Expander Graph and Locally Decodable Source Coding

Kedar Tatwawadi, Yanjun Han

I. Abstract

An expander graph $G = (V, E)$ is a bipartite graph that having the property that every "small" set on the left has a relatively "large" neighbor. This property is desirable because it helps constructing good source and channel codes, and it also has nice locality properties when used to construct locally decodable codes. This report focuses on the application of expander graph in locally decodable source codes.

In the first section, we introduce the existence and basic properties of expander graphs, and how they are applied in channel coding. The second section is devoted to the order-optimal compression of sparse sequences with locality constraint, where the expander graph can be used to achieve a non-adaptive scheme of locally decodable source codes. The last section shows that for general sources with positive entropy, any additive redundancy can be achieved with a bounded locality.

II. Introduction

A. Expander Graph

We inherit notations in graph theory, where V denote the set of vertices and E the set of edges. For a bipartite graph, we use L to denote the set of nodes that are on the left, and similarly R the set of nodes on the right. A d left-regular graph is a bipartite graph where every left node has degree d. The neighbors of a set of node S is denoted as $N(S)$, which are the set of nodes adjacent to S.

Definition 1. *(Expander Graph) [1]: A bipartite graph $G = (L \cup R, E)$ is a $(n, m, D, \gamma, \alpha)$-expander graph if*

1) *G is D left-regular;*
2) *$|L| = n, |R| = m$;*
3) *Any $S \subset L$ with $|S| \leq \gamma n$, $|N(S)| \geq \alpha |S|$.*

The following theorem establishes the existence of such graphs. Moreover, it gives a probabilistic bound on a random construction. Specifically, this theorem states that for a randomly constructed regular bipartite graph, the probability of such graph being a expander graph, with suitable expander parameters, is close to 1 exponentially.

Theorem 1. *Let G be a randomly constructed bipartite graph, with $|L| = n, |R| = m$ which in addition is D left-regular, and the edges are independently added. For any $\alpha \in (0, 1)$, with high probability, the graph G is a $(n, m, D, \gamma, \alpha(\gamma))$ expander graph, where*

$$\alpha(\gamma) = \frac{m}{n}(1 - e^{-D\gamma n/m}) - \sqrt{2\gamma D h_2(\gamma)/\log 2}$$

Proof. Fix any set $S \subset L$ with $|S| = \gamma n$. First, the probability that x is not a neighbor of S is bounded by

$$\mathbb{P}(x \notin N(S)) \leq (1 - \frac{1}{m})^{D|S|}$$

The expected size of its neighbor can then be bounded by

$$\mathbb{E}[|N(S)|] = \sum_{x \in R} \mathbb{P}(x \in N(S)) \geq m(1 - (1 - \frac{1}{m})^{D|S|}) \geq m(1 - e^{-D\gamma n/m})$$

The next step is to show that the size of $N(S)$ concentrated around its mean. To achieve that, we construct a Doob martingale and show it has bounded increment. Then we can apply Azuma's inequality to get the desired concentration property. Let $E(S) = \{E_1, \cdots, E_{D|S|}\}$ be the set of random edges connected to S and let

$$X_i = \mathbb{E}[|N(S)||E_1, \cdots, E_i]$$

be the conditional expectation of the size of $N(S)$ given the first i edges. The following lemma shows X_i is a martingale with bounded difference.

Lemma 1. *$X_i = \mathbb{E}[|N(S)||E_1, \cdots, E_i]$ is a martingale with bounded difference 1.*

Proof. First it's easy to see

$$\mathbb{E}[X_{i+1}|E_1, \cdots, E_i] = \mathbb{E}[\mathbb{E}[|N(S)||E_1, \cdots, E_{i+1}]|E_1, \cdots, E_i] = \mathbb{E}[|N(S)||E_1, \cdots, E_i] = X_i$$

which establishes X_i is a martingale with respect to the filtration $\sigma(\{E_1, \cdots, E_i\})$. Secondly, we show that this martingale has bounded difference. Consider

$$\begin{aligned}
|X_{i+1} - X_i| =& |\mathbb{E}[|N(S)||E_1^i, E_{i+1}] - \mathbb{E}[|N(S)||E_1^i]| \\
\overset{(a)}{=}& |\sum_{e_{i+2}^{D|S|}} \mathbb{P}(E_{i+2}^{D|S|} = e_{i+2}^{D|S|})(\mathbb{E}[|N(S)||E_1^i, E_{i+1}, E_{i+2}^{D|S|} = e_{i+2}^{D|S|}] \\
& - \sum_{e'^{D|S|}_{i+1}} \mathbb{P}(E_{i+1}^{D|S|} = e'^{D|S|}_{i+1})\mathbb{E}[|N(S)||E_1^i, E_{i+1}^{D|S|} = e'^{D|S|}_{i+1}])| \\
\overset{(b)}{=}& |\sum_{e'_{i+1}, e_{i+2}^{D|S|}} \mathbb{P}(E_{i+1} = e'_{i+1})\mathbb{P}(E_{i+2}^{D|S|} = e'^{D|S|}_{i+2}) \\
& * (\mathbb{E}[|N(S)||E_1^i, E_{i+1}, E_{i+2}^{D|S|} = e'^{D|S|}_{i+2}] - E[|N(S)||E_1^i, E_{i+1} = e'_{i+1}, E_{i+2}^{D|S|} = e'^{D|S|}_{i+2}])| \\
\leq& \sum_{e'_{i+1}, e_{i+2}^{D|S|}} \mathbb{P}(E_{i+1} = e'_{i+1})\mathbb{P}(E_{i+2}^{D|S|} = e'^{D|S|}_{i+2}) \\
& * |\mathbb{E}[|N(S)||E_1^i, E_{i+1}, E_{i+2}^{D|S|} = e'^{D|S|}_{i+2}] - E[|N(S)||E_1^i, E_{i+1} = e'_{i+1}, E_{i+2}^{D|S|} = e'^{D|S|}_{i+2}]| \\
\overset{(c)}{\leq}& \sum_{e'_{i+1}, e_{i+2}^{D|S|}} \mathbb{P}(E_{i+1} = e'_{i+1})\mathbb{P}(E_{i+2}^{D|S|} = e'^{D|S|}_{i+2}) = 1
\end{aligned}$$

