SOLUTIONS TO ASSIGNMENT 5

Exercise 1 (List decoding from erasures)**.** We say that a code is (p, L)-erasure list-decodable if for any vector $\mathbf{y} \in \{0, 1, *\}^n$ (where $*$ denotes the erasure symbol) with at most pn erasures, there are at most L codewords that agree with $\mathbf{y}$ in the unerased positions. For any vector $\mathbf{c}$ and $T \subset [n]$, let $\mathbf{c}_T$ denote the restriction of $\mathbf{c}$ to T, i.e., it is the $|T|$-length vector $(c_i : i \in T)$. Formally, a code $\mathcal{C} \subset \mathbb{F}_2^n$ is (p, L)-erasure list-decodable if for every $T \subset [n]$ with $|T| \geq (1-p)n$, and $\mathbf{y}' \in \{0,1\}^{|T|}$, we have

$$|\{\mathbf{c} \in \mathcal{C} : \mathbf{c}_T = \mathbf{y}'\}| \leq L.$$

Prove the following:

1. If $\mathcal{C}$ has minimum distance d, then it is $\left(\frac{d-1}{n}, 1\right)$-list decodable.
2. For every $\epsilon > 0$, there exists a (p, L)-erasure list decodable code of rate

$$R \geq \frac{L}{L+1}(1-p) - \frac{1}{L} - \epsilon$$

Hint: Use random codes. For a fixed $T, \mathbf{y}'$, compute the probability that the codeword for a fixed message is equal to $\mathbf{y}$ when restricted to T. Do this for $L+1$ messages. Then take a union bound over messages, $\mathbf{y}$', and T.

3. For every $\epsilon > 0$, there exists a linear (p, L)-erasure list-decodable code of rate

$$R \geq \frac{J-1}{J}(1-p) - \frac{1}{J-1} - \epsilon$$

where $J = \lceil \log_2(L+1) \rceil$.

4. Show that if a code of rate $1 - p + \epsilon$ is (p, L)-erasure list-decodable, then $L = 2^{\Omega(n)}$.

Solution. 1. If the minimum distance of a code is d, then it can correct every pattern of at most $d-1$ erasures. Hence, it is $(\frac{d-1}{n}, 1)$-list decodable.

2. Define $B(\mathbf{y}, T) \triangleq \{\mathbf{x} \in \mathbb{F}_2^n : \mathbf{x}_T = \mathbf{y}'\}$. We need to show that

$$\Pr_{\mathcal{C}}[\exists T, \mathbf{y}' : |\mathcal{C} \cap B(\mathbf{y}', T)| \geq L+1] = o(1).$$

Fix $T, \mathbf{y}'$. Let $|T| = t$. For any message $\mathbf{m}$,

$$\Pr[\mathbf{c}(\mathbf{m}) \in B(\mathbf{y}', T)] = \frac{1}{2^t},$$

and for any fixed set of $L+1$ messages

$$\Pr[\mathbf{c}(\mathbf{m}_1), \ldots, \mathbf{m}_{L+1} \in B(\mathbf{y}', T)] = \frac{1}{2^{t(L+1)}}.$$

Taking union bound over message sets, T and $\mathbf{y}'$,

$$\begin{aligned}
&\Pr_{\mathcal{C}}[\exists T, \mathbf{y}' : |\mathcal{C} \cap B(\mathbf{y}', T)| \geq L+1] \\
&\leq \sum_{t=(1-p)n}^{n} \binom{n}{t} 2^t \binom{2^{nR}}{L+1} \frac{1}{2^{t(L+1)}} \\
&\leq (1-p)n \times 2^n \times 2^{nR(L+1)} \times 2^{-nL(1-p)}
\end{aligned}$$

The above quantity is $2^{\Theta(n)}$ as long as $R < \frac{L}{L+1}(1-p) - \frac{1}{L+1}$.

