CS250/EE387: Error Correcting Codes M. Wootters
Problem Set 3 Winter 2022
Due: by 11:59pm on Friday February 18, 2022, on Gradescope.

Instructions:

- Please complete all problems in Section 1.
- Try to complete 3 of the problems in Section 2. You are welcome to do more than 3, but please indicate which 3 you want graded.
- No problems in Section 3 are required, but they might be fun to think about (some might be open-ended).
- Problems are labeled with the class number after which you should be able to do them. (This is to aid your time management since all HWs are posted up front).

Guidelines/rules:

- You are encouraged to work in groups (up to 3-ish); each group should turn in one HW assignment.
- You and your group may collaborate on problems in Sections 1 and 2 with other members of the class; please acknowledge your collaborators. You may consult lecture notes, Essential Coding Theory and other posted readings, but please do not use any other written resources (that is, please do not Google for the answers to the questions). It is fine to use computational resources like Sage or Mathematica if you want to.
- You may collaborate on Section 3 problems with anyone, whether or not they are in the class; please acknowledge your collaborators. You may also use whatever resources you want: Googling, reading research papers, etc, is fine.

Typing up your solutions in LaTeX is encouraged (but I don't type up my lecture notes, so I can't be too strict). Legibility and complete sentences are required.

Section 1

1. (4 pts, Class 10) Suppose that $q = 2$. Recall that the Johnson bound implies that there are codes of rate R that are (p, L)-list-decodable (for reasonable L) as long as

$$R < 1 - H_2(2p(1-p)).$$

   What is this, quantitatively, for $p = 2/5$? How does this compare to the list-decoding capacity theorem?

2. (6 pts, Class 10) Let $p = 1/8$. At what rate R can you (or rather, can we so far in this class) guarantee that:

   (a) There exists a family of binary codes of rate R that can correct a p-fraction of adversarial errors?

   (b) There exists a family of binary codes of rate R that can *efficiently* correct a p-fraction of adversarial errors?

   (c) There exists a family of binary codes of rate R that can correct a p-fraction of *random* errors?

   (d) There exists a family of binary codes of rate R that can *l*ist-decode *up to a p-fraction of adversarial errors?*

   Explain your answers. (Also, please give your answers in the format "≈ 0.33" rather than "$\sqrt{H_2^{-1}(1/2)}$.")

Section 2

(Section 2 problems are worth 10 points each; please do at least 3 of them.)

1. (**MDS-like codes**, Class 4) In this problem, we will consider a number-theoretic counterpart of Reed-Solomon codes. Let $1 \leq k < n$ be integers and let $p_1 < p_2 < \cdots < p_n$ be n distinct primes. Let $K = \prod_{i=1}^{k} p_i$. Let $\mathbb{Z}_M$ denote the integers modulo M (that is, the set $\{0, 1, \ldots, M-1\}$ with addition and multiplication mod M). Consider the "code" defined by the encoding map $E : \mathbb{Z}_K \to \mathbb{Z}_{p_1} \times \mathbb{Z}_{p_2} \times \cdots \times \mathbb{Z}_{p_n}$ given by

$$E(m) = (m \mod p_1, m \mod p_2, \ldots, m \mod p_n).$$

That is, let $\mathcal{C} \subseteq \mathbb{Z}_{p_1} \times \cdots \times \mathbb{Z}_{p_n}$ be defined as the image of E. (Notice that $\mathcal{C}$ is not a code under our definition since it has different alphabets for different symbols, which is why "code" above is in quotes.)

   (a) Suppose that $m, m' \in \mathbb{Z}_K$ are distinct messages. Let $A \subseteq [n]$ be the set of indices i so that $m = m' \mod p_i$. Prove that $\prod_{i \in A} p_i$ divides $m - m'$, where we treat m and m' as integers in $\{0, \ldots, K-1\}$.

   (b) Define the "distance" of the "code" above to be

$$d = \min_{m \neq m' \in \mathbb{Z}_K} \sum_{i=1}^{n} \mathbf{1}_{E(m)_i \neq E(m')_i}.$$

   Use part (a) to show that the "distance" of $\mathcal{C}$ is $d = n - k + 1$.

   (c) (Open-ended, you'll get full points for anything reasonable). Do you think this code counts as "meeting the Singleton bound?" Why or why not? (Notice that this is not well-defined since we haven't defined "rate" or "message length" for a code where each symbol has a different alphabet, and because our definition of "distance" in the previous part might not be the only way to define "distance" in this setting. Moreover, the "correct" version of the Singleton bound in this setting might be slightly different.)