Where $(a),(b)$ are because of independence, (c) is because that $|N(S)|$ as a function of $\{E_1,..,E_{D|S|}\}$ is edge Lipschitz; when only one edge in $\{E_1,..,E_{D|S|}\}$ changes, $|N(S)|$ will differ by at most 1. In the above proof, for clarity purpose, we suppressed the dependency of $\mathbb{E}[|N(S)||E_1,..,E_{i+1}]$ on $\omega=(e_1,..,e_{D|S|})$. □

Now that we established the lemma, it is straightforward to apply Azuma's inequality.

$$\begin{aligned}&\mathbb{P}(|N(S)|-\mathbb{E}[|N(S)|]>\lambda\sqrt{\gamma Dn})\\ \leq&\mathbb{P}(|N(S)|-\mathbb{E}[|N(S)|]>\lambda\sqrt{D|S|})\leq\exp(-\lambda^2/2)\end{aligned}$$

Apply a union bound, we get

$$\begin{aligned}&\mathbb{P}(|N(S)|-\mathbb{E}[|N(S)|]>\lambda\sqrt{D|S|}\text{ for some }S\text{ with }|S|=\gamma n)\\ \leq&\text{Vol}(n,\gamma n)\exp(-\lambda^2/2)\\ \leq&2^{nh_2(\gamma)}\exp(-\lambda^2/2)\end{aligned}$$

So we can pick $\lambda=\sqrt{(2+\epsilon)nh_2(\gamma)/\log 2}$ for any positive ϵ (say, $\epsilon=0.5$). The final inequality becomes

$$\begin{aligned}&\mathbb{P}(|N(S)|-\mathbb{E}[|N(S)|]>\sqrt{(2+\epsilon)\gamma Dh_2(\gamma)/\log 2}\text{ for some }S\text{ with }|S|=\gamma n)\\ =&\mathbb{P}(|N(S)|-\mathbb{E}[|N(S)|]>\sqrt{(2+\epsilon)nh_2(\gamma)/\log 2}\sqrt{\gamma Dn}\text{ for some }S\text{ with }|S|=\gamma n)\\ \leq&2^{-\epsilon h_2(\gamma)n}\to 0\end{aligned}$$

□

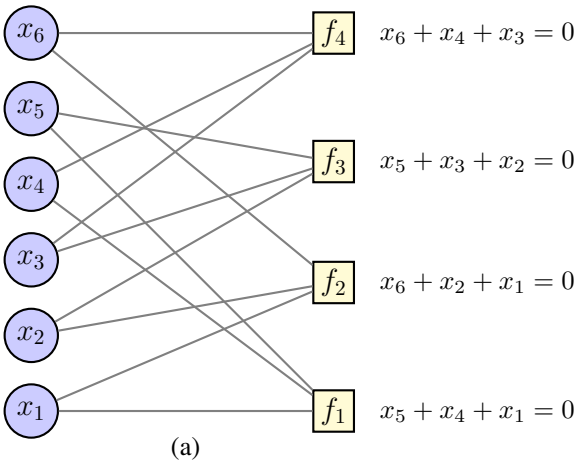
The above theorem gives a probabilistic argument on the existence of such expander graphs. Explicit constructions are also given recently (2002), which will be stated in the next theorem.

Theorem 2. *[4] Explicit construction of a $(n,m,D,\gamma,D(1-\epsilon))$ expander graph exists for any $\epsilon\in(0,1)$.*

Combining theorem 1 and 2, we can say that expander graphs are objects that are within our reach; not only do we know that they exists with high probability, but also can we construct some of them explicitly. The next subsection will then switch to an application of expander graph in coding theory, which also serves as a motivation for our application in source codes with locality.

B. Motivation: an Application in Channel coding

The parity check matrix of a linear code is often represented as a bipartite graph. The left nodes are the bits in the codeword, and nodes on the right represent the constraints. Below is a graph showing a codeword with $n=6$ and 4 parity check bits. The rate for the code is 0.6. Note the graph is also a $(6,4,2,\frac{1}{3},\frac{3}{2})$ expander.



(a)

An important notion is the unique neighbor, which is formally defined as follows:

Definition 2. *For any $S \subset L$, the unique neighbor $U(S)$ is the set of nodes that are adjacent to exactly one node in S.*

Suppose for a graph $G = (L \cup R, E)$ representing a linear code $\mathcal{C}$, if any set $S \subset L$ with $|S| < \gamma n$ has non-empty unique neighbors, then we can immediately conclude that:

$$\mathcal{C} \text{ has relative distance at least } \gamma$$

The reason for the above claim is, suppose c is the codeword with the minimum weight, and let S be the set of indices of 1's in c. If $U(S) \neq \emptyset$, then a constraint in $U(S)$ is adjacent to exactly one non-zero bit, which violates the constraint which says the bits it is adjacent to should sum up to 0.

Theorem 3. *(unique neighbor property of expander graphs) [1] Let G be a $(n, m, D, \gamma, D(1-\epsilon))$ expander graph. If $\epsilon < 0.5$ then for any $S \subset L$ with $|S| < \gamma n$, $U(S)$ is non-empty.*

Proof. Count number of the edges, e, between S and $N(S)$. First start with S. Because G is D left-regular, $e = D|S|$. Second, start with $N(S)$. Due to the expander property, $|N(S)| \geq D(1-\epsilon)|S|$. For nodes in $N(S)$, if it is a unique neighbor, the there is one edge between this node and S. Otherwise, there are at least two edges. Hence we have $e \geq |U(S)| + 2(D(1-\epsilon)|S| - |U(S)|)$. Compare and rearrange terms, we get

$$|U(S)| \geq D(1-2\epsilon)|S| > 0.$$

□

As a result of the above theorem, if we have a "good" expander graph, the code corresponding to it has "good" relative distance (γ) and "good" rate ($1 - \frac{m}{n}$). This helps us to find good linear codes. This procedure can be generalized to construct Tanner codes, which maps a small code into a larger one using expander graphs. In the next few sections, we will go beyond the intuitions discussed here and use expander graphs to construct source codes with local decodability, both in the sparse case and constant rate case.