3. This is very similar to Problem 1 and Problem 2.2.

4. Suppose $\mathcal{C}$ is any code of rate $1-p+\epsilon$ and minimum list size L. Choose T to be a fixed subset of $[n]$ having size $n(1-p)$, and $\mathbf{y}'$ a random vector of length $n(1-p)$ with i.i.d. Bernoulli(1/2) components. Fix any codeword $\mathbf{c} \in \mathcal{C}$. This is in $B(\mathbf{y}', T)$ if $\mathbf{c}_T = \mathbf{y}'$. Hence,

$$\Pr_{\mathbf{y}}[\mathbf{c} \in B(\mathbf{y}', T)] = \frac{1}{2^{n(1-p)}}.$$

Let $\xi = \sum_{\mathbf{c}\in\mathcal{C}} 1_{\{\mathbf{c}\in B(\mathbf{y}',T)\}}$ be the number of codewords in the ball. Then,

$$\mathbb{E}_{\mathbf{y}'}[\xi] = \sum_{\mathbf{c}\in\mathcal{C}} \Pr[\mathbf{c} \in B(\mathbf{y}', T)] = 2^{nR}/2^{n(1-p)} = 2^{n\epsilon}.$$

Therefore, there exists at least one $\mathbf{y}'$ such that $|B(\mathbf{y}', T) \cap \mathcal{C}| \geq 2^{n\epsilon}$.

Exercise 2 (Random graphs are good expanders)**.** In this exercise, we will show that a randomly chosen bipartite graph is a good expander with high probability. Recall that a bipartite graph with n left vertices, m right vertices, and left degree D is a (γ, α) expander if for all subsets S of left vertices with $|S| \leq \gamma n$, we have $|N(S)| > \alpha|S|$. Here, $N(S)$ denotes the set of neighbours of S.

Let us pick a random graph in the following manner: For each left vertex, pick D neighbours uniformly at random from the set of all $\binom{m}{D}$ subsets of right vertices. This is done independently for each vertex. Call the resulting random graph $\mathcal{G}$. We want to show that for all sufficiently large n, and $m > 3n/4$, $D > 32$, $\gamma = 1/10$, $\alpha = 5D/8$

$$\Pr[\mathcal{G} \text{ is not a } (\gamma, \alpha) \text{ expander}] = o(1).$$

1. Choose any set of left vertices S and set of right vertices T, with $|S| = s \leq \gamma n$ and $|T| \leq \alpha s$. Compute the probability that $N(S) \subset T$.

2. Argue that

$$\Pr[\mathcal{G} \text{ is not a } (\gamma, \alpha) \text{ expander}] = \Pr[\exists S \subset \mathcal{L}, T \subset \mathcal{R} : |S| \leq \gamma n, |T| \leq \alpha|S|,\ N(S) \subset T]$$

where $\mathcal{L}, \mathcal{R}$ denote the set of left and right vertices respectively.

3. Use the first two parts to get an upper bound on the probability that $\mathcal{G}$ is not an expander.

4. Using the bound $\binom{a}{b} \leq \left(\frac{ea}{b}\right)^b$, prove that as long as $m > 3n/4$, $D > 32$, $\gamma = 1/10$, $\alpha = 5D/8$, the probability that $\mathcal{G}$ is not an expander is $o(1)$.

Solution. 1. Each left vertex chooses D neighbours uniformly at random, and independently of other vertices.

$$\Pr[N(S) \subset T] \leq \left(\frac{\binom{t}{D}}{\binom{m}{D}} \right)^s$$

2. This is straightforward. The random graph $\mathcal{G}$ is not an expander if and only if there exists some subset S of left vertices with $|S| \leq \gamma n$ whose neighbourhood is of size less than or equal to $\alpha|S|$.

3. Using the union bound,

$$\Pr[\mathcal{G} \text{ is not an expander}] \leq \sum_{s=2}^{\gamma n} \sum_{t=D}^{\alpha s} \binom{n}{s} \binom{m}{t} \left(\frac{\binom{t}{D}}{\binom{m}{D}} \right)^s$$

4. The term inside the summation is maximized when $s = \gamma n$ and $t = \alpha\gamma n$. Using the bound on the binomial coefficient,

$$\begin{aligned}
\Pr[\mathcal{G} \text{ is not an expander}] &\leq (\gamma n)(\alpha\gamma n) \binom{n}{\gamma n} \binom{m}{\alpha\gamma n} \left(\frac{\binom{\alpha\gamma n}{D}}{\binom{m}{D}} \right)^{\gamma n} \\
&\leq (\gamma n)(\alpha\gamma n) \left(\frac{e}{\gamma}\right)^{\gamma n} \left(\frac{me}{\alpha\gamma n}\right)^{\alpha\gamma n} \left(\frac{e\alpha\gamma n}{m}\right)^{\gamma D n} \\
&= \alpha\gamma^2 n^2 e^{n\gamma(+\alpha+D)} \left(\frac{1}{\gamma}\right)^{\gamma n} \left(\frac{\alpha\gamma n}{m}\right)^{\gamma n(D-\alpha)} \\
&\leq \alpha\gamma^2 n^2 e^{n(1+13D/8)/10} 10^{n/10} \times \left(\frac{D}{12}\right)^{3nD/80} \\
&= o(1).
\end{aligned}$$

