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

Lecture 11: Goldreich-Levin Algorithm and DNFs

Lecturer: Eshan Chattopadhyay *Scribe: Ellie Fassman*

1 Goldreich-Levin Algorithm

The Goldreich-Levin Algorithm is a randomized procedure to find heavy Fourier coefficients of a Boolean function f efficiently using query access to f. The setup is that we have query access to $f : \{-1,1\}^n \to \{-1,1\}$ and a parameter δ which is essentially defines the granularity of the sieve we are using to find the heavy Fourier coefficients. The goal is to output a list $\tilde{L}_f = \{S_1, \ldots, S_m\}$ such that with probability at least $\frac{2}{3}$, the following two properties hold:

- If $|\hat{f}(S)| \geq \delta$ then $S \in \tilde{L}_f$ (PROPERTY 1).
- If $S \in \tilde{L}_f$ then $|\hat{f}(S)| \geq \frac{\delta}{2}$ (PROPERTY 2).

To help us with the description of this algorithm, let us define $B_{k,S} \coloneqq \{T : T \cap [k] = S\}$ for all $k \in [n]$ and $S \subseteq [k]$. Note that $|B_{k,S}| = 2^{n-k}$, $B_{0,\emptyset}$ is power set of n, and the weight of a bucket $B_{k,S}$ is $w(B_{k,S}) = \Sigma_{T \in B_{k,S}} \hat{f}(T)^2$.

Algorithm 1 Goldreich-Levin Algorithm

```
Initialize 𝓑 = B_{0,∅} (𝓑 is a collection of buckets).
while there exists some B_{k,S} ∈ 𝓑 such that B_{k,S} contains more than one set do
    Note that B_{k,S} = B_{k+1,S} ∪ B_{k+1,S∪{k+1}}. Let B^0 := B_{k+1,S}, B^1 := B_{k+1,S∪{k+1}}.
    Remove B_{k,S} from 𝓑.
    Measure w(B^0), w(B^1) with accuracy δ^2/4.
    Add B^i to 𝓑 if w(B^i) ≥ δ^2/2.
end while
Output 𝓑.
```

By running this algorithm we can represent the sieving of as a complete binary tree. This tree has height at most n since there are 2^n buckets at the root, and each layer we make a binary split. Additionally, we can make the following observation:

Observation 1.1. *At any level k, the buckets are disjoint.*

Corollary 1.2. *The number of heavy buckets at any level k is $\leq \frac{4}{\delta^2}$.*

The corollary above can be shown by Parseval's Theorem, in addition to the fact that if B is a heavy bucket, $w(B) \geq \frac{\delta^2}{2} - \frac{\delta^2}{4} = \frac{\delta^2}{4}$ where $\frac{\delta^2}{2}$ is the threshold weight for heaviness and $\frac{\delta^2}{4}$ is the accuracy. Thus we can see that the total number of heavy buckets is $n\frac{4}{\delta^2} + 1$ (the $+1$ term accounts for the root node).

We are measuring at most $\leq \mathcal{O}(\frac{n}{\delta^2})$ buckets and the buckets are estimated to $\frac{\delta^2}{4}$ accuracy with $\mathcal{O}(\frac{1}{\delta^4}\log(\frac{1}{\gamma}))$ samples. Note that γ is the confidence parameter where $\gamma = \frac{1}{c}\frac{\delta^2}{n}$, for some large constant c. Thus, the run time of this algorithm can be bounded by $O(n(\log(n) + \log(1/\delta))/\delta^6)$.

To prove the correctness of this algorithm, we must show that PROPERTY 1 and PROPERTY 2 hold. We assume that all Fourier coefficients are estimated within the accuracy bounds (adding the error via union bound to the failure probability of the algorithm).

