CS250/EE387: Error Correcting Codes M. Wootters
Problem Set 1 Winter 2022
Due: by 11:59pm Friday, January 14 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

(Do all of these problems.)

1. (8 pts, Class 2) Please answer the following questions with a brief justification of your answer. Let $\mathcal{C}$ be the binary linear code with generator matrix

$$G = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

   (a) What is the dimension of $\mathcal{C}$?

   (b) Find a parity-check matrix for $\mathcal{C}$.

   (c) What is the distance of $\mathcal{C}$?

   (d) Find another generator matrix G' for the same code $\mathcal{C}$ that represents a systematic encoding; that is, so that the encoding map $x \mapsto G'x$ has the form $(x_1, x_2, x_3) \mapsto (x_1, x_2, x_3, a, b)$ for some $a, b \in \mathbb{F}_2$.

2. (4 pts, Class 1) Suppose that $\mathcal{C}$ is a code of length n and distance d. Consider a noise model that has both errors and erasures: that is, when a codeword $c \in \Sigma^n$ is transmitted, the recieved word $y \in (\Sigma \cup \{\perp\})^n$ has

$$y_i \in \Sigma \setminus \{c_i\}$$

for at most e positions i, and $y_i = \perp$ for at most s positions i. Show that as long as $2e + s < d$, then c can be determined from y. (Above, the symbol $\perp$ is a place-holder symbol not in Σ, which corresponds to "erasure.")

Section 2

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

Some of the problems in Section 2 reference the *Hamming Code* $\mathcal{H}$ which we discussed during the in-class work in Class 2. Recall the following definition:

Definition 1. *Let $n = 2^r - 1$ for some integer r. The Hamming code $\mathcal{H}_r$ of length n is the code whose parity-check matrix $H_r \in \mathbb{F}_2^{r \times n}$ is the matrix which has every nonzero vector in $\{0,1\}^r$ as its columns.*

1. (**Fun with linear algebra over finite fields**, Class 2)

   Let q be a power of a prime and $\mathbb{F}_q$ be the finite field of order q. In this exercise you'll rigorously prove a few statements of the flavor "linear algebra works over finite fields." So, for the following problems, **use only the definitions we've seen and fact that $\mathbb{F}_q$ is a finite field** (that is, you may use the definitions of "linear independent," "subspace," and so on, and the field axioms, but do not appeal to linear algebra "facts" that you may know). (See the lecture notes for the list of definitions that "we've seen.")

   Let $V \subseteq \mathbb{F}_q^N$ be a subspace over $\mathbb{F}_q$. Recall that a **basis** for V is any collection of vectors $a_1, \ldots, a_n \in V$ so that the a_i are linearly independent and $\text{span}(a_1, \ldots, a_n) = V$. Let $\mathcal{A} = \{a_1, \ldots, a_n\}$ be a basis for V.

   (a) Prove the following useful statements about finite fields (using the field axioms):
   - Suppose that $\alpha \in \mathbb{F}_q$. Then $\alpha \cdot 0 = 0$.
   - Suppose that $\alpha, \beta \in \mathbb{F}_q$ are both nonzero. Then $\alpha \cdot \beta \neq 0$.

   (b) Suppose that $b_1, \ldots, b_m \in V$ are linearly independent, with $m \leq n$. Prove that there exists an ordering $a_1, \ldots, a_n$ of $\mathcal{A}$ so that, for all $k \in \{1, \ldots, m\}$, $\{b_1, \ldots, b_k, a_{k+1}, \ldots, a_n\}$ is a basis for V. (Hint: use induction on k; part (a) might be useful).

   (c) Show (using part (b)) that any two bases of V must have the same cardinality. (That is, our definition of "dimension" makes sense).

2. (**Perfect Codes**, Class 2) We say a code $\mathcal{C}$ over $\mathbb{F}_q$ is e-perfect if it has distance $2e+1$ and meets the Hamming bound:

$$q^{n-k} = \text{Vol}_q(n, e).$$

   (a) Suppose that $\mathcal{C}$ is a binary linear code of length n and dimension k, with parity-check matrix $H \in \mathbb{F}_2^{n-k \times n}$. Show that $\mathcal{C}$ is e-perfect if and only if, for all $v \in \mathbb{F}_2^{n-k}$, there is a unique way to write v as a sum of at most e columns of H.

   (b) Conclude that the Hamming code is the only 1-perfect binary linear code. (Up to permutations of codeword symbols).

   (c) Suppose that $\mathcal{C} \subset \mathbb{F}_2^n$ is e-perfect. Show that the nonzero codewords in $\mathcal{C}^\perp$ take on at most e distinct weights. (Recall that the *weight* of a vector $x \in \mathbb{F}_2^n$ is the number of nonzero entries).

[**Note: This may be the most difficult part of Section 2. For partial credit you can just show this for** $e = 1$**.**] (Hint 1: Try it first for $e = 1$. Hint 2: What can you say about the code whose parity-check matrix has as columns all $(\leq e)$-wise sums of columns of H? And, Hint 3: Use part (a)).

3. (**Another lower bound**, Class 2)

   (a) Show that if there exists a linear $(n, k, d)_q$ code, then there also exists a linear $(n - d, k - 1, d')_q$ code for some $d' \geq \lceil d/q \rceil$.

   You may find it useful to use the following algebra fact: for any nonzero $\alpha \in \mathbb{F}_q$, the set $\{\alpha \cdot x : x \in \mathbb{F}_q\}$ is again $\mathbb{F}_q$. That is, multiplication by α just permutes the elements of the field.

   Hint: Consider dropping the non-zero positions of a minimum-weight codeword from the code.

