CS 6817: Special Topics in Complexity Theory	**Spring 2025**

Lecture 12: March 4, 2025

Lecturer: Eshan Chattopadhyay *Scribe: Shaleen Baral*

1 Disjunctive Normal Form

We start by recalling what a disjunctive normal form is.

Definition 1.1 (DNFs)**.** *A **DNF** (**disjunctive normal form**) formula over Boolean variables $x_1,\ldots,x_n$ is defined to be a logical OR of terms, each of which is a logical AND of literals. A **literal** is either a variable x_i or its logical negation $\overline{x_i}$. The number of literals in a term is called its **width**. We will identify a DNF formula with the Boolean function $f:\{0,1\}^n \to \{0,1\}$ it computes.*

Definition 1.2 (size, width)**.** *The **size** of a DNF formula is its number of terms. The **width** is the maximum width of its terms.*

Furthermore, note that the input length is fixed for a DNF as it is a formula over finitely many Boolean variables. So, to recognize an arbitrary $L \subseteq \{0,1\}^\star$ we would need multiple DNFs, one for each possible size of the input. This is the sense in which DNFs are a *non-uniform model of computation*. More formally,

Definition 1.3 (Non-Uniform Models of Computation)**.** *A family of DNFs $\{C_n\}_{n\geq 0}$ computes a language $L \subseteq \{0,1\}^\star$, if the following holds:*

$$\forall n \geq 0. \quad x \in \{0,1\}^n, \quad L(x) = C_n(x).$$

Additionally, any language $L \subseteq \{0,1\}^n$ naturally corresponds to a Boolean function,

$$f(x) = \mathbb{1}\{x \in L\}.$$

Note also that any Boolean-valued function admits a DNF representation.

Lemma 1.4. *Any $f:\{0,1\}^n \to \{0,1\}$ can be computed by a DNF of size at most 2^n and width at most n.*

Proof. We create a term T_x for each of the 2^n possible inputs $x \in \{0,1\}^n$. The term T_x will contain the literal x_i if $x_i = 1$ and $\overline{x_i}$ if $x_i = 0$.

□

2 Spectral Concentration for DNFs

Theorem 2.1. *Suppose $f:\{0,1\}^n \to \{0,1\}$ that is computable by a width w DNF. Then we have*

$$\boldsymbol{I}(f) \leq 2w.$$

Proof. Recall the sensitivity of f at x,

$$\begin{aligned} s(f,x) &= \#\text{neighbors of } x \text{ on the Hamming cube that are colored differently by } f \\ &= \sum_{y\in\{0,1\}^n} \mathbb{1}\{\Delta(x,y)=1\} \cdot \mathbb{1}\{f(y) \neq f(x)\} \end{aligned}$$

For convenience, we define

$$s_0(f,x) = s(f,x) \cdot \mathbb{1}\{f(x) = 0\},$$
$$s_1(f,x) = s(f,x) \cdot \mathbb{1}\{f(x) = 1\}.$$

Then, consider,

$$\begin{aligned}\mathbf{I}(f) &= \mathbb{E}_x[s(f,x)] \\ &= \mathbb{E}_x[s_0(f,x) + s_1(f,x)] \\ &= \mathbb{E}_x[s_0(f,x)] + \mathbb{E}_x[s_1(f,x)].\end{aligned}$$

Note next that,

$$\begin{aligned}\sum_x s_0(f,x) &= \sum_x s(f,x) \cdot \mathbb{1}\{f(x) = 0\} \\ &= \sum_x \sum_y \mathbb{1}\{\Delta(x,y) = 1\} \cdot \mathbb{1}\{f(y) \neq f(x)\} \cdot \mathbb{1}\{f(x) = 0\} \\ &= \sum_y s(f,y) \cdot \mathbb{1}\{f(y) \neq 1\} \\ &= \sum_y s_1(f,y).\end{aligned}$$

Since x is uniformly distributed, this then implies that the expectations above are equal,

$$\mathbf{I}(f) = 2\mathbb{E}_x[s_1(f,x)].$$

So it suffices to show that $\mathbb{E}_x[s_1(f,x)] \leq w$. If $f(x) = 1$ then at least one term T in the DNF representation of f must be made true by x. Note that if you change the value of a literal x_i that isn't present in the term T, the value of $f(x^{\oplus i})$ will still be 1. Thus, any y such that $f(y) = 0$ and $\Delta(x,y) = 1$ must differ from x in one of the literals present in T. Since there are at most w literals in T, we note that $s_1(f,x) \leq w$. Thus,

$$\mathbf{I}(f) \leq 2w.$$

□

There are a few immediate corollaries from this.

