Lecture 5: Local List Decoding

Topics in Pseudorandomness and Complexity theory (Spring 2017)
Rutgers University
Swastik Kopparty
Scribes: Harsha Tirumala

In this lecture, we will see how to efficiently find all low-degree polynomials that approximately ($> 99\%$) fit a given set of points (if they exist).

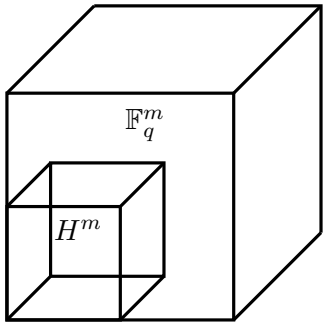
1 Revisiting Hardness Amplification

Given a function $f : \{0,1\}^n \to \{0,1\}$ which cannot be computed by any circuit of size S (i.e. f is worst-case hard for all circuits of size S), we want to produce an average-case hard function $\tilde{f} : \{0,1\}^{\tilde{n}} \to \{0,1\}$ such that: $\forall$ Circuits $\tilde{C}$ of size $\frac{S}{poly(n)}$

$$\Pr_{y \in \{0,1\}^{\tilde{n}}}[C(y) \neq f(y)] > 0.01$$

1.1 Constructing $\tilde{f}$

Pick a field $\mathbb{F}_q$ (q to be specified later). Next, pick $H \subset \mathbb{F}_q$ with $|H| = d < \frac{q}{2m}$ (m to be specified later).



Identify H^m with $\{0,1\}^n$ (via some bijection). So, we can view f as $f : H^m \to \{0,1\} \subset \mathbb{F}_q$ (constrains $|H|^m = 2^n$. Define $g : \mathbb{F}_q^m \to \mathbb{F}_q$ to be the low degree extension of f. So, g has an individual degree $< d$ (for each of its variables).

Assume $q = 2^t$. Identify $\mathbb{F}_q$ with $\{0,1\}^t$. Recall Hadamard encoding :
$Had : \{0,1\}^t \to \{0,1\}^{2^t}$

$< x, y >$ represents the $\mathbb{F}_2$ inner product between x, y.

The choice of Hadamard encoding is based on its distance property.
Specifically, strings $x, y \in \{0,1\}^t$ such that $x \neq y$ satisfy $\Delta(Had(x), Had(y)) \geq (0.5)2^t$.
Now, $\tilde{f} : \mathbb{F}_q^m x[2^t] \to \{0,1\}$ is given by :

$$\tilde{f}(x,i) = Had(g(x))_i \qquad (constraint : 2^{\tilde{n}} = q^m.2^t)$$

1.2 Proof of hardness of $\tilde{f}$

We will show that if $\tilde{f}$ can be approximately (99%) computed by small circuits then f can be computed (with 100% accuracy) by small circuits. Specifically, say circuit $\tilde{C}$ of size $\tilde{S}$ computes $\tilde{f}$ approximately; i.e.:

$$\Pr_{x \in \mathbb{F}_q{}^m, i \in [2^t]}[\tilde{C}(x,i) \neq \tilde{f}(x,i)] \geq 0.99$$

Then, we construct C of size $S = poly(n).\tilde{S}$ such that:

$$\forall x \in \mathbb{F}_q^m \;\; C(x) = f(x)$$

1.2.1 Circuit Candidate 1

On input $x \in H^m$, query $\tilde{C}(x,i)\ \forall \quad i \in [2^t]$

This ought to be the Hadamard encoding of $g(x)$ i.e. $Had(g(x))$. So, if the queries reveal $Had(r)$ for some $r \in H^m$, then output r. Observe that the above circuit works if $\tilde{C}$ is **always** correct. On the other hand, C will not work if $\tilde{C}$ is wrong on H^m (which almost always happens since $\tilde{C}$ may have 0.01 fraction errors)

1.2.2 Real Circuit

Step 1. Correcting Hadamard

We know

$$\Pr_{x \in \mathbb{F}_q{}^m, i \in [2^t]}[\tilde{C}(x,i) \neq \tilde{f}(x,i)] \geq 0.99$$

Since the Circuit $\tilde{C}$ is 0.99 close to f, the following guarantee on $\tilde{C}$ can be given (over the choice of x):

$$\Pr_x[\Pr_i[\tilde{C}(x,i) = \tilde{f}(x,i)] = 0.8] \geq 0.95$$

The reason being that no more than 0.2 fraction of errors can be made on a random x with more than 0.05 probability (else 0.99 approximation assumption collapses).

Circuit $C_0 : \mathbb{F}_q^m \to \mathbb{F}_q$

On input $x \in H^m$: query $v = \tilde{C}(x, i)$ for each $i \in [2^t]$
Find $r \in \{0,1\}^t$ such that $\Delta(Had(r), v) < (0.2)2^t$ and output r.