   (b) Show that if $\mathcal{C}$ is a linear $(n, k, d)_q$ code, then

$$n \geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil .$$

   (c) Show that the bound above can sometimes be stronger than the Hamming bound. That is, find some values for n, k, d (and let $q = 2$) so that the Hamming bound is satisfied but the bound above is not. (Note: If you do not (yet) have a very good intuition for how the volume term in the Hamming bound behaves, you could enlist a computer to find these parameters. However, there is a small example ($n = 7$), and you might learn more about how the Hamming bound behaves if you do it by hand.)

4. (**Simplex Codes**, Class 2) Let $\mathcal{C}_r = \mathcal{H}_r^{\perp}$, where $\mathcal{H}_r$ is the binary Hamming code of length $n = 2^r - 1$. Notice that (by definition) a partity-check matrix for $\mathcal{H}$ is a generator matrix for $\mathcal{C}_r$. So the generator matrix for $\mathcal{C}_r$ is the $n \times r$ matrix that contains every non-zero binary vector of length r as a row. Let $G \in \mathbb{F}_2^{n \times r}$ be this generator matrix, and let $g_i \in \mathbb{F}_2^r$ denote the i'th row of G.

   The code $\mathcal{C}_r \subseteq \mathbb{F}_2^n$ is called the *Simplex Code* or (a slight variant of it is called) the *Hadamard Code*.

   (a) What is the dimension and distance of $\mathcal{C}_r$?

   (b) Suppose that $w \in \mathbb{F}_2^n$ and that there is some $c \in \mathcal{C}_r$ so that $\Delta(w, c) < n/4$, where Δ denotes Hamming distance. Explain (using your answer from part (a)) why there is no other codeword $c' \in \mathcal{C}_r$ with $c \neq c'$ so that $\Delta(w, c') < n/4$. Explain why this immediately suggests a ~~very slow~~ algorithm for finding c, given w. (UPDATE: This was described as "very slow," but in fact it is not *so* slow. Not needed for the problem, but think about how long this algorithm takes!)

   (c) Suppose that $w \in \mathbb{F}_2^n$ and that there is some $c \in \mathcal{C}_r$ so that $\Delta(w, c) < n/4$. In the rest of this part, we will motivate and analyze an algorithm for finding c, given w, that runs in polynomial time in n. (Update: this might actually not sound so interesting in light of your answer to part (b)...do the problem anyway, but think about why this algorithm is more interesting/useful than the one in part (b) :))

      i. Let $g_1 \in \mathbb{F}_2^r$ be the first row of the generator matrix G for $\mathcal{C}_r$. Let g_i be any other row of $\mathcal{C}_r$ (so $i \neq 1$). Explain why there is a unique row g_j of $\mathcal{C}_r$ so that $g_i + g_j = g_1$.

      ii. Suppose that $g_1 = g_i + g_j$ for some i, j. Explain why, for any $c \in \mathcal{C}_r$, we have $c_1 = c_i + c_j$.

      iii. Consider the following algorithm that is supposed to find c_1, given w.

```
Given input w \in F_2^n:
     votes_for_0 = 0
     if w_1 == 0:
          votes_for_0 = votes_for_0 + 1
```

```
for each i=2,...,n:
     Let g_j be the unique row of G so that g_i + g_j = g_1  (in F_2^r)
     If w_i + w_j == 0:
          votes_for_0 = votes_for_0 + 1.
If votes_for_0 > n/2:
     return "c_1 = 0"
else return "c_1 = 1"
```

What is the running time of this algorithm? (In big-Oh notation)

iv. Prove that the algorithm above indeed returns c_1. (You will need to use the assumption that $\Delta(w, c) < n/4$).
Hint 1: Use part (ii).
Hint 2: The rows of G are broken up into pairs $\{g_i, g_j\}$ and the singleton $\{g_1\}$. How many of these groups can be "corrupted", in the sense that they contain an index i so that $w_i \neq c_i$?

v. Adapt the algorithm above to give a polynomial-time algorithm that recovers all of c, given w.

5. (**Coordinated failure**, Class 2) Consider the following n-player cooperative game, for $n = 2^r - 1$. The players $1, \ldots, n$ are placed in separate rooms and are not allowed to communicate during the game. n values $x_1, \ldots, x_n \in \{0, 1\}$ are drawn uniformly at random. Player i is given $\{x_j : j \neq i\}$, (along with labels, so she knows which x_j belongs to which other player), and her goal is to guess x_i. Player i must say "0," "1," or "pass," independently of all the other players. The players collectively lose if:

   - Everyone passes, **OR**
   - Any player i reports a value that is not equal to x_i.

   The players collectively win if they do not lose. The players are allowed to strategize before the game begins.

   (a) Find a strategy where the players win with probability $1 - 1/2^r$.

   (b) Prove that your strategy is optimal.

   Hint: Try it first for $n = 3$ to get some intuition.

   Hint 2: The fact that $n = 2^r - 1$ is not an accident.

   Hint 3: The title of this problem might be a hint too.

Section 3

(These problems are not required and may be open-ended.)

1. Hamming codes are the best (that is, with the largest size $|\mathcal{C}|$) codes with distance 3 and length $n = 2^r - 1$. What about other values of n? In particular, what is the best code you can come up with with distance 3 and length 6? What about length 10? 16? 20? Do you think your constructions are optimal? What if you just need to find the best linear code with these parameters?

2. In Section 2, you showed that all e-perfect linear codes have at most e different possible weights of dual codewords. Show the converse: that any code whose dual has only e different nonzero weights is e-perfect. What is the simplest proof you can find of this fact?

3. In Section 2, you showed that the Hamming code is the only 1-perfect binary linear code. Can you extend your proof to 1-perfect binary non-linear codes? If not, can you find a 1-perfect binary non-linear code? If so, can you find (or interestingly characterize) *all* the 1-perfect binary non-linear codes of a given length?