in A} S_a$. Similarly let B be the set of indices b such that $1 \notin S_b$ and $2 \in S_b$, and let S_B be $\cup_{b \in B} S_b$. It is clear that A and B are disjoint. For each $\ell = 1, \ldots, t$ we have that $\ell \in A$ independently with probability at most $p = k/n$, so $E[|A|] \leq tk/n$. We now consider two cases:

Case 1: $t \leq n/\log n$. In this case we may take $\beta = n \log n/(tk)$ in the Chernoff bound, and we have that $\Pr[|A| \geq \beta pt]$ equals

$$\Pr[|A| \geq \log n] \leq \left(\frac{e}{\beta}\right)^{\beta pt} \leq \left(\frac{ek}{\log^2 n}\right)^{\log n} = \left(\frac{e}{\Omega(\log n)}\right)^{\log n} = \frac{1}{n^{\omega(1)}} \ .$$

The same bound clearly holds for B. Note that in Case 1 we thus have $|S_A|, |S_B| \leq k \log n$ with probability $1 - 1/n^{\omega(1)}$.

Case 2: $t > n/\log n$. In this case we may take $\beta = \log n$ in the Chernoff bound and we obtain

$$\Pr[|A| \geq \beta pt] = \Pr\left[|A| \geq \frac{tk \log n}{n}\right] \leq \left(\frac{e}{\log n}\right)^{kt(\log n)/n} < \left(\frac{e}{\log n}\right)^{k} = \frac{1}{n^{\omega(1)}}$$

where the last inequality holds since $k = \Omega(\log n)$ (since $t > n/\log n$ and (k,t) is α-interesting). In Case 2 we thus have $|S_A|, |S_B| \leq (tk^2 \log n)/n$ with probability $1 - 1/n^{\omega(1)}$.

Let S'_A denote $S_A - \{1\}$ and S'_B denote $S_B - \{2\}$. Since A and B are disjoint, it is easily seen that conditioned on S'_A being of some particular size s'_A, all $\binom{n-2}{s'_A}$ s'_A-element subsets of $\{3,\ldots,n\}$ are equally likely for S'_A. Likewise, conditioned on S'_B being of size s'_B, all $\binom{n-2}{s'_B}$ s'_B-element subsets of $\{3,\ldots,n\}$ are equally likely for S'_B. Thus, the probability that $|S'_A \cap S'_B| \geq C$ is at most

$$\binom{s'_B}{C}\left(\frac{s'_A}{n-2}\right)^C \leq \left(\frac{s'_A s'_B}{n-2}\right)^C \leq \left(\frac{2s'_A s'_B}{n}\right)^C \tag{4.2}$$

(since the expression on the left is an upper bound on the probability that any collection of C elements in S'_B all coincide with elements of S'_A).

In Case 1 ($t \leq n/\log n$) we may assume that s'_A, s'_B are each at most $k \log n$ (recall from above that this holds with probability $1 - n^{-\omega(1)}$), and thus (4.2) is at most $[(2k^2 \log^2 n)/n]^C$. In Case 2 ($t > n/\log n$) we may assume that $s'_A, s'_B \leq (tk^2 \log n)/n$ (here too from above we have that this holds with probability $1 - n^{-\omega(1)}$) and thus (4.2) is at most

$$\left(\frac{2t^2k^4 \log^2 n}{n^3}\right)^C = O\left(\frac{\log^{6C} n}{n^{2\gamma C}}\right) .$$

Thus all in all, we have that except with probability $O(1/n^{C/2})$ event (i) contributes at most $C(k-2)$ vertices v_j such that $\{1,2,j\}$ forms a triangle, and except with probability $O\left(\frac{\log^{6C} n}{n^{2\gamma C}}\right)$ event (ii) contributes at most C vertices v_j such that $\{1,2,j\}$ forms a triangle. This proves the lemma. □

By Lemma 4.9, doing a brute-force search which finds all $(k-2)$-cliques in the graph induced by Δ takes at most $\binom{Ck}{k} \leq \left(\frac{eCk}{k}\right)^k = (eC)^{O(\log n)} = n^{O(\log C)}$ time steps. Thus we can efficiently with high probability identify all the k-cliques in G'. How many of the "true" cliques $S_1,\ldots,S_t$ in G are not present as k-cliques in G'? By Lemma 3.11, with probability at least $1 - t^2(tk^2/n^2)^C$ each edge (v_i, v_j) participates in at most C cliques from $S_1,\ldots,S_t$. Since G' is missing at most N_0 edges from G', with probability at least $1 - t^2(tk^2/n^2)^C$ the set of all k-cliques in G' is missing at most CN_0 "true" cliques from $S_1,\ldots,S_t$.