Exercise 3. Let $\mathcal{G}$ be an $(n, m, D, \gamma, \alpha)$ expander graph. Let $\alpha = D(1 - \epsilon)$ for some $0 < \epsilon < 1/2$. Given any set of left vertices S, a right vertex v is said to be a unique neighbour of S if it is adjacent to exactly one vertex in S. Let $U(S)$ denote the set of unique neighbours of S.

1. Fix any set of left vertices S such that $|S| \leq \gamma n$. How many edges leave S? Using this, compute an upper bound on the number of vertices in $N(S)$ that have more than one incident edge from S.

2. Use the above to argue that $|U(S)| \geq D(1-2\epsilon)|S|$.

3. Use the second part to argue that the minimum distance of the corresponding expander code is at least γn.

   Hint: Choose any nonzero codeword and label the left vertices by the codeword bits. Let S be the support set of vertices labelled 1. What can you say about $U(S)$?

Solution. 1. The number of edges leaving S is $D|S|$. Since the graph is an expander, S has at least $(1-\epsilon)D|S|$ neighbours. Since there are $D|S|$ edges, by the pigeonhole principle, at most $\epsilon D|S|$ neighbours of S can have more than one incident edge from S.

2. S has at least $(1-\epsilon)D|S|$ neighbours, of which at most $\epsilon D|S|$ neighbours of S can have more than one incident edge from S. Therefore, $|U(S)| \geq (1-2\epsilon)D|S|$.

3. Let S be the support of the nonzero codeword (vertices labelled 1) of least Hamming weight. This is a valid codeword if and only if $U(S)$ is empty. This is because for any parity check in $U(S)$, exactly one neighbour comes from S and the rest are outside S. Hence, this equation cannot be satisfied. Using part 2, we know that for every S of size less than or equal to γn, we have $|U(S)| > 0$. Hence, the minimum distance is at least γn.

Exercise 4. Let $\mathcal{G}$ be an $(n, m, D, \gamma, D(1-\epsilon))$ expander. We will now show that the minimum distance of the corresponding expander code is $\geq 2\gamma(1-\epsilon)n$. We will prove this by contradiction. For the sake of contradiction, suppose that c^n is a nonzero codeword of Hamming weight less than $2\gamma(1-\epsilon)n$.

- Label the set of left vertices of $\mathcal{G}$ using c^n, and let S be the set of vertices labelled 1. Note that $|S|$ is equal to the Hamming weight of c^n. What is the size of $U(S)$?

- Pick $Q \subset S$ such that $|Q| = \gamma n$. Compute the size of $U(Q)$ and $N(S \backslash Q)$ and argue that $|U(S)| > 0$.

Solution. 1. Using the same argument as in the last part of the previous question, $|U(S)|$ must be 0 for a valid codeword.

2. Since $|Q| = \gamma n$ and the graph is an expander, we have $|U(Q)| \geq (1-2\epsilon)\gamma Dn$. Now, $|S \backslash Q| < (1-2\epsilon)\gamma n$. The number of edges leaving this is strictly less than $(1-2\epsilon)\gamma Dn$. Therefore, the number of vertices in $U(Q)$ that might have a neighbour in $S \backslash Q$ is strictly less than $(1-2\epsilon)\gamma Dn$, which implies that $U(S) > 0$.

Using the above, we see that the minimum distance is at least $2\gamma(1-\epsilon)n$.

Exercise 5 (Encoding/decoding complexity of expander codes)**.** Expander codes have low encoding and decoding complexity.

- What is the encoding complexity of an expander code?

- What is the computational complexity in each iteration of decoding an expander code? Use this to find the worst-case computational complexity assuming that the number of errors is less than $\gamma(1-2\epsilon)n$.

Solution. 1. An expander code is linear. Hence, the encoding complexity is $O(n^2 R)$.

2. In each iteration, we need to find a left vertex which has more unsatisfied neighbours than satisfied ones, and also update the parities at the right vertices. The complexity is $O(n)$.

   If the number of errors is less than $\gamma(1-2\epsilon)n$, then the algorithm is guaranteed to terminate in $O(n)$ steps. Hence, the overall decoding complexity is $O(n^2)$.