Corollary 2.2. *Suppose $f : \{0,1\}^n \to \{0,1\}$ is computable by a width w DNF, Then we get that $W^{\geq k}(f) < \epsilon$ where $K = 2w/\epsilon$ and $\epsilon > 0$.*

Proof. Recall that the Fourier spectrum of f is ϵ-concentrated on degree up to $\mathbf{I}(f)/\epsilon$ and use the fact that $\mathbf{I}(f) \leq 2w$.

□

Corollary 2.3. *PAC-learning for width w-DNF in the random example model with sample time $n^{O(w/\epsilon)}$.*

Proof. This follows from using the Low-Degree Algorithm with $k = 4w/\epsilon$ and noting that f is $\epsilon/2$ concentrated on degree up to $2\mathbf{I}(f)/\epsilon$.

□

Next, we will show that a small DNF is well-approximated by a narrow DNF. The intuition here is that removing a single term T of a DNF only changes the entire DNFs value on at most $1/2^w$ fraction of inputs (the inputs that makes all terms in T true). So, the underlying idea is to prune high-width terms of our DNF.

Lemma 2.4 (small to narrow)**.** *Suppose $f : \{0,1\}^n \to \{0,1\}$ computable by size s DNF. Then there exists $g : \{0,1\}^n \to \{0,1\}$ such that g is computable by width $\log(s/\epsilon)$ DNFs and*

$$\Pr_x(f(x) \neq g(x)) = \operatorname{dist}(f,g) \leq \delta.$$

Proof. Let $w = \log(s/\delta)$. Then let $g = \vee_{i=1}^{s'} T_{s_i}$ be the Boolean function obtained from $f = \vee_{i=1}^{s} T_i$ by removing all terms of width $> w$. Since every term in the DNF of g is a term in the DNF representation of f, we note that if $g(x) = 1$ then $f(x) = 1$. Furthermore note that for a T_i with widh $> w$, we have $\Pr(T_i = 1) \leq 2^{-w}$. There are at must s such T_i, so using a union bound

$$\Pr_x(\exists\, T_i = 1, \text{ such that width of } T_i \text{ is } > w) \leq s \cdot 2^{-w} \leq \delta.$$

Note that $g(x) \neq f(x)$ only when a term that is present in f but not in g is made true by x. Thus,

$$\Pr_x(g(x) \neq f(x)] = \Pr_x(\exists T_i = 1 \text{ with } T_i \text{ is } > w) \leq \delta.$$

□

This has ramifications with respect to concentration of and our ability to learn f.

Lemma 2.5. *Suppose the Fourier spectrum of $g : \{0,1\}^n \to \{0,1\}$ is ϵ_1-concentrated on $\mathcal{F}$ such that $f : \{0,1\}^n \to \{0,1\}$ satisfies $\|f-g\|_2^2 \leq \epsilon_2$. Then the Fourier spectrum of f is $2\cdot(\epsilon_1+\epsilon_2)$ concentrated on $\mathcal{F}$.*

Proof. Using the fact that $(a+b)^2 \leq 2(a^2+b^2)$, we obtain for any $S \in \mathcal{F}$

$$\hat{f}(S)^2 \leq 2\left(\hat{g}(S) + (\hat{f}(S) - \hat{g}(S))^2\right)$$

Summing over all $S \in \mathcal{F}$, we obtain

$$\sum_{S\in\mathcal{F}} \hat{f}(S) \leq 2\left(\sum_{S\in\mathcal{F}} \hat{g}(S)^2 + \sum_{S\in\mathcal{F}} (\hat{f}(S) - \hat{g}(S))^2\right) \leq 2(\epsilon_1 + \epsilon_2).$$

□

Corollary 2.6. *Suppose $f : \{0,1\}^n \to \{0,1\}$ is computable by a size s DNF. Then $W^{\geq K}(f) \leq \epsilon$ for $K = O\left(\frac{1}{\epsilon}\log\left(\frac{s}{\epsilon}\right)\right)$.*

Proof. Note that, by Lemma 2.4, f is $\epsilon/4$-close to a g with width $\log(4s/\epsilon)$. This gives us $\|f-g\|_2^2 = \operatorname{dist}(f,g) \leq \epsilon/4$. Note that, by Corollary 2.2, the g is $\epsilon/4$-concentrated for $\epsilon = 8\log(4s/\epsilon)/\epsilon$. Then, by Lemma 2.5, we get that

$$W^{\geq K}(f) \leq 2\left(\frac{\epsilon}{4} + \frac{\epsilon}{4}\right) = \epsilon.$$

□

Corollary 2.7. *PAC-learning for size s DNF with sample complexity $n^{O(1/\epsilon \log(s/\epsilon))}$*

Proof. Again, this follows from using the Low-Degree Algorithm with $k = \mathcal{O}(\log(s/\epsilon)/\epsilon)$. □