Problem Set 4

Instructions

See Problem Set 1. Additionally:

1. This problem set has a number of “not to be turned in parts”. You must do these parts for yourselves (for good learning). But you can skip the writing which can be tedious and long. The ”to be turned in parts” of these questions will carry enough information for us to test your understanding.

2. Feel free to use piazza to seek collaborators. Collaboration is hopefully feasible and is important for learning!

3. If you are having difficulty completing the psets on time, given the coronavirus disruptions, please write to Madhu.

Problems

1. (List-decodability vs. distance): Recall that a code $C \subseteq \Sigma^n$ is $(\rho, L)$-list-decodable if for every $w \in \Sigma^n$ we have $|C \cap \text{Ball}_\Sigma(w, \rho n)| \leq L$. The goal of this problem is to show that if a code has relative distance $\delta - 1 - \alpha$, then it is $(\tau, O(1))$-list-decodable provided $\tau < 1 - \sqrt{\alpha}$.

   Specifically, let $\alpha, \delta$ and $\tau$ be as above. Fix $n$ and let $d = \delta n$, $t = \tau n$, $a = \alpha n$ and $b = n - t$. Suppose $C \subseteq \Sigma^n$ is code of distance $d$ and $w \in \Sigma^n$ and $c_1, \ldots, c_L$ are codewords of $C$ such that $\Delta(c_j, w) \leq t$ for every $j \in \{1, \ldots, L\}$. We will show below that $L \leq \frac{1 - \tau - \alpha}{(1 - \tau)^2 - \alpha}$, provided $(1 - \tau)^2 > \alpha$.

   (a) Define a bipartite graph $B = ([n], [L], E)$ where $(i, j) \in E$ if $w_i = (c)_j$.

   (b) Prove that $B$ does not contain a $K_{a+1, 2}$, i.e., the complete bipartite graph with $a + 1$ left vertices and 2 right vertices.

   (c) Prove that if $B$ has $m$ edges, then it must contain $K_{a', 2}$ for $a' = \frac{m(m-n)}{nL(L-1)}$.

   (d) Use the above to prove that $L \leq \frac{1 - \tau - \alpha}{(1 - \tau)^2 - \alpha}$, provided $(1 - \tau)^2 > \alpha$.

2. (List-decodability vs. Rate): Let $\text{Vol}_q(n, r)$ denote the volume of a Hamming ball of radius $r$ in an $n$-dimensional space over a $q$-ary alphabet. Define $H_q(\rho) = \lim_{n \to \infty} \left\{ \frac{\log_q \text{Vol}_q(n, \rho n)}{n} \right\}$.

   (In other words we are defining $H_q(\rho)$ so that $\text{Vol}_q(n, \rho n) \approx q^{H_q(\rho)n}$.)
(a) Prove that for every family of $q$-ary codes $\{C_n\}$ where $C_n \subseteq \Sigma^n$ and $|\Sigma| = q$ and $C_n$ is $(p, \text{poly}(n))$-list-decodable, we must have $\lim_{n \to \infty} (\text{Rate}(C_n)) \leq 1 - H_q(p)$. (Hint: Use the converse to Shannon’s coding theorem.)

(b) Prove the above is tight: Specifically prove that for every $R$ and $\rho$ such that $R + H_q(\rho) < 1$, there exists $L$ such that for all sufficiently large $n$, we have that a random code $C \subseteq \Sigma^n$ of rate $R$ (i.e., of size $|C| = q^Rn$) is $(\rho, L)$-list-decodable, with high probability.

3. (List-decoding of RS Codes): Recall that we saw in the class how to list decode RS code with rate $R$ and $1 - 2\sqrt{R}$ fraction of errors. The goal of this problem is to show that with a small modification on the decoding algorithm, one can actually correct $1 - \sqrt{2R}$ fraction of errors.

To be specific, recall that the “list-decoding” problem for Reed-Solomon codes is the following: Given as input, parameters $n, k, t$ and a set of $n$ points $\{(\alpha_i, \beta_i) \in \mathbb{F}_q^2 | i \in [n]\}$ where $\alpha_1, \ldots, \alpha_n$ are distinct field elements and $\beta = (\beta_1, \ldots, \beta_n)$ denotes the received word, output a short list containing all polynomials $P$ of degree less than $k$ such that $|\{i \in [n] | P(\alpha_i) - \beta_i\}| \leq t$.