Summarizing the results of this section so far, we have:

Theorem 4.10. *Fix $C \geq 2$. Given a DNF formula f drawn from $\mathcal{D}_n^{t,k}$ and a list of pairs of co-occurring variables as described in Theorem 4.8, with probability at least $1 - 1/n^{\Omega(C)}$ the above procedure runs in $n^{O(\log C)}$ time and constructs a a list $Z_1,\ldots,Z_{N'}$ (where $N' = n^{O(\log C)}$) of k-cliques which contains all but at most CN_0 of the cliques $S_1,\ldots,S_t$.*

We construct a hypothesis DNF from the list $Z_1,\ldots,Z_{N'}$ of candidate k-cliques as follows: for each Z_i we form all 2^k possible terms which could have given rise to Z_i (corresponding to all 2^k sign patterns on the k variables in Z_i). We then test each of these $2^k N'$ potential terms against a sample of M randomly drawn negative examples and discard any terms which output 1 on any negative example; the final hypothesis h is the OR of all surviving terms. Any candidate term T' which has

$$\Pr_{x \in U_n}[T'(x) = 1 \wedge f(x) = 0] \geq \frac{\varepsilon}{2^{k+1}N'}$$

will survive this test with probability at most $\exp(-\varepsilon M/(2^{k+1}N'))$. Taking $\varepsilon = 1/2^k$ and

$$M = \frac{2^{k+1}N' \log^2 n}{\varepsilon}$$

we have that with probability $1 - 1/n^{\omega(1)}$ each term in the final hypothesis contributes at most $\varepsilon/(2^{k+1}N')$ toward the false positive rate of h, so with high probability the false positive rate of h is at most $\varepsilon = 1/2^k$.

The false negative rate of h is at most $\frac{1}{2^k}$ times the number of terms in f which are missing in h. Since the above algorithm clearly will not discard any term in f (since such a term will never cause a false negative mistake), we need only bound the number of terms in f which are not among our $2^k N'$ candidates. With probability at least $1 - \delta_{\text{clique}} := 1 - t^2/\binom{n}{k}$, each true clique $S_1, \ldots, S_t$ in G gives rise to exactly one term of f (the only way this does not happen is if two terms consist of literals over the exact same set of k variables, and the probability that this occurs is at most $t^2/\binom{n}{k}$), so Theorem 4.10 implies that h is missing at most CN_0 terms of f. Thus the false negative rate is at most

$$\frac{CN_0}{2^k} \leq \frac{2Ct^{3/4}}{2^k} = \frac{1}{\Omega(t^{1/4})} \quad .$$

All in all the following is our main learning result for non-monotone DNF:

Theorem 4.11. *Fix γ, $\alpha > 0$ and $C \geq 2$. Let (k,t) be a monotone α-interesting pair. For f randomly chosen from $\mathcal{D}_n^{t,k}$, with probability at least $1 - \delta_{\text{co-occur}} - \delta'_{\text{usat}} - \delta_{\text{shared}} - \delta_{\text{many}} - \delta_{\text{clique}} - 1/n^{\Omega(C)}$ the above algorithm runs in $\tilde{O}(n^2t^2 + n^{O(\log C)})$ time and outputs a hypothesis h whose error rate relative to f under the uniform distribution is at most $1/\Omega(t^{1/4})$.*

It can be verified from the definitions of the various δ's that for any $t = \omega(1)$ as a function of n, the failure probability is $o(1)$ and the accuracy is $1 - o(1)$.