Proof. Suppose $|\hat{f}(S)| \geq \delta$. We are trying to show that if the Fourier coefficient is heavy, then it is in the final output set. We know that $S \in B_{0,\emptyset}$. $S \in \tilde{L}_f$ unless it has been in an eliminated bucket (one that is deemed as "light"), which means $w(B_{\{S\}})<\frac{\delta^2}{2}$. However note that because $w(B_{\{S\}}) \geq \hat{f}(S)^2 \geq \delta^2$, and we measure with accuracy $\delta^2/4$, the measurement will be at least $\delta^2 - \frac{\delta^2}{4} = \frac{3}{4}\delta^2 > \frac{\delta^2}{2}$. So set S will not be eliminated. Thus $S \in \tilde{L}_f$ (PROPERTY 1).

Now suppose $S \in \tilde{L}_f$. We are trying to show that if S is in the final set, then it's Fourier coefficient is heavy. If $S \in \tilde{L}_f$, then $\hat{f}(S)^2 = w(B_{\{S\}}) \geq \frac{\delta^2}{4}$. Thus $|\hat{f}(S)| \geq \frac{\delta}{2}$ (PROPERTY 2). □

Now, regarding the true weight of the buckets:

Claim 1.3. $w(B_{k,S}) = \mathbb{E}_{y_1,y_2\sim\{-1,1\}^k, z\sim\{-1,1\}^{n-k}}[f(y_1,z)f(y_2,z)\chi_S(y_1)\chi_S(y_2)]$

Now let's prove the claim.

Proof. $\mathbb{E}_{y_1,y_2\sim\{-1,1\}^k, z\sim\{-1,1\}^{n-k}}[f(y_1,z)f(y_2,z)\chi_S(y_1)\chi_S(y_2)]$
$= \mathbb{E}_{y_1,y_2\sim\{-1,1\}^k, z\sim\{-1,1\}^{n-k}}[\Sigma_{T_1,T_2\subseteq[n]}\hat{f}(T_1)\hat{f}(T_2)\chi_{T_1}(y_1,z)\chi_{T_2}(y_2,z)\chi_S(y_1)\chi_S(y_2)]$
$= \Sigma_{T_1,T_2\subseteq[n]}\hat{f}(T_1)\hat{f}(T_2)\mathbb{E}_{y_1}[\chi_{T_1\cap[k]}(y_1)\chi_S(y_1)]\mathbb{E}_{y_2}[\chi_{T_2\cap[k]}(y_2)\chi_S(y_2)]\mathbb{E}_z[\chi_{T_1\cap[k+1,\ldots,n]}(z)\chi_{T_2\cap[k+1,\ldots,n]}(z)]$

Where in the last line, the first expectation (over y_1) has nonzero terms only when $T_1 \cap [k] = S$ and the second expectation (over y_2) has nonzero terms only when $T_2 \cap [k] = S$. This effectively forces the same intersection for the first k coordinates, which is S. Similarly for the last expectation (over z) the nonzero terms effectively force the same intersection for the last $k+1,\ldots,n$ coordinates, which we call W.

Thus the above terms $= \Sigma_{W\subseteq\{k+1,\ldots,n\}}\hat{f}(S\cup W)^2 = w(B_{k,S})$ which proves the claim. □

2 DNFs

The next complexity class we care about are DNFs (disjunctive normal form) which are the OR of terms and AND of literals.

For example: $(x_1 \wedge \bar{x_2} \wedge x_4) \vee (x_2 \wedge \bar{x}_{13} \wedge x_{21} \wedge x_{22}) \vee \ldots$

The complexity measures of DNFs are **width**: the max width of a term in the DNF, and **size**: the number of terms in the DNF.

Claim 2.1. *Any $f : \{0,1\}^n \to \{0,1\}$ can be computed by a width $w \leq n$, size $s \leq 2^n$ DNF.*

Proof. Go through the truth table of this function f, create a clause for every time the function evaluates to 1. OR all of these clauses. □

Next class we will discuss polynomial-sized DNFs.