We have seen how to solve this problem for $t = n - 2\sqrt{kn}$ and now we want to improve this to $t = n - \sqrt{2kn}$. Recall that the algorithm from class involved two steps: (1) Find a non-zero polynomial $Q(x, y)$ of appropriately small degree such that $Q(\alpha_i, \beta_i) = 0$ for every $i \in [n]$. (2) Factor this polynomial and include $P$ in the output if $y - P(x)$ divides $Q(x, y)$.

Our modification will be obtained by carefully picking a set of monomials $M \subseteq \{x^iy^j | i, j \geq 0\}$ and requiring that $Q$ be only supported on the monomials of $M$. (I.e., If $Q(x, y) = \sum_{i,j} c_{ij}x^i y^j$ and $c_{ij} \neq 0$ for some $i, j$ then $x^iy^j \in M$.)

Describe a set of monomials $M$ that allows you to solve the list-decoding algorithm above with $t = n - \sqrt{2kn}$. (No need to write the details of all remaining steps, but do verify them for yourselves.)

4. (List-recovery and Concatenated Codes): Recall that a Folded-Reed-Solomon (FRS) code is given by $\mathbb{F}_q$, integers $n, k, r$ and elements $\lambda, \alpha_1, \ldots, \alpha_n \in \mathbb{F}_q$ such that the elements of the set $\{\lambda^j \alpha_i | i \in [n], 0 \leq j \leq r\}$ are all distinct. The message is an element of $\mathbb{F}_q^k$ representing a polynomial $P$ of degree less than $k$. Its encoding is given by $(P(\alpha_1), P(\lambda \alpha_1), \ldots, P(\lambda^{r-1} \alpha_1))_{i \in [n]} \in (\mathbb{F}_q^r)^n$.

(a) For integers $\ell$ and $t$, the $(\ell, t)$-list-recovery problem for the FRS code consists of inputs $S_1, \ldots, S_n$ where for every $i$, $S_i \subseteq \mathbb{F}_q^\ell$ with $|S_i| \leq \ell$; and the goal is to output a list of all polynomials $P$ such that $|\{i | [P(\alpha_i), P(\lambda \alpha_i), \ldots, P(\lambda^{r-1} \alpha_i)] \in S_i\}| \geq n - t$. (To make sense of this problem, try to understand what problem corresponds to the special case of $\ell = 1$.)

i. (Not to be turned in): Verify that for every $\epsilon > 0$, rate $R$ and input-list size $\ell$, for all sufficiently large $n$ we can find FRS of rate at least $R$, that is $(\ell, (R + \epsilon)n)$-list recoverable in polynomial time.

ii. (To be turned in): How do you choose the parameters $r$ and $k$ as a function of $\epsilon, R$ and $n$ for the Folded Reed-Soloman code? Justify the choice.

(b) Let $q = p^f$ for some prime power $p$ and consider concatenating a FRS code $C_1$ over alphabet $\mathbb{F}_q$ with a $p$-ary code $C_2$ with $p^t$ messages. Let $C = C_1 \circ C_2$ denote the concatenated code.
i. (To be turned in): Informally describe a list-decoding algorithm for $C$ that exploits the list-recoverability of $C_1$. (By informal we mean that you should describe the structure though you can skip the exact choice of parameters etc.)

ii. (Not to be turned in): Given a rate $R$ and parameter $\epsilon > 0$ show that there exists a constant $p$ such that for sufficiently large $n$ you can get $p$-ary codes $C'$ using the concatenation above satisfying the conditions that $C'$ has rate $R$ and $C'$ is list-decodable from $1 - R - \epsilon$ fraction error in polynomial time.

iii. (To be turned in): What are the parameters of the codes $C_1$ and $C_2$ in terms of parameters $R$, $\epsilon$ and $n$ of the code $C'$? Justify your choice.