Observation 1. *The choice of 80% accuracy for $\tilde{C}(x, i)$ could have been any value greater than 75%. This is essential for unique decoding for Hadamard encoding which has a distance of 0.5*

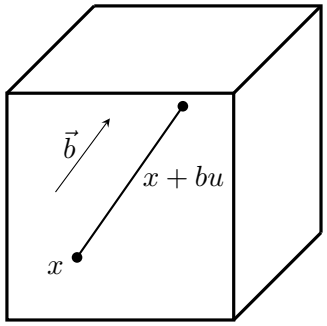
Now we know

$$\Pr_{x \in \mathbb{F}_q}[[C_0(X) = g(x)] \geq 0.95]$$

Step 2. Correcting the polynomial encoding

Circuit C:

On input x, Pick $b \in \mathbb{F}_q^m$ uniformly at random. Define $S = \{(u, C_0(ub + x)) : u \in \mathbb{F}_q(u \neq 0)\} \subset \mathbb{F}_q^2$. Note that S is the restriction of C_0 to a random line passing through x.



Run the univariate polynomial decoding algorithm on S for error fraction 0.2 to get $P(U) \in \mathbb{F}_q[U]$ and output $P(0)$

Let L be the line through x in direction b. i.e. $:L = \{x + ub, u \in \mathbb{F}_q\}$. Then $A(U) = g(x + Ub) \in \mathbb{F}_q(U)$ of degree $\leq d.m < \frac{q}{2}$. $C_0(x + Ub)$ ought to equal $A(U)$.

Claim 2. *Over the internal randomness of the above circuit C we have :*
$\forall\, x \;\; Pr[C(x) = g(x)] \geq 0.75$

Let the set of erroneous inputs y for C_0 be $E = y \in \mathbb{F}_q^m : C_0(y) \neq g(y)$. We know that:
$\frac{|E|}{q^m} \leq 0.05$

Bad event B: More than 20% of $\{ub + x : u \in \mathbb{F}_q\}$ lie in E.

Observation 3. *For any fixed $u \neq 0$, $ub + xis$ uniformly distributed in $\mathbb{F}_q$.*

$\Rightarrow$ for fixed u, $Pr[ub + x] \leq 0.05$

$$\mathbb{E}[|u| : ub + x \in E] \leq 0.05(q-1)$$

$$Pr[|u| > (0.2)(q-1) : ub + x \in E] \leq \frac{0.05(q-1)}{0.2(q-1)} = \frac{1}{4} \qquad Markov's Inequality$$

$\Rightarrow Pr[B] \leq 0.25$

If the bad event B is avoided, then S has at most 0.2 disagreements with the graph of $A(U) = g(x + Ub)$. So, polynomial decoder will recover $A(U)$. And the output is $A(0) = g(x)$

Repeat this 0.75 accurate circuit enough times with fresh random bits so that the majority answer is correct. The error upper bound in this case would be q^{-m}, Then we can conclude that $\exists$ a setting of the random bits which can be hardwired to always give the right answer.

This concludes the proof that if $\tilde{f}$ can be approximately (99%) computed by small circuits then f can be computed (with 100% accuracy)by small circuits.

1.2.3 Parameter Setting

The list of constraints to satisfy the above construction are:
1. $d^m = 2^n$
2. $q = 2^t$
3. $d < \frac{q}{m}$
4. $\tilde{n} = (m+1)\log q$

The final setting of parameters based on the above constraints are: $d = n$
$m = n \log n$
$q = 2md = \frac{2n^2}{\log n}$
$\tilde{n} = (m+1)\log q = \frac{n}{\log n}(\log \frac{2n^2}{\log n}) = \Theta(n)$

2 ϵ fraction approximation

Given $(x_i, y_i)_{i=1}^n \in \mathbb{F}^2$ (distinct points), efficiently find all polynomials P of degree $\leq d$ such that $P(x_i) = y_i$ for $\geq \epsilon$ fraction of $i \in [n]$.

2.1 Berlekamp Welch algorithm

Observation 4. *The Berlekamp Welch algorithm only outputs one candidate polynomial $F(x)$. Hence, we proceed to the following algorithm which outputs the entire list of candidate polynomials.*

A detailed description/analysis of the Berlekamp Welch algorithm can be found here

Algorithm 1: Berlekamp Welch algorithm

Input : $(x_i, y_i)_{i=1}^n \in \mathbb{F}^2$
Output : A polynomial $F(x)$ (if one exists) of low degree (d) that satisfies $F(x_i) = y_i$ for $\geq \epsilon$ fraction of $i \in [n]$.
1. Find polynomials $E(x) \neq 0$ (of degree$< \frac{n-D}{2}$) and $N(x)$ (of degree $< \frac{n+D}{2}$) such that

$$E(\alpha).r(\alpha) = N(\alpha) \qquad \forall \alpha \in S$$

2. If the $E(x)$ does not divide $N(x)$ then no such $F(x)$ exists (i.e. low degree polynomial that has $\geq \epsilon$ fraction agreements.
Otherwise output the polynomial $F(x) = \frac{N(x)}{E(x)}$

