

Lecture 19: Local decoding, list decoding and hardness amplification.

Lecturer: Kristoffer Arnsfelt Hansen

Scribe: Claus Thrane

Last time we introduced a notion of *hardness amplification*, *error correcting codes* and *local decoding*, today we introduce the notion of *local list decoders*.

Local Decoding

We recall some of the previous definitions:

Definition 1 (Local Decoding) Let $E : \Sigma^n \rightarrow \Gamma^m$ be a error correcting code (ECC). A local decoder is a randomized algorithm D , handling ρ errors (i.e. with error ratio ρ) that, given :

- random access to y s.t. $\exists x \in \Gamma^n : \Delta(y, E(x)) < \rho$, and
- an index $j \in \mathbb{N}$

runs in time $(\log(m))^{\mathcal{O}(1)}$ and outputs x_j with probability at least $\frac{2}{3}$

Moreover, recall that when seeking *hardness amplification*, we view a function $f : \{1, 0\}^n \rightarrow \{1, 0\}$ as binary string in $\{1, 0\}^N$ for $N = 2^n$ and denote by \hat{f} , the encoded function $\hat{f} = E(f)$ for some ECC E .

Walsh-Hadamard Local Decoding

Recall the Walsh-Hadamard code $\text{WH} : \{1, 0\}^n \rightarrow \{1, 0\}^{2^n}$ s.t. $\text{WH}(x) = z$, $z_y = x \cdot y \pmod{2}$ ¹

Theorem 2 For $\rho < \frac{1}{4}$ There is a local decoder for WH

Proof The proof is given by providing an algorithm:

- On input
 - $j \in \{1, \dots, n\}$, and (random access to)
 - $g : \{1, 0\}^n \rightarrow \{1, 0\}$ s.t. $\Pr_y[g(y) = x \cdot y \pmod{2}] \geq 1 - \rho$
 where $\rho < \frac{1}{4}$ and $x \in \{1, 0\}^n$ being the encoded string, and j its j^{th} bit.
- Choose (y at random) $y \in_R \{1, 0\}^n$ and output $g(e_j + y) + g(y) \pmod{2}$

Where e_j denotes a vector in $\{1, 0\}^n$ with value $\mathbf{0}$ in all coordinates except the j^{th} which is $\mathbf{1}$ and $y + e_j$ denotes component-wise addition modulo 2.

¹Here z_y denotes the y^{th} index of the string z and $x \cdot y$ denotes the inner product of x and y i.e. $x \cdot y = \sum_{i=1}^n x_i y_i$

Analysis: Since y is chosen uniformly at random, both y and $y + e_j$ are uniformly random. We get that:

$$\Pr[g(e_j + y) = x \cdot (e_j + y) \wedge g(y) = x \cdot y] \geq 1 - 2\rho \geq \frac{1}{2} + \epsilon \text{ for } \epsilon > 0$$

When this occurs we have

$$g(y) + g(y + e_j) = x \cdot y + x \cdot (y + e_j) = 2(x \cdot y) + x \cdot e_j = x \cdot e_j \pmod{2}$$

Since $x \cdot e_j$ is desired result, the algorithm outputs the correct value with probability $1 - 2\rho$ as claimed \square

As usual the success probability can be amplified (boosted) by the majority vote of successive trials.

Reed -Muller Local Decoding

We will now change the definition of the Reed-Muller code to allow for a local decoder. Before we interpreted the input as the coefficients of a polynomial. Now we will instead interpret the input as evaluations of the polynomial on the same number of inputs as there are coefficients. Thus the task of the local decoder will just be to evaluate the polynomial on a given input.

Note that by interpolation we will still have a polynomial time encoding algorithm with the changed representation.

Theorem 3 *For every field \mathbb{F} , numbers d, ℓ there is a $\text{poly}(|\mathbb{F}|, \ell, d)$ -time local decoder for the Reed-Muller code, handling error rate $\rho = (1 - \frac{d}{|\mathbb{F}|-1})/6$*

Proof The proof is given by presenting such an algorithm. Assuming (random) access to a function $f : \mathbb{F}^\ell \rightarrow \mathbb{F}$ s.t. $f(x) = P(x)$ for some degree d polynomial P on at least a $(1 - \rho)$ -fraction of the inputs².