III. Compression of Sparse Sequences using Expander Graph

In this section we consider the compression problem of a sparse sequence under locality constraints: given a binary sequence $x^n = (x_1, \cdots, x_n)$ with Hamming weight at most s, we would like to compress this sequence losslessly into $c(x^n)$ such that it only requires probing t bits of the compressed sequence $c(x^n)$ to decode each bit of the original sequence x^n. Formally, we have the following definition for a locally decodable (n, s, R, t) code.

Definition 3 (Locally Decodable (n, s, R, t) Code)**.** *A (fixed-length) (n, s, R, t) locally decodable code consists of a mapping*

$$c : \binom{[n]}{\leq s} \to \{0,1\}^{nR}$$

and subsets $S_1, \cdots, S_n$, mappings $f_1, \cdots, f_n$ which may depend on R but not on x^n such that $|S_i| \leq t, x_i = f_i(c_{S_i}(x^n))$ for any $i \in [n]$ and $x^n \in \binom{[n]}{\leq s}$. Here $\binom{[n]}{\leq s}$ denotes the set of all possible binary sequences of length n and Hamming weight at most s, and $z_S \triangleq (z_i)_{i \in S}$ denotes the subsequence indexed by S for any sequence z.

We note that there can also be some variable-length variant of the above definition by imposing some probability distribution on $\binom{[n]}{\leq s}$, and we will discuss this variant later.

The locally decodable sourcing coding problem has various applications in practice and we list two of them. The first example is the *static membership problem*: given a subset S of at most s keys, store it so that queries of the form "Is x in S?" can be answered quickly. This is a fundamental data structure problem with a long history. Yao [8] showed that if the data structure consists of a table with s cells where the keys are stored explicitly and the univese from which the set S is chosen is large enough, then the sorted table with binary search is optimal. In order to study data structures where elements of the set S are not stored explicitly, [8] proposed the cell probe model. In this model, the set S is stored as a table of cells, each capable of holding one element of the universe; that is, if the universe has size $|S|$ being a power of two, then each cell holds $\log_2 |S|$ bits. The queries are to be answered by probing the table adaptively; that is, each probe can depend on the results of earlier probes and the query element x. The goal is to process membership queries with as few probes as possible, and at the same time keep the size of the table small.

The second application is in bioinformatics [9], in which a DNA sequence is stored as a binary string with relation to a reference sequence, with 1s representing single nucleotide polymorphisms (SNPs) at those positions. In SNP calling, we are interested in learning whether there is a SNP at position i. Since we are not interested in any other information about the sequence, we would ideally like to accomplish this by accessing few bits in the compressed representation of the DNA sequence. In this specific instance, local decodability is strongly motivated, since decompressing the whole genome can be prohibitively expensive from a memory standpoint.

The most general problem in studying the locally decodable source code can be determining the entire region of all achievable (n, s, R, t). For the sake of simplicity and brevity, here we only consider the case where the sequence is sparse, i.e., $s = \delta n$ with $\delta = o_n(1)$, and the rate R is *competitive* to the optimal compression rate without the

locality constraint. Specifically, by a simple counting argument we know that without the locality constraint, the optimal rate is given by

$$\frac{1}{n}\log_2 \sum_{i=0}^{n\delta} \binom{n}{i} = \delta \log_2(\frac{1}{\delta}) \cdot (1 + o_n(1))$$

and thus we require that $R = O(\delta \log_2(\frac{1}{\delta}))$. Now our question becomes: what is the minimum locality t we can achieve given $s = \delta n$ and $R = O(\delta \log_2(\frac{1}{\delta}))$? Despite the great simplification, we remark that many key ideas will be reflected in studying this simple problem, and the correct answer here is $t = \Theta(\log(\frac{1}{\delta}))$ [10]:

Theorem 4. *There exists a (fixed-length) locally decodable $(n, \delta n, O(\delta \log_2(\frac{1}{\delta})), t)$ code iff $t = \Omega(\log(\frac{1}{\delta}))$.*

The variable-length variant was also proved in [11] with the same answer. We also remark that there exists a non-adaptive scheme (i.e., choose the positions of all probed bits in advance) which achieves this bound, and this bound is also non-beatable even with adaptive approaches (i.e., one is allowed to choose the positions of probed bits based on all available history). Next we will prove Theorem 4 by showing the achievability and the converse parts separately.

A. Achievability via Expander Graph

We adopt a graph-theoretic viewpoint of this problem: consider a bipartite graph $G = (V_L \cup V_R, E)$ with $|V_L| = n$ representing source bits and $|V_R| = nR$ representing compressed bits, then viewing the pair of source bit and probed bit as an edge, a non-adaptive locally decodable (n, s, nR, t) code corresponds to such a bipartite graph with all left-degrees at most t. Given a realization of the original sequence and compressed sequence, we call a node in V_L as 1-code if the source bit corresponding to this node is 1 in this realization, and 0-node otherwise. Now we introduce the definition of (p, q)-colorable graph:

Definition 4 ((p, q)-colorable graph)**.** *The bipartite graph $G = (V_L \cup V_R, E)$ is called (p, q)-colorable iff we can color V_R in two colors $\{0, 1\}$ such that each 1-node in V_L is connected to at least p nodes in V_R colored in 1, and each 0-node is connected to at least q nodes colored in 0.*

Assuming the existence of a (p, q)-colorable graph, we can use the following simple non-adaptive encoding and decoding rules:

- Fix some suitable bipartite graph first and reveal it to both the encoder and decoder;
- In the encoding process, for each source realization we simply find a (p, q)-coloring and compress into a binary sequence consisting of the colors of the right nodes V_R;
- In the decoding process for each source bit, probe the colors of the nodes connected to it. If there are at least p nodes colored as one, decode the source bit as one; if there are at least q nodes colored as zero, decode it as zero; if neither of the above happens, decode arbitrarily.