2.2 Sudan's Algorithm

Step 1. Interpolate non-zero $Q(X, Y)$ of degree $a, \frac{n}{a}$ respectively in Y, X such that $Q(X, Y)$ vanishes on all points.

$$Q(X, Y) = \sum_{i=0}^{a} \sum_{j=0}^{\frac{n}{a}} c_{ij} X^i Y^j$$

Where we require Q to vanish on $\{(x_i, y_i)\}_{i=1}^n$. This is a homogeneous linear constraint on $\{c_{ij}\}$. So, there are a total of n constraints imposed on $(a+1)(\frac{n}{a}+1) > n$ variables. So, there exists a nonzero solution to this system.

Suppose $Y = P(X)$ agrees with $\geq \epsilon$ fraction of the points. Then $h(X) = Q(X, P(X))$ vanishes on $(\epsilon)n$ points. From the definition of Q we have $deg(Q) \leq \frac{n}{a} + a.d$ (Since $Y = P(X)$ has degree d). So, if $deg(h) < (\epsilon)n$ then h is identically 0. Choose $a = \sqrt{\frac{n}{d}}$. Then, $deg(h) \leq 2\sqrt{nd}$. So, if $2\sqrt{\frac{d}{n}} < \epsilon$ then $h(X) = Q(X, P(X)) = 0$.

If $f(Y)$ is a polynomial and $f(a) = 0$ then $(Y - a)|f(Y)$. Analogous to this, we have in this case, $Y - P(X)|Q(X, Y)$

Step 2.

Factor $Q(X, Y)$ and output $P(X)$ for any factor of the form $Y - P(X)$. If $\frac{n}{a} + ad < (\epsilon)n$ then this algorithm outputs a list of at most a polynomials such that:
every $P(X)$ with $\geq (\epsilon)n$ fraction agreements with the input is in this list.

2.2.1 Typical Parameter setting

$a = \frac{2}{\epsilon}$ is a typical setting and for this to work :
$\frac{2d}{\epsilon} < \frac{(\epsilon)n}{2}$ i.e. $d < \frac{(\epsilon)^2 n}{4}$

3 Multi variate list decoding

Let $g : \mathbb{F}_{||}^{\geqslant} \to \mathbb{F}_{||}$ be a polynomial of total degree D. Given access to $C : \mathbb{F}_{||}^{\geqslant} \to \mathbb{F}_q$ such that $\underset{x \in \mathbb{F}_q^m}{Pr}[C(x) = g(x)] \geq \epsilon$. Want to output circuit $C_1, C_2, ..C_L$ such that :

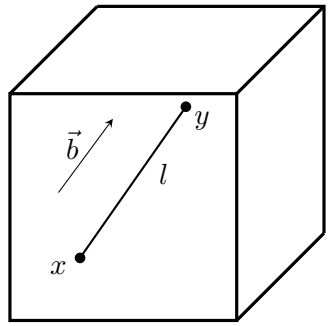
$$\exists i \in [L] \quad C_i(x) = g(x) \quad \forall x \in \mathbb{F}_q^m$$

in time $poly(m, q, \frac{1}{\epsilon})$ where $|C_i| \leq poly(m, q, \frac{1}{\epsilon})|C|$ and $|L| \leq poly(m, q, \frac{1}{\epsilon})$. Here $D < O(\epsilon^2 q)$.

3.1 Sudan's Algorithm

Step 1.

Pick a random $y \in \mathbb{F}_q^m$. For each $a \in \mathbb{F}_q$, pretend that $g(y) = a$



Trial 1

Pick $b \in \mathbb{F}_q^m$ uniformly at random. Let l be the line $x + bU$. Look at $C|_l$. We expect ϵ fraction agreement with the univariate poly $g|_l$. Run Sudan's algorithm to find $P_1(U), P_2(U), ...P_L(U)$ and then output $P_1(0), P_2(0), ...P_L(0)$

$C_{y,a}(x)$ - Let l be the line passing through x, y i.e. $l = x + (y - x)U$. Look at $C|_l$. Let $S = \{(u, C(x + u(y - x)) : u \in \mathbb{F}_q \setminus 0\}$. Find $P_1(U), P_2(U), ...P_L(U)$ such that agreements $(P_i, s) > 2\epsilon$. Now find j such that $P_j(1) = a$ (if any unique j exists) and output $P_j(0)$

Claim 5. $\underset{y}{Pr}[C_{y,g(y)}(x) = g(x)] \geq 0.99$