- On input $x \in \mathbb{F}^\ell$, we wish to compute $P(x)$.
 1. Let $L_x = \{x + tz \mid t \in \mathbb{F}\}$ for a random $z \in \mathbb{F}^\ell$ (i.e. a random line through x)
 2. Query $f(x + tz)$ for all $t \neq 0$
 3. Run the Reed-Solomon (RS) decoding algorithm, on $\{(t, f(x + tz))\}$ to get $Q : \mathbb{F} \rightarrow \mathbb{F}$ (a univariate polynomial)
 4. output $Q(0)$

Analysis: For $t \neq 0$ we have that $x + tz$ is uniform random (because z is random). Hence

$$\Pr[f(x + tz) \neq P(x + tz)] \leq \rho$$

Hence

$$\mathbb{E}[|\{t \neq 0 : f(x + tz) \neq P(x + tz)\}|] \leq (|\mathbb{F}| - 1)\rho$$

²that is $\Pr_{x \in \mathbb{F}^\ell}[P(x) \neq f(x)] \leq \rho$

By Markov's inequality,

$$\Pr[\{t \neq 0 : f(x + tz) \neq P(x + tz)\} \geq 3(|\mathbb{F}| - 1)\rho] \leq \frac{(|\mathbb{F}| - 1)\rho}{3(|\mathbb{F}| - 1)\rho} = \frac{1}{3}$$

Note that

$$3(|\mathbb{F}| - 1)\rho = 3(|\mathbb{F}| - 1)\left(1 - \frac{d}{|\mathbb{F}| - 1}\right)/6 \leq \frac{|\mathbb{F}| - 1}{2} - \frac{d}{2}$$

When this is the case the RS decoding works and we will obtain the correct Q , i.e the restriction of P to the line L_x . Thus we compute the correct output with probability at least $\frac{2}{3}$. \square

List decoding

The following introduces *list decoders* which obtains a list of candidates as the possible source of a corrupted string. With this relaxation we will be able to decode with close to 1/2 error rate. The following theorem gives a bound on the number of candidate codewords of a corrupted codeword.

Theorem 4 (Johnson Bound) *If $E : \{1, 0\}^n \rightarrow \{1, 0\}^m$ is an ECC w. distance $\frac{1}{2} - \epsilon$, then every $y \in \{1, 0\}^m$ and $\delta \geq \sqrt{\epsilon}$ there exists at most $\frac{1}{2\delta^2}$ points $\{x_1, \dots, x_\ell\} \subseteq \{1, 0\}^n$ s.t. $\Delta(E(x_i), y) \leq \frac{1}{2} - \delta$ for all $1 \leq i \leq \ell$.*

For the proof see [1].

We will repeat everything done so far but now with list-decoding. The first step will be a list decoder for the Reed-Solomon code.

Reed-Solomon List Decoder

Theorem 5 *There is a poly-time algorithm that given, $(a_1, b_1), \dots, (a_m, b_m)$ returns a list of degree d polynomials G such s.t. $|\{i : G(a_i) = b_i\}| \geq t$ for some parameter $t > 2\sqrt{dm}$*

Proof We show that the theorem hold by providing the following algorithm:

1. Find a nonzero $Q(x, y)$ of degree $\leq \sqrt{dm}$ in x and $\leq \sqrt{m/d}$ in y . s.t. $\forall i \leq m : Q(a_i, b_i) = 0$
 Observe that Q is guaranteed to exist since, we can express the requirement $\forall i \leq m : Q(a_i, b_i) = 0$ as a homogeneous system of m linear equations with $(\sqrt{dm})(\sqrt{m/d}) > m$ variables, corresponding to the coefficients of Q . Since we have more variables than equations there must be a non-zero solution.
2. Factor $Q(x, y)$ in poly-time (We will use without proof that this can be done in polynomial time).
3. For every factor of the form $(y - P(x))$ output P if $|\{i : P(a_i) = b_i\}| \geq t$

Analysis Assume that G is s.t. $|\{i : G(a_i) = b_i\}| \geq t$. Then $Q(x, G(x))$ has $\geq t$ zeros. Observe that we have $\deg(G(x)) \leq \sqrt{dm} + d\sqrt{m/d} = 2\sqrt{dm} < t$. This means that $Q(x, G(x))$ must be identically 0, $Q(x, G(x)) = 0$, for all x .

Do polynomial division of $Q(x, G(x))$ by $(y - G(x))$ in y . Write $Q(x, y) = (y - G(x))A(x, y) + R(x, y)$ with $\deg_y(R(x, y)) < \deg_y(y - G(x)) = 1$. Hence $\deg_y(R(x, y)) = 0$ and we can write $Q(x, y) = (y - G(x))A(x, y) + R(x)$. Substituting $G(x)$ for y we have

$$0 \equiv Q(x, G(x)) = (G(x) - G(x))A(x) + R(x) = R(x)$$

and hence we have $Q(x, y) = (y - G(x))A(x, y)$ meaning that $(y - G(x))$ is a factor of Q . □

Local List Decoding

We can proceed to define local list decoding, which can be used for hardness amplification.