2. ((ℓ, ℓ)-**list-recoverability**, Class 4)

   (a) Let $f, g \in \mathbb{F}_q[X]$ be polynomials of degree at most $k < (q-1)/2$. Suppose for every $\alpha \in \mathbb{F}_q$, we are given the sets $\{f(\alpha), g(\alpha)\}$; these sets come labeled with α, but we do not know which element of the set comes from f and which comes from g. Give an efficient (aka, polynomial in q) algorithm for recovering f and g.

   NOTE: Please do this from first principles (and/or from the second hint below), do not use the Guruswami-Sudan algorithm (which we may have seen in class by the time you get to this problem). The solution to this problem is much easier than the GS algorithm.

   Hint 1: Consider the polynomial $p(X, Y) = (Y - f(X))(Y - g(X))$.

   Hint 2/supplementary resource: You may use the following fact: if you have a polynomial $p(X, Y)$ as in Hint 1, you can factor it to find $f(X)$ and $g(X)$ in polynomial time. In the special quadratic case above, it might be fun to figure out an algorithm to do this from scratch. If you are curious to see how it's done for general bivariate polynomials, see Section 2 here: `http://sites.math.rutgers.edu/~sk1233/courses/ANT-F14/lec10.pdf`.

   (b) Later in the course (or perhaps by now depending on when you do this PSET), we will see the following definition:

   Definition 1. *A code $\mathcal{C}$ is* (ℓ, L)-list-recoverable *if for all collections of sets $S_1, \ldots, S_n \subset \Sigma$ so that $|S_i| \leq \ell$ for all i, there are at most L codewords $c \in \mathcal{C}$ so that $c_i \in S_i$ for all i.*

In part (a), you saw that full-length RS codes (aka, RS codes over $\mathbb{F}_q$ with length $n = q$) of an appropriate dimension have the following property of a code $\mathcal{C} \subseteq \Sigma^n$:

> For any $c, c' \in \mathcal{C}$, given the unordered sets $\{c_i, c'_i\}$ for $i = 1, \ldots, n$, it is possible to recover c, c'.

What is the relationship between this property and $(2,2)$-list-recoverability? (That is, are they the same? Is one stronger than the other? Are they uncomparable?)

(c) Suppose that a family of MDS[1] codes is (ℓ, ℓ)-list recoverable. Show that the rate of the code satisfies $R \leq 1/\ell + o(1)$.

3. (**Efficient group testing algorithms**, Class 8) Let $\mathcal{C} \subset \mathbb{F}_q^n$ be any code, and let $A \in \{0,1\}^{nq \times |\mathcal{C}|}$ be the group testing matrix obtained by the Kautz-Singleton construction we saw in the lecture videos/notes. (That is, the columns of A are codewords of $\mathcal{C}$ concatenated with the identity code).

   (a) Suppose that $\mathcal{C}$ is (d,d)-list-recoverable, as per Definition 1, and suppose that $|\mathcal{C}| > d$. Show that A can identify up to d defective items.

   NOTE: For partial credit, (say, 8 out of the 10 points for this problem, assuming you do part (b)) you can prove the (easier) statement that A can identify any set of *exactly* d defectives.

   (b) Suppose that $\mathcal{C}$ has a (d,d)-list-recovery algorithm that runs in time $\text{poly}(n)$. Give a sublinear-time algorithm (sublinear in $|\mathcal{C}|$) for the corresponding group testing scheme. (That is, given the outputs of the tests, give an algorithm to identify the $\leq d$ defective items in time $\text{poly}(nq)$, where nq is the number of tests in A; notice this is much less than $|\mathcal{C}|$ which is the total number of items that were pooled.)

4. (**Codes which are good for random errors aren't *terrible* for worst-case errors**, Class 9) Let $\mathcal{C} \subset \{0,1\}^n$ be a code of rate k/n, with encoding and decoding algorithms $E : \{0,1\}^k \to \{0,1\}^n$ and $D : \{0,1\}^n \to \{0,1\}^k$. Fix any constant $\gamma > 0$ and $p \in (0, 1/2)$. Suppose that $\mathcal{C}$ has error probability at most $2^{-\gamma n}$ on the BSC_p channel: for all $x \in \{0,1\}^k$,

$$\mathbb{P}_{BSC_p}\{D(BSC_p(E(x))) \neq x\} \leq 2^{-\gamma n}.$$

   Show that there is some constant C_p, which depends only on p, so that the relative distance of $\mathcal{C}$ is at least $C_p \cdot \gamma$.