5 Future work

We can currently only learn random DNFs with $o(n^{3/2})$ terms ($o(n^2)$ terms for monotone DNF); can stronger results be obtained which hold for all polynomial-size DNF? A natural approach here for learning n^c-term DNF might be to first try to identify all c'-tuples of variables which co-occur in a term, where c' is some constant larger than c. Also, our current results for $t = \omega(1)$-term DNF let us learn to some $1 - o(1)$ accuracy but we cannot yet achieve an arbitrary inverse polynomial error rate for non-monotone DNF. Finally, another interesting direction is to explore other natural models of random DNF formulas, perhaps by allowing some variation among term sizes or dependencies between terms.

Acknowledgement. Avrim Blum suggested to one of us (JCJ) the basic strategy that learning monotone DNF with respect to uniform might be reducible to finding the co-occurring pairs of variables in the target function. We thank the anonymous referees for helpful suggestions and corrections. This material is based upon work supported by the National Science Foundation under Grant No. CCR-0209064 (JCJ) and CCF-0347282 (RAS).

References

[1] * H. AIZENSTEIN AND L. PITT: On the learnability of disjunctive normal form formulas. *Machine Learning*, 19:183–208, 1995. [ML:n226835168336578]. 1.1

[2] * NOGA ALON AND JOEL H. SPENCER: *The Probabilistic Method*. John Wiley and Sons, 2000. 2.1

[3] * D. ANGLUIN: Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988. [ML:u228266621966h58]. 1.1

[4] * DANA ANGLUIN AND MICHAEL KHARITONOV: When won't membership queries help? *Journal of Computer and System Sciences*, 50(2):336–355, 1995. [JCSS:10.1006/jcss.1995.1026]. 1.1

[5] * A. BLUM: Learning a function of r relevant variables (open problem). In *Proc. 16th Ann. Conf. on Computational Learning Theory (COLT'03)*, volume 2777 of *Lecture Notes in Computer Science*, pp. 731–733. Springer, 2003. [COLT:fxdg79cvndr6n05r]. 1.2

[6] * A. BLUM: Machine learning: a tour through some favorite results, directions, and open problems. FOCS 2003 tutorial slides, available at http://www-2.cs.cmu.edu/ avrim/Talks/FOCS03/tutorial.ppt, 2003. 1.1

[7] * A. BLUM, C. BURCH, AND J. LANGFORD: On learning monotone boolean functions. In *Proc. 39th FOCS*, pp. 408–415. IEEE Computer Society Press, 1998. [FOCS:10.1109/SFCS.1998.743491]. 1.1

[8] * A. BLUM, M. FURST, J. JACKSON, M. KEARNS, Y. MANSOUR, AND S. RUDICH: Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proc. 26th STOC*, pp. 253–262. ACM Press, 1994. [STOC:195058.195147]. 1.1

[9] * B. BOLLOBÁS: *Combinatorics: Set Systems, Hypergraphs, Families of Vectors and Combinatorial Probability*. Cambridge University Press, 1986. 3.2

[10] * M. GOLEA, M. MARCHAND, AND T. HANCOCK: On learning μ-perceptron networks on the uniform distribution. *Neural Networks*, 9(1):67–82, 1994. [NeuralNet:10.1016/0893-6080(95)00009-7]. 2.2

[11] * T. HANCOCK: Learning $k\mu$ decision trees on the uniform distribution. In *Proc. 6th Ann. Conf. on Computational Learning Theory (COLT'93)*, pp. 352–360. ACM Press, 1993. [ACM:168304.168374]. 2.2

[12] * J. JACKSON: An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55(3):414–440, 1997. [JCSS:10.1006/jcss.1997.1533]. 1.1

[13] * J. JACKSON, A. KLIVANS, AND R. SERVEDIO: Learnability beyond AC^0. In *Proc. 34th STOC*, pp. 776–784. ACM Press, 2002. [STOC:509907.510018]. 2.2

[14] * J. JACKSON AND R. SERVEDIO: Learning random log-depth decision trees under the uniform distribution. In *Proc. 16th Ann. Conf. on Computational Learning Theory (COLT'03) and 7th Kernel Workshop*, volume 2777 of *Lecture Notes in Computer Science*, pp. 610–624. Springer, 2003. [doi:10.1007/b12006, COLT:31wxjv7lb9l5]. 2.2

[15] * J. JACKSON AND R. SERVEDIO: On learning random DNF formulas under the uniform distribution. In *Proc. 9th Internat. Workshop on Randomization and Computation (RANDOM'05)*, volume 3624 of *Lecture Notes in Computer Science*, pp. 342–353. Springer, 2005. [RANDOM:2y5933y326xhbgar]. 1.2

