ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE
School of Computer and Communication Sciences

Handout 25 Information Theory and Coding
Solutions to homework 10 December 18, 2007

Problem 1.

(a) Since the code corrects up to (and including) e errors, the decoder must decide on x for all sequences y that differ from x in at most e positions. Since the number of sequences that differ from x in i places is $\binom{n}{i}$, we see that the number of such y is

$$\sum_{i=0}^{e} \binom{n}{i}.$$

We thus conclude that $|S| \geq \sum_{i=0}^{e} \binom{n}{i}$. Note that there might be other sequences y for which the decoder decides on x. This is the reason for the inequality rather than equality. [For example, take $n = 4$ and $e = 1$. There are a total of $2^4 = 16$ sequences of length n. $\sum_{i=0}^{1} \binom{4}{i} = 5$. However, 5 does not divide 16, so there must be sequences that cannot be accounted for by just considering sequences that differ from the codewords by at most 1 position.]

(b) Let S_x be the set of sequences for which the decoder decides x. Since these sets are disjoint (for a given y, the decoder has to know which x to declare), the union of these sets contains at least

$$M \sum_{i=0}^{e} \binom{n}{i}$$

sequences. But the total number of sequences of length n is 2^n, thus

$$M \leq 2^n \Big/ \sum_{i=0}^{e} \binom{n}{i}.$$

(c) For the Hamming codes, $n = 2^m - 1$, $M = 2^{2^m - m - 1}$, $e = 1$. Thus,

$$\sum_{i=0}^{e} \binom{n}{i} = 1 + n = 2^m$$

and $2^n / \sum_{i=0}^{e} \binom{n}{i} = 2^{2^m - 1} / 2^m = 2^{2^m - m - 1}$. We thus see that the bound in part (a) is met with equality.

Problem 2.

(a) This is more or less obvious: any codeword chosen differs from the previously chosen codewords in more than d places since those sequences that do not were eliminated in the previous steps. In fact, even though the problem claims that the minimum distance is at least d, procedure yields a minimum distance of at least $d + 1$.

(b) It is clear that at each iteration one codeword is added to the list of chosen codewords. The sequences removed from the candidate list are those that differ from the chosen sequence in less than or equal to d places. Since some of these sequences might have been removed in the previous iterations, the number of sequences that differ from the codeword in d or less places is an upper bound on the number of sequences removed. But this is

$$\sum_{i=0}^{d} \binom{n}{i}.$$

When the algorithm terminates all 2^n sequences are removed from the candidate list. Thus the number of iterations the algorithm makes is at least $2^n / \sum_{i=0}^{d} \binom{n}{i}$. Since the number of codewords chosen is equal to the iterations of the algorithm,

$$M \geq 2^n \Bigg/ \sum_{i=0}^{d} \binom{n}{i}.$$

Problem 3. Let x and x' be two different codewords in the extended Hamming code. Let z and z' be the parts of x and x' that come from the Hamming code (i.e., z is all but the last bit of x, and z' that of x'), and p and p' be the bits appended to z and z' to get x and x'. Since x and x' are different then so are z and z': if $z = z'$ then $p = p'$ and x and x' would have been the same. Thus, $d_H(z, z') \geq 3$ since z and z' are different Hamming codewords. On the other hand, if $d_H(z, z') = 3$, then z and z' must have different parity: if they both had an even number of 1's or both had an odd number of 1's they would have differed in an even number of places and $d_H(z, z')$ would have been an even number. Thus, if $d_H(z, z') = 3$ then $p \neq p'$ and we have $d_H(x, x') = 4$. If $d_H(z, z') \geq 4$ then clearly $d_H(x, x') \geq 4$. We thus see that the miminum distance of the new code is 4.

Consider the following procedure to decode

> Given a sequence y, compare it to all the codewords and find the number of positions in which y differs from them. If there is a unique codeword for which this number is smallest, declare that codeword. If not, declare 'errors were detected'.

If the minimum distance d of a code is an even number, $d = 2j$, then if a sequence y differs from the transmitted codeword x by up to $j - 1$ places, then y will be close to the transmitted codeword than to any other and the decoder will correctly decode x. If however, y differs from x in j places, then no other codeword will be closer to y, but there might be a codeword x' which also differs from y in j places. In such a case the decoder will not be able to correct but detect the errors. In particular, if $d = 4$, then all single errors are corrected and all double errors are detected (may even be corrected).

