CS 229r Essential Coding Theory, Lecture 18 Mar 30, 2017

Lecture 18

Instructor: Madhu Sudan *Scribes: Biswaroop Maiti*

1 Locality

In this lecture we will look at locality in the context of coding theory and get acquainted with definitions of notions such as:

1. Local Decodability
2. Local Testability
3. Local Recoverability

Locality is a notion relevant in algorithmic studies in general and is particularly relevant in coding theory. We will again look at Reed Muller Codes, this time as an example of a Locally Decodable Code.

In an algorithmic setting, we would usually have a function $f : \{0,1\}^n \to \{0,1\}^m$, and would require an algorithm to compute f. To read the input and write the output would require $O(n+m)$ time. Locality is a study of whether we can go beyond such obvious bounds. In other words, the question we consider is whether we can compute f in $o(m+n)$. This however requires a different set of guarantees, since not all of the input can be read and all of output can be written, which implies we have to allow approximations since exact computation is not possible. So, we allow that the function can be computed implicitly. That is, instead of f, we are allowed to compute

$$\tilde{f} : \{0,1\}^n \times [m] \to \{0,1\}$$
$$\tilde{f}(x,i) = f(x)_i$$

Here, we don't see m in the upper bound. In such scenarios, obvious bounds of $O(m+n)$ are off the table. We reduced the trivial bound from $\Omega(n+m)$ to $\Omega(n+\log m)$ The algorithm is allowed to probe certain positions of the argument as a block box. The algorithm queries with a set of positions and receives corresponding symbols. Here, we note that, alternatively these queries can be adaptive i.e. the algorithm can query one position at a time and its subsequent queries may depend on what responses it received till that time. However, we will consider only non-adaptive setting whereby the queries are sent in one shot and responses are received together. This is primarily because adaptive queries are not so useful in linear codes. Here, the descriptions of input, output are implicit. The algorithm A is provided with a blackbox / oracle access of the input x. We denote this by the notation $A^x(i)$, where i is given as regular input.

Now, if A is very sensitive to every bit or position of the input, and in this local model, it is not reading all of them, then a lot of information is not obtained by the algorithm. Therefore, we will not get a perfect guarantee. Instead we will get a guarantee that involves some error. We will guarantee that A computes $\tilde{f}$ on $\hat{x}$ that is not too far off from the actual input x in Hamming distance. We will state:

$\exists \hat{x} : \delta(\hat{x}, x) \leq \varepsilon,$ such that

$$\Pr[A^x(i) = \tilde{f}(\hat{x}, i)] \geq 2/3$$

Here, the probability is on the randomness of the algorithm A, and δ denotes the Hamming distance, as usual. We can set ε as small as we want, something like 0.01. In the study of locality, the measure of complexity is the number of times A probes x. Now, we may ask is this a good guarantee? In the domain of error correction, where the input is noisy and there may be sampling error, such a guarantee is quite reasonable.

We may consider various basic problems under this notion of locality.

- Encoding ✗
- Decoding ✓
- Testing Errors ✓

Clearly we cannot encode information with error. However, decoding and testing errors may seem reasonable in principle.

Remark Are there codes that are locally testable but not locally decodable or vice versa?

Is it possible to have a local algorithm A and ask if x is a valid codeword?

Definition 1. *l-local (non adaptive)*

Given input i, and an oracle access to input $x \in \{0,1\}^n$, an $l-$local algorithm $A^x(i)$ samples $S \sim P_i\binom{[n]}{l}$, makes (non adaptive) queries $X_S := \{x_i | i \in S\}$ and determines output based on (X_S, i).

Here, the notation $S \sim P_i\binom{[n]}{l}$ implies S is a sample drawn from the coordinates of x by A according to the distribution P_i and the size of the sample is t. Note that the distribution is indexed by i, i.e. it may depend on i. The output does not depend on the unread coordinates of x, that is, $X_{\bar{S}}$.

Remark We need the distribution P to be computable by some algorithm efficiently. However, in the current scope, we will ignore this. We will assume existence of a non uniform procedure that provides a vector signifying a distribution and allows the algorithm to efficiently sample X according to that distribution. here, our objective is show if l can be made as small as possible.

Remark If there is an adversary, should we not require that the number of choices such subsets of indices S be large?