[16] * J. JACKSON AND C. TAMON: Fourier Analysis in Machine Learning. ICML/COLT 1997 tutorial slides, available at http://learningtheory.org/resources.html, 1997. 1.1

[17] * M. KEARNS: Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998. [JACM:293347.293351]. 1.1

[18] * M. KEARNS, M. LI, L. PITT, AND L. VALIANT: On the learnability of Boolean formulae. In *Proc. 19th STOC*, pp. 285–295. ACM Press, 1987. [STOC:28395.28426]. 3.1

[19] * M. KEARNS, M. LI, L. PITT, AND L. VALIANT: Recent results on Boolean concept learning. In *Proc. 4th Internat. Workshop on Machine Learning*, pp. 337–352. Morgan Kaufmann, 1987. 1.1

[20] * M. KEARNS AND U. VAZIRANI: *An introduction to computational learning theory*. MIT Press, Cambridge, MA, 1994. 3.4

[21] * A. KLIVANS, R. O'DONNELL, AND R. SERVEDIO: Learning intersections and thresholds of halfspaces. In *Proc. 43rd FOCS*, pp. 177–186. IEEE Computer Society Press, 2002. [FOCS:10.1109/SFCS.2002.1181894]. 2.2

[22] * L. KUČERA, A. MARCHETTI-SPACCAMELA, AND M. PROTASSI: On learning monotone DNF formulae under uniform distributions. *Information and Computation*, 110:84–95, 1994. [IandC:10.1006/inco.1994.1024]. 2.2

[23] * Y. MANSOUR: *Learning Boolean functions via the Fourier transform*, pp. 391–424. Kluwer Academic Publishers, 1994. 3.3

[24] * C. MCDIARMID: On the method of bounded differences. In *Surveys in Combinatorics 1989*, pp. 148–188. London Mathematical Society Lecture Notes, 1989. 2.1, 3.1

[25] * R. SERVEDIO: On learning monotone DNF under product distributions. In *Proc. 14th Ann. Conf. on Computational Learning Theory (COLT'01)*, volume 2111 of *Lecture Notes in Computer Science*, pp. 558–573. Springer, 2001. [COLT:3j42gw4570jb08yt]. 1.1, 2.2

[26] * L. VALIANT: A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984. [CACM:1968.1972]. 1.1, 1.2, 3.1

[27] * K. VERBEURGT: Learning DNF under the uniform distribution in quasi-polynomial time. In *Proc. 3rd Ann. Workshop on Computational Learning Theory (COLT '90)*, pp. 314–326. Morgan Kaufmann, 1990. [ACM:92571.92657]. 1.1, 2.2

AUTHORS

Jeffrey C. Jackson
Department of Mathematics and Computer Science
Duquesne University
Pittsburgh, PA 15282, USA
jacksonj@duq.edu
http://www.mathcs.duq.edu/~jackson

Rocco A. Servedio
Department of Computer Science
Columbia University
New York, NY 10027, USA
rocco@cs.columbia.edu
http://www.cs.columbia.edu/~rocco

ABOUT THE AUTHORS

JEFFREY C. JACKSON has a distinctive educational background, having received his B. S. from Oral Roberts University and his Ph. D. from Carnegie Mellon, where Merrick Furst was his advisor. He has been a member of the faculty of Duquesne University since 1995, where he is currently chair of the Department of Mathematics and Computer Science. Jeff has also been a software engineer and manager in both the aerospace and dot-com industries and is the author of the textbook Web Technologies: A Science Computer Perspective. He is the proud father of four children (think about his last name for a moment and you'll know why he and his wife didn't stop at three).

ROCCO A. SERVEDIO received his B. S., M. S. and Ph. D. from Harvard University, where his Ph. D. was supervised by Leslie Valiant. For a change of pace, he then held an NSF postdoc at Harvard University, where he was supervised by Leslie Valiant. Since 2003 he has been an assistant professor at Columbia University in the Department of Computer Science He is interested in computational learning theory and computational complexity, and has received the NSF Career award and a Sloan Foundation Fellowship. He enjoys spending time with his family and hopes to have dinner with Herman Melville in the afterlife.