It is easy to see that the decoding rule always returns the correct answer if $p + q > t$, and we arrive at the following reduction of the achievability part:

Lemma 2. *There exists a non-adaptive locally decodable (n, s, R, t) code if for any subset S of 1-nodes in V_L with $|S| \leq s$, there exists a $(\lceil \frac{t+1}{2} \rceil, \lceil \frac{t+1}{2} \rceil)$-coloring.*

Based on the previous lemma, it suffices to construct a bipartite graph G such that this graph is $(\lceil \frac{t+1}{2} \rceil, \lceil \frac{t+1}{2} \rceil)$-colorable for each legal source realization. The main result of this subsection is that a suitable expander graph satisfies this property. By Theorem 1, there exists a $(n, O(n\delta \log_2(\frac{1}{\delta})), t = O(\log_2(\frac{1}{\delta})), 2\delta, (1-\epsilon)t)$ expander graph G with some $\epsilon \in (0, \frac{1}{4})$. Now for each subset of 1-codes in V_L with cardinality at most δn, we consider the following coloring algorithm:

1) Initially, all nodes in V_R are not colored;
2) If some node in V_R is only connected to 1-nodes, color it as 1; if some node in V_R is only connected to 0-nodes, color it as 0; use an arbitrary order if there are multiple choices;
3) If some node in V_L have more than half of its neighbors colored, remove this node and all associated edges from the graph; use an arbitrary order if there are multiple choices;
4) Repeat the previous two steps until convergence.

This algorithm has the following performance guarantee:

Lemma 3. *For each subset of 1-codes in V_L with cardinality at most δn, the previous coloring algorithm yields a $(\lceil \frac{t+1}{2} \rceil, \lceil \frac{t+1}{2} \rceil)$-coloring.*

Proof. It is easy to see that the (p, q)-colorable condition is not violated during the process of this algorithm, so it suffices to prove that this algorithm can make progress (color one node in V_R or remove one node in V_L) as long as the resulting graph is non-empty. Assuming the contrary, and let S_1, S_0 be the set of 1-nodes and 0-nodes in V_L, respectively, and T be the remaining uncolored node in V_R. If $|S_1| \leq |S_0|$, we arbitrarily pick a subset $S_0' \subset S_0$ of size $|S_1|$ and consider $S = |S_1 \cup S_0'|$. For any $x \in S_0'$, since x cannot be removed, we have

$$|N(x) \cap T^c| \leq \frac{t}{2}.$$

Summing over $x \in S_0'$ yields

$$|N(S_0') \cap T^c| \leq \sum_{x \in S_0'} |N(x) \cap T^c| \leq \frac{t|S_1|}{2}.$$

Now using $N(S) = N(S_1) \cup (N(S_0') \backslash T)$, we have

$$|N(S)| \leq N(S_1) + |N(S_0') \cap T^c| \leq t|S_1| + \frac{t|S_1|}{2} = \frac{3t}{2}|S_1|.$$

However, on the other hand we have $|S| = 2|S_1| \leq 2\delta n$, by the property of expander graph we know that

$$|N(S)| \geq t(1-\epsilon)|S| = 2(1-\epsilon)t|S_1|.$$

By our choice, $\epsilon < \frac{1}{4}$, then comparing the previous two inequalities yields the desired contradiction. The case where $|S_0| \leq |S_1|$ can also be dealt with analogously. □

Combining all previous parts concludes the achievability of Theorem 4.

B. Converse via LYM Inequality

Next we show that even for an adaptive scheme, the locality $t = o(\log_2(\frac{1}{\delta}))$ cannot be achieved. The proof here is combinatorial: for any source coding scheme, define the set

$$A(x^n) = \{(i, c_i(x^n)) : \text{position } i \in [Rn] \text{ is probed to decode some bit } j \in [n] \text{ where } x_j = 1\}.$$

We restrict our attention to all source realizations x^n with Hamming weight exactly δn, and have the following observations:

- $A(x^n) \subset [Rn] \times \{0, 1\} \equiv T$, where $|T| = 2Rn$;
- $|A(x^n)| \leq t \cdot \delta n$: obvious from definition since there are $n\delta$ ones in the source sequence and decoding each of them only requires probing t bits;
- $A(x^{1,n}) \nsubseteq A(x^{2,n})$ for $x^{1,n} \neq x^{2,n}$: consider decoding the position where $x^{1,n}$ is one and $x^{2,n}$ is zero. Then suppose that the true source is $x^{2,n}$ and $A(x^{1,n}) \subset A(x^{2,n})$, the decoder will observe (part of) $A(x^{1,n})$ which is a subset of $A(x^{2,n})$, and it will incorrectly decode into one, a contradiction.

These properties suffice to give a lower bound on the locality t. In fact, if we simply use that $A(x^{1,n}) \neq A(x^{2,n})$ which is weaker than the third point, we immediately have

$$\binom{n}{\delta n} = |\{A(x^n) : |x^n|_0 = \delta n\}| \leq \left|\binom{[|T|]}{\leq t \cdot \delta n}\right| = \sum_{i=0}^{t\cdot\delta n} \binom{2Rn}{i} \leq 2^{t\delta n}\binom{2Rn}{t\delta n}$$

and plugging in $R = O(\delta \log_2(\frac{1}{\delta}))$ gives $t = \Omega(\log_2(\frac{1}{\delta}))$. However, fully exploiting the third observation will yield to a tighter bound, which is crucial in the general (n, s, R, t) case and the variable-length source coding case [11]. We begin with the LYM inequality characterizing the property of the family of sets which are not subsets of each other.