Definition 2. $(\varepsilon, l)-$ ***local decoder***

D is defined to be an $(\varepsilon, l)-$ local decoder for a given code C with encoding function $E : \{0,1\}^k \to \{0,1\}^n$, if the following hold:

- *$D^x(i) : i \in [k]$ is an l local algorithm*
- *$\forall x : \delta(x, E(m)) \leq \varepsilon \cdot \delta(C)$, if we have:*

$$\forall i : \Pr[D^x(i) = m_i] \geq 2/3$$

Definition 3. $(\varepsilon, l)-$ ***local correcter***

D is defined to be an $(\varepsilon, l)-$ local correcter for a given code C with encoding function $E : \{0,1\}^k \to \{0,1\}^n$, if the following hold:

- *$D^x(i) : i \in [n]$ is an l local algorithm*
- *$\forall x : \delta(x, E(m)) \leq \varepsilon \cdot \delta(C)$, if we have:*

$$\forall i : \Pr[D^x(i) = E(m_i)] \geq 2/3$$

We can also talk about local list decoding but it is out of the scope of this class. We would like to have guarantees of the following kind:

1. If the coordinates read correspond to the correct codeword x, then $D(i)$ must decode (correct)

2. If the coordinates read correspond to a string within a Hamming ball of radius ε around x, then $D(i)$ must decode (correct) with constant probability like $2/3$. The Hamming ball here will be smaller than what we would get for usual error correcting codes.

3. If the coordinates read correspond to a string beyond a Hamming ball of radius ε around x, then there are no guarantees

Definition 4. $(\varepsilon, l)-$ ***local tester***
T is defined to be an $(\varepsilon, l)-$ local tester for a given code C with encoding function $E : \{0,1\}^k \to \{0,1\}^n$, if the following hold:

- *T^x is an l local algorithm*
- $\forall x \in C$,

$$\Pr[T^x : \mathsf{YES}] = 1$$

- $\forall x \notin C$,

$$\Pr[T^x : \mathsf{NO}] \geq \varepsilon \cdot \delta(x, C)$$

Remark Can we measure (approximately) the distance between a given point x and $E(m)$, where x is within a radius of ε of $E(m)$? If we have a local decoder and a local tester both for a code, then we measure the distance. There are very few codes for which we have both local tester and local decoder.

2 Example: Reed Muller Code

Recall Reed Muller codes are denoted by $\mathsf{RM}(q, m, d)$ where q is the field size i.e. the field is $\mathbb{F}_q$, m is the the number of variables, and d is the degree of the multi-variate polynomial. It is a linear code in which the message is represented as an m-variate polynomial f and the encoding is a vector of length $n = q^m$ which are the evaluations of the polynomial on the set of all points in the space. Reed Muller Code is an example of a Locally Decodable as well as a Locally Testable Code. This code has also been studied in the regime $d > q$, but we will study the case $d < q$. We will give a corrector and analyze it.

$$\mathsf{RM}[q, m, d] := \{f : \mathbb{F}_q^m \to \mathbb{F}_q | \deg f \leq d\}$$

- We assume the degree $d < q - 1$ and the size of the codeword is $n = q^m$.
- The distance of the code will be $q^m \cdot \frac{q-d}{q} = q^{m-1}(q - d)$.
- The dimension of the code is $k = \binom{d+m}{m} \geq \exp \min\{m, d\}$. The value of k is such because this is the number of coefficients of the degree d multivariate polynomial.
- Relative distance $\delta(C) = 1 - d/q$

We will implicitly consider $q \approx 2d, d = m$. The size of the codeword is therefore exponential in d. Our goal will be to decode or test in time polynomial in d. We will want locality in terms of q, d, m, which are roughly similar apart from the constants, whereas the size of the codeword is exponential in these parameters.

2.1 Main Idea

The main idea behind the locality in Reed Muller Code is to exploit local redundancies. We have the $m-$ary cube of q^m points of $\mathbb{F}_q^m$ in which we know the evaluations of the multi-variate polynomial. There is a small region which is corrupted and we want to decode or test at a point a, given as an input to the decoder. We are given the access to the evaluations of f in the whole cube, albeit with some corruptions, in the form of the codeword.

Usual decoding attempts to reconstruct all the coefficients which requires it to look at a large number of points. Here, we will not decode or correct the coefficients one point at a time. Instead, we will correct an evaluation point. Local correctability is a stronger concept than local decodability. So, this will suffice.

We have an oracle access over $f : \mathbb{F}_q^m \to \mathbb{F}_q$, and given a guarantee of the existence of a polynomial: $\exists p : \mathbb{F}_q^m \to \mathbb{F}_q$ such that the relative distance $\delta(p, f)$ is small.

The algorithm gets a point $a \in \mathbb{F}_q^m$ as an input coordinate and needs to answer the value of $p(a)$. We want to correct the value of $f(a)$ with $p(a)$.

