Coding Theory - Exercise Set 1

June 17, 2011

Please give some of these problems a try. You don’t have to do them all. I will discuss some of them on Monday/Thursday and will write up a solution set at some point.

Please email me (nkaplan@math.harvard.edu) if you have any questions.

1. Please read the introductory section of Claude Shannon’s 1948 paper, “A Mathematical Theory of Communication” (up to Part I: Discrete Noiseless Systems). I just want you to note that it’s sort of an awesome achievement that none of this material really existed before and this is still pretty much the framework we use for talking about error-correcting codes today. The paper both feels entirely current and completely dated (I laughed out loud at the phrase “two punched cards should have twice the capacity of one”). This paper contains the first published use of the word ‘bits’ for ‘binary digits’ (which he attributes to Tukey). We are very lucky he made this choice, as opposed to something like, ‘b-diggies’.

2. Send me via email a fun fact about Claude Shannon. As noted in lecture, the dude was pretty fun. These could be in the form of a Youtube video related to Shannon in some way (who says that one can’t incorporate modern media into a math course effectively?).

3. Please read the intro section of Hamming’s 1950 paper, “Error Detecting and Correcting Codes” up to Part I. Also, send me a fun fact about Hamming, but a brief search indicates that he was not nearly as fun as Shannon. He does have all of these good math quotes though. Feel free to send one of those. Basically, this is just an attempt to get you to look at some of the history, and to at least read their Wikipedia pages.

4. I hope to incorporate some computational aspects of coding theory into the course. The computer algebra system SAGE, which is free and open-source, has some pretty cool coding theory functions built in. Please install SAGE on your computer. It is available at http://www.sagemath.org/, and is very well-documented. If you have questions about installation you can probably find the answers in the documentation section. Also, there are definitely worse ways for you to spend your time then watching some of the introductory videos on the site.

If you still have problems, let me know and I can point you in the right direction. I will demonstrate some SAGE coding theory stuff next week.

The underlying language of SAGE is Python, but if you’ve never used SAGE or Python before, that’s ok. It might be helpful to read a little of the tutorial in the documentation section.

5. Recall that in the repetition code of length $n$, the probability that the decoder makes a mistake is

$$\sum_{0 \leq k < n/2} \binom{n}{k} (1-p)^k p^{n-k}.$$ 

Show that if $p < \frac{1}{2}$ this goes to 0 as $n \to \infty$. 
6. Let $C$ be a binary linear code such that the weight of each row of the generator matrix is even. Show that the weight of every codeword of $C$ is even.

7. This is probably the hardest exercise.
   Recall the extended Hamming code of length 8. If you don’t remember the generator matrix for this code, remember that the $[7, 4]$ Hamming code has parity check matrix with columns equal to all nonzero binary vectors of length 3. This should get you to the generator matrix, and we get the extended code by appending a column so that each row of the generator matrix has even weight. Note that this code corrects one error and detects two errors (but does not correct two errors).
   (a) Let $y \in \mathbb{F}_2^8$. Show that if the weight of $y$ is odd, then it is Hamming distance 1 away from a unique vector in the code.
   (b) Suppose that $y$ has even weight but is not in the extended Hamming code. What is the minimal distance from $y$ to a codeword?
   (c) Show that given any two positions chosen from $\{1, \ldots, 8\}$, there are exactly three words of the code of weight 4 which have 1s in these two positions.
   (d) To how many codewords does it have the minimal distance from part (b)?

8. This is probably the next hardest exercise.
   Suppose we decode the $[8, 4]$ extended Hamming code so that it corrects exactly one error, and if there are two errors it does not change the received word. (So, $(1, 1, 0, 0, 0, 0, 0, 0)$ which is not in the code is decoded as is, rather than guessing at which codeword was sent).
   What is the probability that we make an error after decoding?
   Give an upper bound for the expected number of errors after decoding? (Hint: How many new errors can we introduce by decoding incorrectly?)

9. Suppose we now consider the extended Hamming code of length $2^m$. The expected number of errors before decoding is $np$ where $p$ is the probability that any individual bit is sent incorrectly.
   Show that the expected number of errors after decoding is at most $(np)^2$.
   Hint: If there are exactly two errors in our received word we do not try to decode it. If there are more than two, how many extra errors can we introduce by decoding to the wrong codeword? (There is a slick way to do this. It uses the 291415139112 20851518513. (It is coding theory class, so if you want to read this semi-spoiler the numbers correspond to the letter in the alphabet, so $A = 1$, $B = 2$, etc.))

10. What is the weight distribution for the dual, $C^\perp$ of the $[7, 4]$ Hamming code? Check that in this case, $W_C(x, y) = \frac{1}{8}W_{C^\perp}(x + y, x - y)$.
    What do you think the weight distribution of the dual of the Hamming code of length $2^m - 1$ looks like in general?

11. (a) Let $C$ be a binary linear code of length $n$. Show that $z \in \mathbb{F}_2^n$ satisfies $z \cdot y = 0$ for all $y \in C^\perp$ if and only if $z \in C$. What is a better way to say this? (Remember, inner products are mod 2.)
    (b) Now let $C$ be a binary code that is not necessarily linear. Let $D$ be the set of all $y \in \mathbb{F}_2^n$ with $x \cdot y = 0$ for all $x \in C$. Is it true that $z \cdot y = 0$ for all $y \in D$ if and only if $y \in C$?
        (You may want to just stare at some nonlinear subsets of $\mathbb{F}_2^n$ for small $n$.)

12. (a) Fix $x \in \mathbb{F}_2^n$. How many vectors in $\mathbb{F}_2^n$ are Hamming distance exactly $k$ for $x$? How many vectors are at distance at most $k$?
    (b) Let $C$ be a linear code of length $n$ and minimal distance $d = 2e + 1$. Does the previous fact suggest an upper bound on $|C|$? Would it help if I told you this was called the ‘Sphere Packing Bound’?