Theorem 5 (LYM inequality [12])**.** *Suppose $A_1, \cdots, A_m \subset T$ are not subsets of each other, then*

$$\sum_{k=1}^{m} \binom{|T|}{|A_k|}^{-1} \leq 1.$$

Proof. For each $k \in [m]$, we map A_k to a subset of permutations of T: this subset contains all permutations of T in which the first $|A_k|$ elements are just a permutation of A_k. Now it is straightforward to verify that our assumption implies that different A_k cannot be mapped to a common permutation. Since each A_k generates $|A_k|!(|T| - |A_k|)!$ permutations, we conclude that

$$\sum_{k=1}^{m} |A_k|!(|T| - |A_k|)! \leq |T|!.$$

A rearrangement of the previous inequality gives the LYM inequality. □

The following corollary is immediate.

Corollary 1 (Sperner's Theorem)**.** *Suppose $A_1, \cdots, A_m \subset T$ are not subsets of each other and $|A_k| \leq r \leq \frac{|T|}{2}$ for any $k \in [m]$, then*

$$m \leq \binom{|T|}{r}.$$

Proof. Just invoke Theorem 5 to conclude that

$$m\binom{|T|}{r}^{-1} = \sum_{k=1}^{m}\binom{|T|}{r}^{-1} \leq \sum_{k=1}^{m}\binom{|T|}{|A_k|}^{-1} \leq 1.$$

□

Now applying the Corollary to our case, we either have $t \cdot \delta n \geq \frac{|T|}{2} = Rn$, or have

$$\binom{n}{\delta n} = |\{A(x^n) : |x^n|_0 = \delta n\}| \leq \binom{|T|}{t \cdot \delta n} = \binom{2Rn}{t\delta n}.$$

Plugging into $R = O(\delta \log_2(\frac{1}{\delta}))$, both cases give $t = \Omega(\log_2(\frac{1}{\delta}))$, as desired.

C. Further Discussions

Theorem 4 is only devoted to the sparse competitive case where $s = n\delta$ and $R = O(\delta \log_2(\frac{1}{\delta}))$, while there are also some other interesting scenarios. For example, people may favor a small locality at the expense of an increasing rate or some error probability in practice. The achievability part based on expander graph still sheds light on the general case: for example, $(n, \delta n, O(\delta \log_2(\frac{1}{\delta})), 1)$ is possible if we allow some error probability [10]. In fact, after our construction of (p, q)-colorable graph, instead of probing all neighbors and use a "majority vote" rule, we can also randomly probe one neighbor and declare its color. As $\epsilon \to 0$ in the construction of expander graph, we can let $p, q \to t$ which yields to a vanishing error probability. Moreover, if we are restricted to use a deterministic rule, we can also use an expander graph with a small left-degree (say, 4). However, Theorem 1 is an asymptotic result and cannot be used in this case, and one should verify carefully whether the desired expander graph exists or not (our construction involves an $O(\cdot)$ notation). Fortunately, this is still doable and it was shown that all $t = 2, 3, 4$ can yield some non-trivial compression rate [13], where it was also shown that the (p, q)-colorable graph is intrinsically related to its expander property. For example, an expansion factor of r is sufficient and necessary if $t = 2r - 1$ and we seek an (r, r)-coloring, and an expansion factor of $\frac{8}{3}$ is sufficient and necessary for $t = 4$ and a $(3, 2)$-coloring [13]. The lower bound argument can also be applied to these general cases. For an intensive overview of achievability and converse results in the sparse compression case, we refer to [14].

Another interesting case is the non-sparse case where the Hamming weight $s = \delta n$ may be linear in n, where the major difference is that here the optimal compression rate $h_2(\delta)$ without the locality constraint is bounded away from zero and can be quite significant. As a result, a rate $C \cdot h_2(\delta)$ can be significantly worse than $h_2(\delta)$ and cannot be called "competitive", and we look for some small-locality code with rate $h_2(\delta) + \epsilon$. However, this problem with an additive redundancy is considerably harder than the previous one with a multiplicative redundancy, as will be discussed in the following section.

IV. LDSC CODES FOR KNOWN I.I.D SOURCES

The problem of Locally decodable source codes for i.i.d sources has been studied briefly in the literature. [3] presents a very good argument for the case of rational sources, which is based on LDGM codes. We show a different proof than that of [3] for the known iid source case, and extend these arguments for any general stationary & ergodic source.

Let $x^n = x_1, \ldots, x_n$ be a n length sequence generated by a binary i.i.d process X with distribution $B(p, 1-p)$ and entropy $H(X) = h_2(p)$. Our problem is to achieve a rate $R < H(X) + \delta$, while simultaneously having the local decompressibility to be finite as $n \to \infty$. The compression scheme for the sequence x^n is explained below.

A. Encoding Scheme

Given the following constraint on the rate: $R < H(X) + \delta$, the compression scheme is:

1) The sequence x^n is processed block by block, with each block of size B.
2) There are two output streams $\mathcal{Y}$ and $\mathcal{U}$:
   a) Let $\epsilon < \delta$. Then, for the input sequence of length n, the stream $\mathcal{Y}$ has an associated sequence y^L, of length $L_n = n(H(X) + \epsilon)$. Let $L_B = B(H(X) + \epsilon)$
   b) Also, the stream $\mathcal{U}$ has an associated sequence u^n, of length n
3) Each block $B_i = x_{iB}^{(i+1)B-1}$ corresponds to the output blocks $Y_i = y_{iL_B}^{(i+1)L_B-1}$ and $U_i = u_{iB}^{(i+1)B-1}$ in the $\mathcal{Y}$ and $\mathcal{U}$ streams.
4) We define the typical set $\mathcal{T}_\delta(V, n)$ for a i.i.d process V as :

$$\mathcal{T}_\epsilon(V, n) = \{v^n : 2^{-n[H(V)+\epsilon]} \leq P(v^n) \leq 2^{-n[H(V)-\epsilon]}\} \tag{1}$$