We will take a line passing through a in this space, and as we vary t, the degree of the polynomial does not increase, it remains d. A line passing through a with slope b is denoted by $l_{a,b}$, where $a, b \in \mathbb{F}_q$ are vectors and $t \in \mathbb{F}_q$ is a scalar.

$$l_{a,b} := \{a + t \cdot b | t \in \mathbb{F}_q\}$$

$$a : (a_1, a_2, \ldots a_m)$$
$$b : (b_1, b_2, \ldots b_m)$$
$$a + tb = (a_1 + tb_1, \ldots a_m + tb_m)$$

When we take a polynomial p and restrict it to a line $l_{a,b}$, its degree can only be less. So, we get a function of degree at most d. We define the restriction as below:

$$p|_{l_{a,b}}(t) := p(a + tb)$$

This gives us the following fact.

Fact 5. *We have* $\forall l_{a,b} \forall p, \deg(p|_{l_{a,b}}) \leq \deg(p)$

$d < q - 1 \implies$ values of $p|_l$ are constrained. If we have a function with degree $\leq q - 1$, then not very assignment on the line is legitimate i.e. not all of those assignments will give a degree at most d. The set of polynomials of degree $d \leq q - 1$ is a linear space. But not all assignments on the line will give us the space of all degree d polynomial if $d \leq q - 1$.

This implies $\exists \lambda_1 \ldots \lambda_{q-1}$ such that

$$p|_l(0) = \sum_i \lambda_i p|_l(\eta_i)$$

$$p(a) = \sum \lambda_i p(a + \eta_i \cdot b)$$

We reduced the problem to $q-1$ points but we can get a random point on the line by choosing a random b, and look for evaluations of f on random points $a + \eta_i \cdot b$, and at most of the points f and p agree and have no corruptions. So, we need to query $q - 1$ points and gives us a $q - 1$ local algorithm. We will have to find the ε next.

Remark Reed Muller codes are $q - \mathsf{LDC}$ while their codewords are of size q^m. So, it does not exhibit non-trivial locality unless $m \geq 2$. The special case Reed Solomon codes have codewords of length q. Moreover, $(1/3, q) - LDC$ implies no correction.

2.2 Analysis

RM Decoder (Corrector): $\mathsf{RM}D^f(a)$

- Pick $b \in_\$ \mathbb{F}_q^m$ at random.
- **output:** $\sum_{i=1}^{q-1} \lambda_i f(a + \eta_i \cdot b)$

Lemma 6. $\mathsf{RM}D^f(a)$ *is* $(\varepsilon, \frac{1}{3} \cdot (q-1))$ *local for* $\varepsilon = \frac{1}{q-1}$.

Proof. First, we recognize that: $\forall \eta_i \neq 0$, $a + \eta_i b$ is independent of a, over the choice of the random b. This tells us:

$$\Pr_b[f(a + \eta_i b) \neq p(a + \eta_i b)] = \delta(f, p)$$

where $\delta(f, p)$ is the relative distance between f and p. We use union bound to bound the probability of the following bad event over all indices $i \in [q]$:

$$\Pr_b[\exists i \in [q] : f(a + \eta_i b) \neq p(a + \eta_i b)] \leq (q-1) \cdot \delta(f, p)$$

Now, we recognize that, when the above bad event does not happen, we get evaluations of p that agree on f, when we probe the codeword:

$$\begin{aligned} \sum_{i=1}^{q-1} \lambda_i f(a + \eta_i \cdot b) &= \sum_{i=1}^{q-1} \lambda_i p(a + \eta_i \cdot b) \\ &= p(a) \end{aligned}$$

We have oracle access to f and we are attempting to approximate it to p. We can decode correctly with probability $2/3$, if:

$$\begin{aligned} & (q-1) \cdot \delta(f, p) \leq \frac{1}{3} \\ \implies & \delta(f, p) \leq \frac{1}{3} \cdot \frac{1}{q-1} \\ \implies & \delta(f, p) \leq \frac{1}{3} \cdot \frac{1}{q-1} \cdot \delta(C) \end{aligned}$$

□

This is the ε for which RM is $(\varepsilon, q) - \mathsf{LDC}$.

Remark (Food for thought) Can we make this locality constant? We evaluated $q-1$ points and assumed them to be correct everywhere except a. But, we do not need so many values to be correct everywhere. We need only a fraction of them to be correct and use Markov inequality and remove this dependence on the evaluations being correct everywhere else.