Definition 6 (Local list decoder) Let $E : \{1, 0\}^n \rightarrow \{1, 0\}^m$ be an ECC and let $\rho = 1 - \epsilon$ for $\epsilon > 0$. An algorithm D is a local list decoder for E if for every $x \in \{1, 0\}^n$ and $y \in \{1, 0\}^m$ s.t. $\Delta(E(x), y) \leq \rho$ there is an index $i_0 \leq \text{poly}(\frac{n}{\epsilon})$ s.t. for every $j \in \{1, \dots, n\}$ on input (i_0, j) and random access to y , D runs for $\text{poly}(\log(m)/\epsilon)$ time, and outputs x_j with probability $\geq 2/3$

In the following we will use the composed ECC $E = \text{WH} \circ \text{RM}$. We will show that E is local list decodable. This code can then be used for hardness amplification as follows.

Theorem 7 Let $S(n) \geq n$, if $f \in \mathcal{E}$ requires circuits of size $S(n)$ then there exists $\hat{f} \in E = \text{DTIME}(2^{O(n)})$ with hardness $H_{\hat{f}} \geq S(n/c)^{n/c}$, for some $c > 0$ and for all sufficiently large n .

Proof [sketch] Let $\hat{f} = E(f)$, and proceed similarly as we did for local decoding. we then hardwire the proper i_0 in the circuit w. We can hardwire the correct index i_0 into the list such that we can reconstruct the correct function f . □

The task is now to construct a local list decoder for the Walsh-Hadamard code and for the Reed-Muller code. The latter will use the list-decoder for the Reed-Solomon code. We first give a local list decoder for the Walsh-Hadamard code.

Theorem 8 The Walsh-Hadamard code has a local list decoder.

Proof [sketch] Assume random access to $r \in \{1, 0\}^m$. Using parameter t , let $\alpha_1, \dots, \alpha_t \in \{1, 0\}$ constitute a list index. Choose $v_1, \dots, v_t \in \{1, 0\}^n$ at random.

Consider some $x \in \{1, 0\}^n$ s.t. $\Pr_y[x \cdot y = r_y] \geq \frac{1}{2} + \epsilon$ also satisfying $\forall i : x \cdot v_i = \alpha_i$.

For $c \in \{1, 0\}^t$ define w_c as the linear combination $w_c = \sum_{i=1}^t c_i v_i$ of the vectors v_1, \dots, v_t given by c .

Now we have $x \cdot w_c = \sum_{i=1}^t x \cdot c_i v_i = \sum_{i=1}^t c_i \alpha_i$, and thus we already “know” the values $x \cdot w_c$.

Now, fix a coordinate i . For all $c \neq 0$, w_c is a uniform random vector (since all v 's are). Thus for a fixed $c \neq 0$, we have

$$\Pr[r_{(e_i+w_c)} = x \cdot (e_i + w_c)] \geq \frac{1}{2} + \epsilon$$

Every vector w_c gives a guess at x_i , $r_{(e_i+w_c)} - x \cdot w_c \pmod{2}$.

We take the majority guess at x_i over all w_c , $c \neq 0$, of $r_{(e_i+w_c)} - x \cdot w_c \pmod{2}$. The vectors w_c are not independent, however for $c \neq c'$, the pair of vectors w_c and $w_{c'}$ are independent. Hence, the w_c are pairwise independent, and this means we can make use of Chebychevs inequality. Using that it is possible to derive (For details see the notes from the course by Dieter van Melkebeek).

$$\Pr[\text{fail to guess } x_i] \geq \frac{1}{2^t \epsilon^2}$$

and hence by a union bound we have

$$\Pr[\exists i : \text{fail to guess } x_i] \geq \frac{n}{2^t \epsilon^2} \leq \frac{1}{3} \text{ for } t = \mathcal{O}(\log(\frac{n}{\epsilon^2}))$$

Observe that the list size we get from this decoder is of size $2^t = (\frac{n}{\epsilon^2})^{\mathcal{O}(1)}$ as required.

The running time is also $(\frac{n}{\epsilon^2})^{\mathcal{O}(1)} = (\frac{\log m}{\epsilon^2})^{\mathcal{O}(1)}$ as required.

□

At the lecture we also briefly covered the ideas behind the needed local list decoder for the Reed-Muller code. We refer to [1] for details.

References

- [1] *Computational Complexity: A Modern Approach*,
<http://www.cs.princeton.edu/complexity/>