   As, $|\mathcal{T}_\epsilon(V, n)| \leq 2^{n[H(V)+\epsilon]}$, thus every sequence $v^n \in \mathcal{T}_\epsilon(V, n)$ can be assigned an index $I(v^n)$ of size $n[H(V) + \epsilon]$. Also, for a sufficiently large n, $(n > N(\epsilon))$, $P(x^n) \in \mathcal{T}_\epsilon(V, n) \geq 1 - \epsilon$. Now that we have defined these quantities, we proceed to describe the block-by-block encoding procedure:
   a) If $B_i \in \mathcal{T}_\epsilon(X, B)$, then: $Y_i = I(B_i)$ and $U_i = 000\ldots0$.
   b) If $B_i \notin \mathcal{T}_\epsilon(X, B)$, then: $Y_i = 000\ldots0$ and $U_i = B_i$.
5) The y^L sequence is encoded as it is (identity map).
6) The u^N sequence encoding is as follows:
   a) If the sparsity of the u^n is less than $\delta > \epsilon_1 > \epsilon$, then the sequence u^N is encoded using the expander graph based construction of Theorem[4]
   b) If not we encode as it is (i.e. without any compression)
   c) To differentiate between these cases, we also store a bit S_b. $S_b = 1$ if the ϵ sparsity condition is met.

   Let the length of the encoded stream be L_U.

B. Local Decoding Scheme

We discuss the local decoding scheme below, to access the symbol x_r.

1) Let $bid = \lfloor r/B \rfloor$, be the index of the block corresponding to x_r
2) We read the block Y_{bid} from the $\mathcal{Y}$ stream, by accessing L_B bits. If $Y_{bid} \neq 000\ldots0$, then we output the $r \mod (B)$ bit of the block Y_{bid}, and skip the next step.
3) If $Y_{bid} = 000\ldots0$, then:
   a) If the sparsity identifier bit S_b is 1, then using the expander graph based retrieval method, we read the symbol u_r from the $\mathcal{U}$ stream. This takes $\mathcal{O}\left(log\left(\frac{1}{\epsilon_1}\right)\right)$ bits of access. We output this symbol u_r.

b) If $S_b = 0$, then we directly output u_r.

C. Performance Analysis

1) Compression performance: We first analyze the compression performance of the scheme. First consider the probability of sparsity of the u^n being more than ϵ. Using the Hoeffding's inequality, this is upper bounded by (for a $B > N(\epsilon)$):

$$P_{\epsilon_1} = P(wt(u^n) > \epsilon_1 n) \leq e^{-2(\epsilon_1 - \epsilon)^2 H(X) \frac{n}{B})} \tag{2}$$

As our scheme is block-by-block, the compression rate R, can be found as:

$$\begin{aligned} R &= (L_B + L_U)/B \\ &= H(X) + \epsilon + (1 - P_{\epsilon_1})\mathcal{O}\left(\epsilon_1 log\left(\frac{1}{\epsilon_1}\right) + P_{\epsilon_1}\right) \end{aligned}$$

As, $P_{\epsilon_1} \to 0$ as $n \to \infty$, thus choosing B, ϵ, ϵ_1 appropriately ($B = \mathcal{O}\left(\frac{1}{\delta^2}\right), \epsilon = \delta/2, \epsilon_1 = 0.75\delta$), we can obtain $R < H(X) + \delta$, as $n \to \infty$. Note that this is a perfectly lossless encoding, where all the randomness is absorbed in the bound on the rate.

2) Random Access Performance: We analyze the random access performance of the scheme. Based on the encoding section, we understand that, the random access t has a worst case bound of:

$$t \leq B(H(X) + \epsilon) + 1 + \mathcal{O}\left(log\left(\frac{1}{\epsilon_1}\right)\right) \tag{3}$$

Based on the parameters chosen to satisfy the constraints on the rate, we get the effective local decodability to be $\mathcal{O}\left(\frac{1}{\delta^2}\right)$. Note that, although we used typical sequences for block-wise encoding of the source, any other fixed block-length source code would have resulted in the same performance. This can be seen from the Theorem 12 from [[16]]:

Theorem 6. *Let $R(n, d, \epsilon)$ be defined as the rate distortion function for fixed-blocklength schemes for length n, distortion d, and the probability of distortion being above d to be ϵ. Then:*

$$R(n, 0, \epsilon) = H(X) + \frac{C}{\sqrt{n}} Q^{-1}(\epsilon) \tag{4}$$

Thus, for a rate bound $R < H(X) + \delta$, any fixed block length coding can achieve local decodability at the best: $\mathcal{O}\left(\frac{1}{\delta^2}\right)$. Finally note that, in practice we rarely know the exact distribution of the i.i.d source. Thus, we next discuss a slight variation of the problem, where we assume that the process is unknown to the encoder or the decoder.

V. Random Access for unknown I.I.D sources

In practice, we rarely know the exact distribution of the process, and is learned by the encoder and communicated to the decoder via the codebook. For an i.i.d binary process, the counts n_1, n_0 of the number of 1's and 0's serve as the sufficient statistics, and are thus communicated by the encoder by storing them using $\mathcal{O}(\log_2 n)$ bits. However, although this does not affect the rate of compression, this affects the local decodability. Every time we need to read a bit from the compressed source, we require first to read these $\mathcal{O}(\log_2 n)$ bits to form the codebook. This defeats

the aim of the problem, which is to have constant time locality as $n \to \infty$. One solution we consider is quantize the learned probability distribution, based on the allowed margin δ. If we wish to obtain constant random access performance, we cannot have number of quantized distributions an increasing function of n, and must only be a function $g(\delta)$, for the rate constraint $R < H(X) + \delta$

We first define the quantized set of distributions: $\mathcal{Q}_k(\epsilon)$ as the set of distributions $Q(\epsilon) = \{q_1, \ldots, q_{|Q|}\}$ such that $\{\forall P \in \mathcal{M}_k, \exists q_i \in Q(\epsilon), D(P||q_i) \leq \epsilon\}$

We show the following lemma:

Lemma 4. *Let* $Q_0^{opt}(\epsilon) = argmin_{Q(\epsilon)\in\mathcal{Q}_0(\epsilon)}|\mathcal{Q}(\epsilon)|$. *Then:*

$$|Q_0^{opt}(\epsilon)| \leq \frac{2}{\epsilon} \tag{5}$$

a) Proof: Consider the quantization levels: $q = \frac{\lfloor pn \rfloor + 0.5}{n}$.