We can decode arbitrarily half the distance locally. In the union bound, do we need to bound over so many points? If the degree is small, we should be able to tolerate a constant fraction of errors. May be, we can pick a line and do Reed Solomon decoding on the line?

2.3 Local Tester for RM Codes

Any Locally Decodable Code gives us a natural tester. However, it requires more analysis to show whether it works.

Tester for RM codes

- Pick a random line $l = l_{a,b}$
- Verify $\deg(f|_l) \leq d$

This test works for $\deg < q/2$ but the analysis is non trivial and we will talk not talk about it at this stage. However, we will describe a barrier, where it fails. If $\deg > q$, then this test does not work. For $q = 2^l, d = q/2$, this test fails for the following function.

A function for which the tester fails:

$$\begin{aligned} f(X_1 \dots X_m) &= X_1^d \cdot X_2^d \\ f|_{a+tb} &= (a_1 + tb_1)^d \cdot (a_2 + tb_2)^d \\ &= A \cdot t^{2d} + B \cdot t^d + C \\ &= A \cdot t + B \cdot t^d + C \end{aligned}$$

So, the polynomial is degree d on all points and this test accepts with probability 1.

Remark Low degree testing works great for $d < q/2$, but fails at $d = q/2$ as shown above. Local testability works for these codes but there are barriers both algebraic and probabilistic.

3 General Questions

We would like to know the influence of locality on the usual parameters viz. n, k of error correcting codes. There are two extreme regimes that we consider, viz: low queries and high rate.

3.1 Locally Decodable Code parameters

3.1.1 Low query:

For small values of l, we know a few results. For $l = 1$, LDC's don't exist beyond a finite message length. [KT00] For $l = 2$, we cannot have an efficient family of LDCs, there is exponential growth in n and k.

There are lower bounds for $l > 3$, that show sub-exponential growth. In general, we must have: $n \geq k^{1+1/l}$. On the positive side, for $l = 3$, there is construction that achieves $n \leq \exp(\exp(\sqrt{k}))$ [Yek08, Efr09]. If we impose the conditions $q = 2, l = 2$, then we can get $n = \exp(k)$ lower bound. For larger alphabets $q \approx l$, we get better constructions. [DG16]

3.1.2 High Rate

Here, we consider the regime of rate $R \approx 1 - \delta$. The minimum l we can get is $l = n^{o(1)} = \exp(\widetilde{\sqrt{k}})$ [KMRZS15]

3.2 Locally Testable Code parameters

3.2.1 Low Query

For $l = 3$, there is a construction that achieves $n = k\text{poly}\log k$. The rate here is effectively zero, but this is still considered very respectable.

3.2.2 High Rate

We consider the rate to be

$$R \approx 1 - \delta$$

There are codes that achieve $l = \exp(\text{poly}(\log\log k))$. We will talk about this in next few lectures.

4 Practical Application

In cloud storage, locality in the context of coding theory found a practical application. The errors relevant in this setting are erasures. Consider a number of servers in locations spread all over the globe together storing a symbol from a large alphabet. We consider the model of erasures, whereby we denote the failure of a server as an erasure. Erasure of such a server is a significant event and the data need to be recovered. This may be of two kinds:

- frequent single erasures
- occasional catastrophic multiple erasures

Clearly, if we can recover from multiple erasures, we will be able to recover from single erasures. However, we require the guarantee that single erasures can be recovered quickly. Multiple erasures can be less efficient. Such codes were defined as follows.

Definition 7. *(l, d) Locally Recoverable Code [GHSY12]*
A systematic code C of distance d with encoding

$$E : m \to (m, p)$$

(where p denotes the parities) is $(l, d)-$locally recoverable, with locality l if for every $\forall i \in [k]$ there exists a subset $S \subseteq [n] \setminus \{i\} : |S| \leq l$, such that:

$$(m, p)_S \implies m_i$$

References

[DG16] Zeev Dvir and Sivakanth Gopi. 2-server pir with subpolynomial communication. *Journal of the ACM (JACM)*, 63(4):39, 2016.

[Efr09] Klim Efremenko. 3-query locally decodable codes of subexponential length. 2009.

[GHSY12] Parikshit Gopalan, Cheng Huang, Huseyin Simitci, and Sergey Yekhanin. On the locality of codeword symbols. *IEEE Transactions on Information Theory*, 58(11):6925–6934, 2012.

[KMRZS15] Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High rate locally-correctable and locally-testable codes with sub-polynomial query complexity. *arXiv preprint arXiv:1504.05653*, 2015.

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 80–86. ACM, 2000.

[Yek08] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *Journal of the ACM (JACM)*, 55(1):1, 2008.