Problem 4.

(a) Since C is non-empty, it contains some codeword x. By linearity C must contain $x + x$. But, for any x, $x + x$ is the all zero sequence since we are doing modulo-2 sums. So, C contains the all zero sequence.

(b) The elements of D' are those sequences of the form $x + y$ where y is in D. Since x is in C and D is a subset of C, any x and y are both in C, and so is their sum.

(c) Suppose there was an element z common to D and D'. Then $z = x + y$ where y is in D. Since we assumed that D is a linear subset, then $z + y$ is also in D. But $z + y$ equals x, and we arrive at the contradiction that x is in D.

(d) Since the mapping $y \mapsto x+y$ is a bijection, D and D' are in one-to-one correspondence, and hence have the same number of elements.

(e) Suppose z_1 and z_2 are in $D \cup D'$. There are four possibilities: (1) both z_1 and z_2 are in D, (2) both z_1 and z_2 are in D', (3) z_1 is in D, z_2 is in D', (4) z_1 is in D', z_2 is in D. In case (1), the linearity of D implies that $z_1 + z_2$ is in D. In case (2), $z_1 = x + y_1$ and $z_2 = x + y_2$ for some y_1 and y_2 both in D, then $z_1 + z_2 = x + x + y_1 + y_2 = y_1 + y_2$ is in D. In case (3) $z_2 = x + y_2$ and $z_1 + z_2 = x + (z_1 + y_2)$, which is in D', and similarly in case (4). Thus in all cases $z_1 + z_2$ is in $D \cup D'$ and we see that $D \cup D'$ is a linear subset of C.

(f) We thus see that if at the beginning of step (ii) D is a linear subset of C, at the end of step (iii) $D \cup D'$ is linear, a subset of C because both D and D' are, and has twice as many elements of D since D' has the same number of elements of D and is disjoint from it. Thus, when the algorithm terminates, D contains all elements of C and since it is a subset of C it must equal C. Furthermore, its size, being equal to successive doublings of 1, is a power of 2.

PROBLEM 5.

(a) Note first that the sum of two even weight codewords is of even weight, the sum of two odd weight codewords is of even weight and the sum of an odd weight codeword with an even weight codeword is of odd weight.

If the code contains no odd weight codeword then we are done. Otherwise let x be an odd weight codeword. Then the mapping $y \mapsto x+y$ is a bijection between even weight and odd weight codewords, and we conclude that there must be an equal number of odd and even weight codewords.

(b) The same proof above applies: either all codewords have a zero at the nth digit, or there is a codeword x with has a 1 in its nth digit. The mapping $y \mapsto x + y$ gives a bijection between codewords who have a zero at the nth digit and codewords which have a 1 at the nth digit. In the first case, when all codewords have a zero at the nth digit, one can improve the code by simply deleting the nth digit from each codeword: no matter what the message, the same symbol would have been transmitted, giving no additional information.

(c) To find the average number 1's per codewords, one would find the total number of 1s in all codewords, and divide this sum by the number of codewords. Suppose there are M codewords. Arrange the codewords in rows, and count the total number of 1's by going over columns one by one. Since each column contains at most $M/2$ ones, and there are N columns, the total number of 1's is less than or equal to $MN/2$. Dividing by M we see that the average number of 1's per codeword is at most $N/2$.

PROBLEM 6. Let S_0 be the set of codewords at Hamming distance n from $\mathbf{x}_0$ and S_1 be the set of codewords at Hamming distance n from $\mathbf{x}_1$. For each $\mathbf{y}$ in S_0, note that $\mathbf{x}_1 + \mathbf{y}$ is at distance n from $\mathbf{x}_1$, and thus $\{\mathbf{x}_1 + \mathbf{y} : \mathbf{y} \in S_0\} \subset S_1$. Similarly, $\{\mathbf{x}_1 + \mathbf{y} : \mathbf{y} \in S_1\} \subset S_0$. These two relationships yield $|S_0| \leq |S_1|$ and $|S_1| \leq |S_0$, leading to the conclusion that $|S_0| = |S_1|$.