Class 6 Exercises

CS250/EE387, Winter 2022

1. **(QR Codes I)** This 21×21 QR code is encoding a message with a fair amount of redundancy (level "Q" in QR-code lingo).

More precisely[1], the encoding process starts with a message (in this case, plain text) and interprets it as $k = 13$ elements of $\mathbb{F}_{256}$ (aka, as 13 bytes, where each byte encodes a unicode symbol). Then it encodes these k field elements using a RS code over $\mathbb{F}_{256}$ with $k = 13$ and $n = 26$. It takes the 26 codeword symbols (elements of $\mathbb{F}_{256}$), and treats them as bytes again. Then these bytes get written into the QR code in a specified order: the 8 bits that represent a given symbol in $\mathbb{F}_{256}$ get mapped to contiguous 4×2 blocks. A mask is applied over the top, and the remaining pixels are used for formatting information (what level of encoding, what type of message, those little squares to help your phone position it, what the mask is, etc).

   (a) What is the rate of the (256-ary) RS code used? What is the distance?

   (b) How many pixels in the QR code are auxiliary formatting information?

   (c) Explain how you can view this code (not including the auxiliary pixels or the mask) as a binary concatenated code. What is the rate of this binary code? What can you say easily[2] about the distance? (Bonus: can you figure out the actual distance?)

2. **(QR Codes II)** Building on the previous problem, let's do a thought experiment about a different way a QR code could work, based on concatenated codes.

   (a) Show that there is a binary linear code of dimension 7, length 8, and distance 2.

   Now, back to the QR codes, instead of choosing an RS code over $\mathbb{F}_{256}$, let $\mathcal{C}_{out}$ be an RS code with $k = 15$ and $n = 26$ over $\mathbb{F}_{128}$. Let $\mathcal{C}_{in}$ be a binary code from part (a). Now encode a QR code as outlined in the previous problem, but instead with the concatenated code $\mathcal{C} = \mathcal{C}_{out} \circ \mathcal{C}_{in}$.

   (b) What is the rate and designed distance of $\mathcal{C}$? How does it compare with the one in the previous problem?

   (c) Why do you think that QR codes don't actually work this way?

[1] This is my understanding of how QR codes work; take it as fact for this problem, but if you are inspired to read up on it and happen to be able to correct me, please let me know!

[2] We're looking for a statement like "The distance is at least x." The statement "the distance is at least 0" is too easy.

3. **(An application of BCH codes)** Consider the following problem (which doesn't have to do with coding theory).

   - A sender (Sally) sends a set $S \subseteq \{0,1\}^m$ of size N to a receiver (Rolanda). Think of the elements of S as being packet headers or something; Sally is going to send N packets to Rolanda, with packet headers in S. (Also, think of N as being pretty large, like $(2^m)/10$ or something like that.)
   - Unfortunately, some of the packets get lost on the way, and Rolanda receives a set $T \subseteq S$. Let's say that $|T| = N - x$, so x packets have gone missing. (Think of $x \ll N$ as being WAY smaller than N).
   - Rolanda would like to tell Sally which packets were lost, so that Sally can re-send them. The goal is to minimize the amount of communication from Rolanda back to Sally. Assume that Rolanda has a perfect communication channel back to Sally.
   - (Note: let's assume that each packet is really really big (much bigger than the m-bit header), so Sally really doesn't want to re-send any packet she doesn't need to; in particular "resending all N packets" is not a good solution).

   (a) Here are three simple schemes to use as a baseline.

      i. Rolanda sends all the headers in T back to Sally. Sally computes $S \setminus T$ and resends the missing packets.

      ii. Rolanda sends a binary vector $\mathbf{1}_T \in \{0,1\}^{2^m}$ back to Sally, to indicate which packets she received. Sally computes $S \setminus T$ and resends the missing packets.

   How much communication from Rolanda back to Sally, measured in bits, is required in these simple schemes? If $N \approx (2^m)/10$ and $x = O(1)$, what is this in terms of N, in big-Oh notation?

   (b) Now we'll work out a better scheme that uses BCH codes. Here's the outline of the scheme:

   - Let's assume that the all-zero string will never be in S. (Why is this an okay assumption?)
   - Sally sends $S \subseteq \{0,1\}^m$ to Rolanda, who receives $T \subset S$, as above.
   - Rolanda computes $H\mathbf{1}_T$, where $\mathbf{1}_T \in \{0,1\}^{2^m-1}$ is the indicator vector for the set $T \subseteq \{0,1\}^m \setminus \{\vec{0}\}$, and where H is the parity-check matrix for an appropriate BCH code (you will need to choose the parameters!)
   - Sally computes $v = H\mathbf{1}_S - H\mathbf{1}_T$.
   - Sally (somehow...) recovers the identities of the missing packets from v, and sends them to Rolanda.

   Answer the following:

      i. Fill in the details. What are the parameters of the BCH code you should choose? How does Sally figure out which packets Rolanda needs? Why is it okay to ignore the all-zero string (up to, say, one extra bit of communication)? And what is the final amount of communication from Rolanda back to Sally?

      ii. Argue that this amount of communication is essentially optimal (when, say, $N \approx (2^m)/10$ and $x = O(1)$).

4. **(Extra: RM Codes)** [Probably won't have time for this question...it's here in case you finish early and want some practice with RM codes too. This is not meant to be a tricky question, don't overthink it.] Recall from the lecture videos/notes that $RM_2(m,r)$ is binary Reed-Muller code formed by m-variate polynomials of total degree at most r. For what m do there exist binary Reed-Muller codes of rate exactly $1/2$? For those m, what should r be?