Coding Theory
Lecture 11

Lecture by Dr. Elette Boyle
Typeset by Steven Karas

2018-01-04
Last edited 18:18:39 2018-01-04

Disclaimer These lecture notes are based on the lecture for the course Coding Theory, taught by Dr. Elette Boyle at IDC Herzliyah in the fall semester of 2017/2018. Sections may be based on the lecture notes written by Dr. Elette Boyle.

Agenda

- LDCs in Private Information Retrieval
- Codes for Universal Hashing
- Fuzzy Vaults
- Finding Defects - Group Testing

1 Review

Last time, we covered Locally Decodable Codes, and showed two examples of such codes: Hadamard codes and Reed-Muller codes.

$$(\underbrace{r}_{\#\text{ of queries}} , \underbrace{\delta}_{\leq \delta \text{ corruptions}} , \underbrace{\varepsilon}_{\leq \varepsilon \text{ decoding error}})\text{-LDC}$$

1.1 Hadamard Codes

Hadamard codes are $(2, \delta, 2\delta)$-LDC.

Local decoding is done with some random vector $\vec{v}$:

$$m_i = \vec{v} \oplus (\vec{v} \oplus \vec{e_i})$$

The analysis is based on the union bound on the error hitting either $\vec{v}$ or $\vec{v} \oplus \vec{e_i}$.

1.2 Reed-Muller Codes

Multivariate Reed-Solomon codes in a nutshell. Parameterized by m, q, ℓ.

m is the dimension (number of variables). q is $\mathbb{F}_q$. ℓ is the maximal degree. Derived from these is code dimension $k = \binom{m=\ell}{\ell}$ and $n = q^m$.

Can achieve $(\ell + 1, \delta, (\ell + 1)\delta)$-LCC.

Basically sample ℓ points along a random line, and interpolate the missing codeword symbol.

2 Private Information Retrieval

Specifically, we'll discuss a primitive called Private Information Retrieval (PIR).

Given a set of servers with an identical set of data $DB \in \{0,1\}^k$, we want to learn the ith bit in the database. We don't want the servers to learn any information about what i is. A basic assumption here is that the servers do not communicate with each other.

A trivial solution is to send the entire database.

Interesting questions here are the number of servers needed, and how much communication is necessary?

1 server As a special case, we cannot achieve PIR without sending the full database without computational assumptions (e.g. factoring is superpolynomial).

2 servers Before the work by Dvir and Gopi[1], the best known was n^{ε}. They presented sub-polynomial communication: $< n^{\varepsilon}$ where $\forall \varepsilon > 0$.

3+ servers Best known solutions use locally decodable codes.

r-query LDC with special smoothness property gives us r-server PIR.

2.1 Formal Definition

Given a database $x \in \{0,1\}^k$, $r \in \mathbb{N}$ non-communicating servers, where a client holds an index $i \in [k]$, and the servers follow instructions but try to learn i.

PIR is a triple of algorithms[1]:

$$(\underbrace{Q}_{\text{query}}, \underbrace{A}_{\text{answer}}, \underbrace{\quad R \quad}_{\text{Reconstruct}})$$

Query The client on input $i \in [k]$ samples entropy source `rand` and evaluates:

$$(q_1, \ldots, q_r) = Q(i; \texttt{rand})$$

For every $j \in [r]$, the client sends q_j to server S_j.

Answer Each server S_j computes and sends the answer back to the client:

$$\text{ans}_j = A(j, \vec{x}, q_j)$$

Reconstruct The client reconstructs the output:

$$x'_i = R(a_1, \ldots, a_r, i, \texttt{rand})$$

Correctness For any $k \in \mathbb{N}$, $\vec{x} \in \{0,1\}^k$, and $i \in [k]$, then with high probability over the choice of initial query generation, it holds that $x'_i = x_i$.

Privacy Each server individually learns no information on i.

For any k, any $j \in [r]$, and any $i \neq i^* \in [k]$, it holds that:

$$\{q_j(i, \texttt{rand})\} \equiv \{q_j(i^*, \texttt{rand})\}$$

[1] multi-round approaches exist, but a single round approach is the best we know of

Smoothness A correct local decoding algorithm Dec for an r-query LDC is smooth if for any k, and any target index $i \in [k]$, then for any query $j \in [r]$ it is individually uniform over $[n]$.

Note that both Hadamard and Reed-Muller decoders are smooth, which arises from them sampling uniformly over the codeword.

In some sense, smoothness is without loss of generality. The intution for this is that if we are too far from uniform, then there are some positions that are very likely to be queried, which causes too high probability for decoding error. Formally, we can show that the existence of an LDC with certain properties implies the existence of a smooth LDC with similar properties.