   Clarification: You may assume that γ is a fixed constant and that n is sufficiently large in terms of γ.

   Hint: There is probably a correct solution that has lots of $H(p)$'s in it and a bunch of mucky algebra. But there's also a correct solution that doesn't have any $H(p)$'s and no mucky algebra.

   Second Hint: Suppose you have two codewords c, c' that disagree on $\{1, \ldots, \delta n\}$. Consider the set of vectors $y \in \{0,1\}^n$ that agree with c on $\{1, \ldots, \delta n/2\}$ and agree with c' on $\{\delta n/2 + 1, \ldots, \delta n\}$. (And say that δn is an even integer—you can assume this if you want). What is the probability that $BSC_p(c)$ ends up in this set? What about $BSC_p(c')$?

5. (**Average-radius version of list-decoding**, Class 10) Consider the following Theorem.

   Theorem 1. *Let $\mathcal{C} \subset \mathbb{F}_2^n$ be a binary code, and fix a parameter L. Then for any $\Lambda \subset \mathcal{C}$ with $|\Lambda| = L$,*

$$\min_{z \in \mathbb{F}_2^n} \frac{1}{L} \sum_{c \in \Lambda} \delta(c, z) \geq \frac{1}{2} \left(1 - \sqrt{1 - \frac{2}{L^2} \sum_{c' \neq c'' \in \Lambda} \delta(c', c'')} \right).$$

   (a) Show that Theorem 1 implies that:

[1] Recall that an MDS (Maximum Distance Separable) code is a linear code that meets the Singleton bound; that is, the distance is $d = n - k + 1$.

Let C be a binary code with relative distance δ. For any $p < J_2(\delta)$, for sufficiently large n, C is $(p, \text{poly}(n))$-list-decodable.

Note that this is similar to the version of the Johnson bound that we stated in the lecture videos/notes (for $q = 2$).

Hint: To try to parse the left-hand side, explain to yourself why $(p, L-1)$-list-decodability is the same thing as: for all sets $\Lambda \subseteq \mathcal{C}$ with $|\Lambda| = L$,

$$\min_{z \in \mathbb{F}_2^n} \max_{c \in \Lambda} \delta(c, z) \geq p.$$

Update: a typo was fixed in the hint, it used to say $(p, L+1)$-list-decodability and now reads $(p, L-1)$-list-decodability.

(b) Let $\Phi \in (\pm 1)^{n \times 2^k}$ be the matrix whose columns are indexed by $c \in \mathcal{C}$, so that $\Phi_{j,c} = (-1)^{c_j}$. Let $\mathbf{1}_\Lambda$ denote the indicator vector for a set Λ. Relate the quantity on the left-hand-side of Theorem 1 to the ℓ_1 norm of $\|\Phi \mathbf{1}_\Lambda\|_1$.

(c) Conclude that

$$L - \min_z \sum_{c \in \Lambda} \delta(c, z) \leq \frac{1}{2}\left(L + \frac{1}{\sqrt{n}} \|\Phi \mathbf{1}_\Lambda\|_2\right),$$

Hint: The Cauchy-Schwarz Inequality is your friend.

(d) Prove Theorem 1.

Section 3

1. In Section 2 we defined (ℓ, ℓ)-list-recoverability at you showed that any MDS code that is (ℓ, ℓ)-list-recoverable has rate at most $1/\ell$. Can you show this for any *linear* code? What about for any (possibly non-linear) code?

2. In class we proved one side of the list-decoding capacity theorem by looking at a completely random code. Recall that we proved the GV bound by looking at a random *linear* code. Does the proof we saw for list-decoding work if you look at a random linear code? Can you show that there exists a linear code that approaches list-decoding capacity? Can you show that a random linear does with high probability?

3. The capacity on the BSC_p is $1 - H(p)$, which is the same as list-decoding capacity. Why are these the same? Can you come up with a formal relationship between list-decoding and decoding on the BSC?

4. Can you find a Reed-Solomon code (that is, a way of choosing evaluation points) that provably does *not* approach list-decoding capacity? Can you find a Reed-Solomon code that does? (Or even prove that one exists?)

5. In Section 2, we saw MDS-like codes based on the Chinese Remainder Theorem. Can you adapt the Guruswami-Sudan algorithm (Class 11) to work for these codes?