Also, we define $n_0 = \lfloor pn \rfloor, n_0 = n - n_0, \alpha = pn - n_0$.

$$\begin{aligned}
D(p||q) &= p(x)\log\left(\frac{p(x)}{q(x)}\right) + p'(x)\log\left(\frac{p'(x)}{q'(x)}\right) \\
&= \frac{1}{n}\left((n_0+\alpha)\log\left(\frac{n_0+\alpha}{n_0+0.5}\right) + (n_1-\alpha)\log\left(\frac{n_1-\alpha}{n_1-0.5}\right)\right) \\
&\leq \frac{\log_2 e}{n}\left((n_0+\alpha)\left(\frac{\alpha-0.5}{n_0+0.5}\right) + (n_1-\alpha)\left(\frac{0.5-\alpha}{n_1-0.5}\right)\right) \\
&= \frac{(\log_2 e)(\alpha-0.5)^2}{(n_0+0.5)(n_1-0.5)} \\
&\leq \frac{(4\log_2 e)(\alpha-0.5)^2}{n} \\
&\leq \frac{\log_2 e}{n} \\
&< \frac{2}{n}
\end{aligned}$$

Thus, for the binary case, $|\mathcal{Q}_0^{opt}(\epsilon)| \leq \left(\frac{2}{\epsilon}\right)$. Hence proved.

As $D(p||q)$ corresponds to the loss incurred by compressing a source with distribution $p(X)$ by a quantized source with distribution $q(x)$, thus by choosing the quantization level appropriately ($\epsilon < \delta$), we can achieve constant time random access as well as $R < H(X) + \delta$. The encoding will be almost the same as the known i.i.d case, however, the only difference is we also store the index to the quantized distribution using $\log_2 |\mathcal{Q}_0^{opt}(\epsilon)| \leq \mathcal{O}\left(\log_2\left(\frac{1}{\epsilon}\right)\right)$ bits. Thus, the effective local decodability remains unchanged even for the unknown i.i.d process scenario. We next take a look at locally decodable source coding for any generic unknown stationary & ergodic sources.

VI. Universal Random Access

We now take a look at the scenario of compression of any unknown generic stationary ergodic source X, with the entropy rate $\mathbb{H}(X)$. Our aim is to have a scheme which achieves the rate close to the entropy rate of the process, $R < \mathbb{H}(X) + \delta$, while simultaneously having the random access bounded, as $n \to n$, i.e. $t < f(\delta)$. We first start by showing the existence of such a scheme.

A. Existence of LDSC

Our first step is to extend the Lemma[4] to general $k-Markov$ processes. The proof is very similar, and we skip it here.

Lemma 5. $\min_{Q(\epsilon)\in\mathcal{Q}_k(\epsilon)}|Q(\epsilon)| = \mathcal{O}\left(\frac{1}{\epsilon^k}\right)$.

We also define $K_\gamma(X)$ as:

$$K_\gamma(X) = \min_k H(X_0|X_{-k}^{-1}) < \mathbb{H}(X) + \gamma \tag{6}$$

There always exists a finite $K_\gamma(X)$ for any stationary process by definition of the the entropy rate $\mathbb{H}(X)$. In a sense, $k = K_\gamma(X)$ represents the level of markovity required to approximate the process by a $k-Markov$ process, so that after compression we are in the worst γ away from the entropy rate. Thus, we can consider a $K_\gamma(X)-Markov$ approximation to the process X, which can be quantized using $\log_2|\mathcal{Q}_k^{opt}(\epsilon)| \leq \mathcal{O}\left(K_\gamma(X)\log_2\left(\frac{1}{\epsilon}\right)\right)$ bits. As typical sequences are defined for any stationary ergodic process, thus the same encoding scheme described for i.i.d processes can be used here to achieve the $R < \mathbb{H}(X)+\delta$, and the finite locality constraint, for appropriately chosen parameters ϵ, γ. Hence, this shows the existence of the LDSC scheme.

Note that, although the existence is shown, this by no means is a practical scheme as learning the process distribution, or even the parameter $K_\gamma(X)$ is difficult. We next present a methodology of a practical universal compressor with local decodability.

VII. Practical Universal LDSC

Universal schemes like LZ78 [15], can achieve the entropy rate $\mathbb{H}(X)$, for any stationary ergodic process X, however to retrieve any small part of the data, most of the time the entire previous substring needs to be extracted. We present a method of modifying any general universal compression scheme into a locally decodable version.

The problem setting remains the same: we are given a length n subsequence x^n generate by a stationary ergodic process X over binary alphabet. We wish to achieve rate $R < \bar{L}(x^n) + \delta$, for any individual sequence x^n, and also have finite locality as $n \to \infty$. Here, $\bar{L}(x^n)$ is the performance of the universal compressor.We describe the encoding procedure next. For clarity, we only describe the methodology for LZ78.

A. LZ78 LDSC Encoding

For LZ78, we first state the following result:

Theorem 7. *Let $\bar{L}$ be the average length of compression for the sequence x^n by LZ78. (i.e. $1/n$ times the length of the encoding $LZ78(x^n)$). Then:*

$$\bar{L} \leq H_k(x^n) + \frac{Ck}{f(n)} \tag{7}$$

where: $f(n) = \frac{\log\log n}{\log n}$, and C is some constant. here, $H_k(x^n)$ denotes the k^{th} order empirical entropy of the sequence x^n.

Also, let $g(n)$ be some function such that: $\lim_{n\to\infty}\frac{g(n)}{f(n)} = 0$ and $\lim_{n\to\infty} g(n) = \infty$.

