Mathematics 392 – Seminar in Coding Theory
Discussion 1 – Basic Notions of Coding Theory
January 23, 2002

Background

We have introduced the basic language of binary block codes (codes consisting of collections of binary words of some fixed length n). In particular, we have introduced the binary symmetric channel and analyzed error probabilities. We have also defined the *Hamming distance* between two words v, w to be $d(v, w) =$ the number of positions in which the words differ, and seen how the Hamming distance can be used to analyze the error-detection and error-correction capacity of a binary code. Today, we want to practice using these ideas, and work out a proof that the Hamming distance has the properties we used in our theorems on Monday.

Discussion Questions

A) Suppose that a binary symmetric channel with reliability $p = .999 = 1 - 10^{-4}$ can transmit 10^5 binary digits per second. We want to use this channel to transmit messages divided into words of length $n = 11$.

1) With no channel coding for error control, how often could we expect a word to be transmitted incorrectly (how many incorrect words per second)?
2) Now, suppose that a parity-check digit is added to all words (making them length $n = 12$), so that a single bit error can be detected. How often could be we expect a word to be transmitted incorrectly, but *not detected* (how many per second)? Indicate whether your answer is exact or an approximation and explain.
3) We will see in a few days that it is possible to encode words of length 11 by adding *four additional parity check digits* to each one (making the codewords length $n = 15$) in such a way that the resulting code has minimum distance 3. Hence by our results from Monday, a maximum likelihood decoder can correct any single bit error in a received word. For this coding scheme, no other errors can be corrected, though, and if any two or more bit errors occur in a received word, it will be decoded incorrectly. Assuming we have a decoding algorithm for this code (we'll see how to construct one later), how often could we expect a word to be received but *decoded incorrectly* (how many per second)?
4) Discuss the tradeoffs in efficiency and reliability of communication between scenarios 1,2,3.

B) Show that the Hamming distance has the following properties.

1) For all v, w, $d(v, w) \geq 0$ and $d(v, w) = 0$ only when $v = w$.
2) For all v, w, $d(v, w) = d(w, v)$.
3) For all u, v, w, $d(u, w) \leq d(u, v) + d(v, w)$. (This is a triangle inequality.)

C) Determine the *minimum distance* of the following codes.

1) $C = \{1010, 1110, 0011\}$
2) $C = \{000000, 101010, 010101, 111111\}$
3) The code obtained by adding a parity check digit to K^n, any n.

D) Error control in the form of check digits is also often used for messages over non-binary alphabets. A possibly familiar case is the ISBN numbers used by publishers to identify books. For instance the ISBN number of the HHLLPRW text we are using is 0824704657 (see the barcode on the back cover – the dashes are omitted since they are not part of the actual code). All ISBN's have the form $d_1 d_2 \cdots d_9; d_{10}$ where the first 9 digits are ordinary decimal digits 0 - 9 and are the actual identifier for the book. The last digit d_{10} is a check digit that is added to make the number $d_1 + 2d_2 + 3d_3 + \cdots + 9d_9 + 10d_{10}$ evenly divisible by 11. For example, the ISBN for our text has:

$$1 \cdot 0 + 2 \cdot 8 + 3 \cdot 2 + 4 \cdot 4 + 5 \cdot 7 + 6 \cdot 0 + 7 \cdot 4 + 8 \cdot 6 + 9 \cdot 5 + 10 \cdot 7 = 264 = 24 \cdot 11.$$

(In cases where the check digit must be $d_{10} = 10$, the symbol X is used instead to keep the total length to 10 symbols.)

1) Explain why, given any nine digits $d_1, \ldots, d_9$, there is always an integer $d_{10} \in \{0, 1, 2 \ldots, 10\}$ such that $d_1 + 2d_2 + 3d_3 + \cdots + 9d_9 + 10d_{10}$ is divisible by 11. Also show the check digit serves to *detect* any error in a single digit of an ISBN.
2) Probably the most common error involving more than one digit made by humans copying codes like ISBN's is to *transpose* or interchange two symbols. Show that the ISBN code can also be used to *detect any single transposition error*. Also show by example that correcting a transposition error is not always possible.

Extra Credit Challenge Question

In addition to the situation we discussed in class where an error in a received word changes a 0 to a 1 or a 1 to a 0, there are also instances in which a digit in a received word is simply not recognizable as either a 0 or a 1. In this case we say that an *erasure* has occurred in that location. Show that if a code has minimum distance d and any combination of t errors (not erasures) and e erasures with $2t + e < d$ errors occurs in a received word, then there is a unique closest codeword. Hence maximum likelihood decoding will correct the errors and erasures. Which are *easier to correct* – errors or erasures? Why?

Assignment

Group write-ups due in class Wednesday, January 30.