On the other hand, smoothness implies a form of robustness to errors (to within a factor of r). This follows from the probability that our queries will hit outside the corrupted regions.

2.2 PIR from LDC

Let $C : [q]^k \to [q]^n$ be a smooth $(r, 0, 0)$ LDC[2]. We claim there exists a r-server PIR with $O(r \log_2(nq))$ bits of communication to privately access a database in $[q]^k$.

Proof Sketch Each server encodes $C(\vec{x}) \in [q]^n$. The client runs the local decoding algorithm.

Query

Run Dec on target index i
$l_1, ..., l_r$ = r query locations into $C(\vec{x})$

Answer

Given l_j
$\vec{y}$ = $C(\vec{x})$
Return $\vec{y}_{l_j}$

Reconstruct

Finish executing Dec on $y_{l_1}, ..., y_{l_r}$
Output x_i

Correctness Follows from the correctness of LDC decoding.

Communication complexity Client sends index $l_j \in [n]$ per server. Server responds with $y_{l_j} \in [q]$ each.

$$\begin{aligned} &= r \cdot (\log_2 n + \log_2 q) \\ &= r \log_2 nq \end{aligned}$$

Privacy By smoothness, the query is uniform, and therefore private.

Remark Note that for Hadamard, the communication overhead in the 2-server PIR is $n = 2^k \Rightarrow \log n = k$.

[2] note that we defined $q = 1$ until now for convenience

3 Codes for Universal Hashing

Utilizing ECCs for constructing a Strong Universal Hashing function.

Hash Family Let D, Σ be sets and $m \in \mathbb{N}$. A hash family $\mathcal{H}$ for domain D, range Σ is a set of functions $\{h_1, \ldots, h_m\}$ such that $h_i : D \to \Sigma$.

Note that for $|D| \gg |\Sigma|$, many applications desire it to be computationally hard to find any $x \neq x' \in D$ such that $h_i(x) = h_i(x')$. We will consider a weaker bound.

Almost Universal Hash Family A hash family $\{h_1, \ldots, h_m\}$ defined over some domain D, range Σ is an ε-almost universal hash family for some $0 < \varepsilon \leq 1$ if for any $x \neq y \in D$, for a randomly chosen $i \leftarrow [m]$:

$$\Pr_{i \leftarrow [m]}[h_i(x) = h_i(y)] \leq \varepsilon$$

An lower bound on ε is $\frac{1}{|\Sigma|} \leq \varepsilon$. Such a hash family where $\varepsilon = \frac{1}{|\Sigma|}$ is called *Universal*.

3.1 Application of Universal Hashing

Fast table lookup data structure. Store a set of N items from D where $|D| \gg N$.

We want to do this such that we have low storage. We also want low lookup time: given some $x \in D$, quickly check if $x \in$ set S.

Ordered List Store N items in sorted arrays. $O(N)$ space, with $O(N \log N)$ preprocessing and $O(\log N)$ lookup time.

Universe Array Store array of size D of flags. $O(D)$ space, $O(D)$ preprocessing, and $O(1)$ lookup.

Using Universal Hashing With domain D and some range Σ, create a hash table. Store an array of linked lists (hash table with chaining).

Choose the Σ such that the expected bucket chain lengths will be constant.

Uses $O(N + |\Sigma|)$ space, and $O(N + |\Sigma|)$ preprocessing, with $O(\varepsilon)$ lookup time.

Complexity of Lookup Let $\mathcal{H} = \{h_1, \ldots, h_m\}$ ε-almost universal hash family. Then for any distinct $x, a_1, \ldots, a_N \in D$:

$$\mathop{\mathrm{E}}_{i \leftarrow [m]}[|\{a_j \mid h_i(a_j) = h_i(x)\}|] \leq \varepsilon N$$

A formal proof of this follows by linearity of expectation.

Suppose that we want $|\Sigma| = O(N)$ to not hurt our space or preprocessing time. There exist (left as an exercise to the reader) such families that $\varepsilon = \frac{1}{|\Sigma|} \sim \frac{1}{N}$

4 Fuzzy Vaults

For using biometric data as authentication, where scans are not identical, and to map them to bitstreams (e.g. after rotational/lens correction, uses ECC to map to a canonical bitstream).

5 Defect detection - Group testing

Given a population of size n where a small subset $\ell \ll n$ have a rare disease. When given a test that acts on a subset and detects any presence, identify the subset of the population with a minimum number of tests.

If we consider the test as an almost XOR-like construction, we can construct some parity check matrix that produces a syndrome, especially when we use a low density parity check.

6 Next Time

Next week's lecture was moved to this week. Next lecture will be held on January 18th.

References

[1] Zeev Dvir and Sivakanth Gopi. 2-server PIR with subpolynomial communication. *J. ACM*, 63(4):39:1–39:15, 2016.