Given the following constraint on the rate: $R < \bar{L} + \delta$, the compression scheme is:

1) The sequence x^n is processed block by block, with each block of size B.
2) There are two output streams $\mathcal{Y}$ and $\mathcal{U}$:
   a) For the input sequence of length n, the stream $\mathcal{Y}$ has an associated sequence y^L, of length L. Let $L_B = LN/B$
   b) Also, the stream $\mathcal{U}$ has an associated sequence u^n, of length n
3) Each block $B_i = x_{iB}^{(i+1)B-1}$ corresponds to the output blocks $Y_i = y_{iL_B}^{(i+1)L_B-1}$ and $U_i = u_{iB}^{(i+1)B-1}$ in the $\mathcal{Y}$ and $\mathcal{U}$ streams.
4) If $LZ78(B_i)$ corresponds to the LZ78 encoding of the block B_i. Then, the first L_B bits of $LZ78(B_i)$ are stored in the block Y_i, and the remaining are stored in the block U_i. Zeros are appended to the encoding, in case the length of the encoding is smaller than $L_B + B$, and the blocks Y_i and U_i are filled accordingly.
5) The y^L sequence is encoded as it is (identity map).
6) The u^N sequence encoding is as follows:
   a) If the sparsity of the u^n is less than $\delta > \epsilon_1 > \epsilon$, then the sequence u^N is encoded using the expander graph based construction of Theorem[4].
   b) If not we encode as it is (i.e. without any compression)
   c) To differentiate between these cases, we also store a bit S_b. $S_b = 1$ if the ϵ sparsity condition is met.

   Let the length of the encoded stream be L_U.

The encoding scheme is very similar to the iid scenario and is block based. Note that, we have not yet described yet, how to choose the block length B and L_B. For any given sequence x^n, B is chosen as follows:

Let k_ϵ be defined as:

$$k_\epsilon = \min \left[g(n), \arg\min_k \left[H_k(x^n) < \bar{L} + \frac{\epsilon}{2} \right] \right] \tag{8}$$

For such a $k\epsilon$, we choose the block length B, and L_B are chosen such that:

$$\frac{Ck_\epsilon}{f(B)} < \frac{\epsilon}{2}$$

$$L_B = \bar{L} + \epsilon$$

The intuition here is that, as $\lim_{k\to\infty} \lim_{n\to\infty} H_k(x^n) = \mathbb{H}(X)$, thus k_ϵ is bounded, which makes B finite. It can then be shown that for a sufficiently large n, the rate constraint $R < \bar{L} + \epsilon$ is satisfied with a finite locality.

Note that, this is a more practical LDSC as all the decisions are only based on the individual sequence x^n.

VIII. CONCLUSION & FURTHER WORK

During the course of the project, we tried to understand the problem of Locally Decodable Source Codes. The problem of sparse sequence encoding has been studied well in the theoretical CS literature, which we took a look at in the report. We also presented some ideas which lead to schemes optimal rate with constant locality for general stationary and ergodic sources.

As a part of further work, it would be interesting to look at achieving better locality (and not just finite locality). Another interesting problem would be to look at the best rates achievable for a given locality. We plan to continue working on these interesting problems.

ACKNOWLEDGEMENT

We would like to thank Prof. Mary Wootters for helpful discussions. We would also like to thank our labmate Shirin Saeedi Bidokhti for helpful discussions.

REFERENCES

[1] Guruswami, Venkatesan. Notes 8: Expander Codes and their decoding. Introduction to Coding Theory, CMU courses, 2010.

[2] Sipser, Michael, and Daniel A. Spielman. “Expander codes.” IEEE Transactions on Information Theory 42.6 (1996): 1710-1722.

[3] Mazumdar, Arya, Venkat Chandar, and Gregory W. Wornell. “Local recovery in data compression for general sources.” 2015 IEEE International Symposium on Information Theory (ISIT). IEEE, 2015.

[4] Capalbo, Michael, et al. “Randomness conductors and constant-degree lossless expanders.” Proceedings of the thiry-fourth annual ACM symposium on Theory of computing. ACM, 2002.

[5] Makhdoumi, Ali, et al. “On locally decodable source coding.” 2015 IEEE International Conference on Communications (ICC). IEEE, 2015.

[6] Pananjady, Ashwin, and Thomas A. Courtade. “Compressing sparse sequences under local decodability constraints.” 2015 IEEE International Symposium on Information Theory (ISIT). IEEE, 2015.

[7] Chandar, Venkat Bala. Sparse graph codes for compression, sensing, and secrecy. Diss. Massachusetts Institute of Technology, 2010.

[8] Yao, Andrew Chi-Chih. “Should tables be sorted?.” Journal of the ACM (JACM) 28.3 (1981): 615-628.

[9] Pavlichin, Dmitri S., Tsachy Weissman, and Golan Yona. “The human genome contracts again.” Bioinformatics 29.17 (2013): 2199-2202.

[10] Buhrman, Harry, et al. “Are bitvectors optimal?” SIAM Journal on Computing 31.6 (2002): 1723-1744.

[11] Pananjady, Ashwin, and Thomas A. Courtade. “Compressing sparse sequences under local decodability constraints.” 2015 IEEE International Symposium on Information Theory (ISIT). IEEE, 2015.

[12] Yamamoto, Koichi. ”Logarithmic order of free distributive lattice.” Journal of the Mathematical Society of Japan 6.3-4 (1954): 343-353.

[13] Alon, Noga, and Uriel Feige. “On the power of two, three and four probes.” Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms. Society for Industrial and Applied Mathematics, 2009.

[14] Nicholson, Patrick K., Venkatesh Raman, and S. Srinivasa Rao. ”A survey of data structures in the bitprobe model.” Space-Efficient Data Structures, Streams, and Algorithms. Springer Berlin Heidelberg, 2013. 303-318.

[15] Ziv, Jacob, and Abraham Lempel, Compression of individual sequences via variable-rate coding, IEEE Trans. Info. Theory, vol. IT-24, pp. 530 - 536, September 1978

[16] Kostina, Victoria, and Sergio Verd. ”Fixed-length lossy compression in the finite blocklength regime.” IEEE Transactions on Information Theory 58.6 (2012): 3309-3338.