IDEAS OF HAMMING CODE

CHENYAN WU

Problem: Want to transmit n bits, which we call data bits. But the transmission process can corrupt the bits. How to detect if there is corruption.

Assumption: The transmission process corrupts at most 1 bit, also called bit flip, i.e., changing 0 to 1 or 1 to 0.

Solve the problem under this assumption and then consider worse situations later (not included in the current notes). In any case, we cannot allow too much corruption. Sometimes you just have to improve the medium of transmission first.

Idea: Parity check.
Transmit $n+1$ bits. The n data bits and 1 more bit such that the sum of the all transmitted bits is 0 modulo 2.

Example 0.1. Data bits: 11011. Parity check bit: 0. Should transmit: 110110.

When the receiver receives the bits, add them up modulo 2. If the result is 0, then no corruption. If the result is 1, then there is corruption and ask the sender to send again and pray the next time there is no corruption.

Example 0.2. Transmitted: 110110.
Received: 110110. Sum: 0. No corruption.
Received: 110010. Sum: 1. Has corruption.
Received: 110111. Sum: 1. Has corruption. (It is possible that the bit flip happens for the parity check bit.)
Received: 110000. Sum: 0. Not possible under the at most 1-bit flip assumption.
In this example, Sum is taken modulo 2.

New problem: Too passive! The receiver wants to be able to detect if there is corruption and, in addition, wants to determine which bit is flipped and thus will be able to correct it.

Idea: Put in more parity checks and the combo of failed parity checks somehow should pinpoint where the corruption is.

Estimate how many parity checks are needed: Transmit $n+r$ bits: n bits are the data bits and r bits for parity checks.

Total $n+r+1$ outcomes: one of the $n+r$ bits is corrupted or nothing is corrupted. The r bits for parity checks can represent 2^r numbers. Thus given n, we choose r such that $2^r \geq n+r+1$.

Need a dictionary from combo of failed checks to position of corrupted bit: (If all parity checks pass, then there is no corruption.)

Label the positions of all total bits: $1, 2, 3, \ldots, n+r$. Set $P := \{1, 2, 3, \ldots, n+r\}$. Let C be the subset of P corresponding to the positions of the parity check bits. We will choose C appropriately later. Let v_i denote the value of the i-th bit for $i \in P$.

For $j \in C$, we want to associate a subset P_j of P such that $j \in P_j$. We set v_j to be the number such that $\sum_{i \in P_j} v_i = 0$ modulo 2, i.e., the j-th parity check checks the bits at positions in P_j.

When the receiver receives the bits, they find which parity checks fail, in other words, they find the subset C' of C such that for $j \in C'$, $\sum_{i \in P_j} v_i \neq 0$ modulo 2 and for $j \in C \setminus C'$, $\sum_{i \in P_j} v_i = 0$ modulo 2. We want to choose P_j's appropriately so that only the flip of one particular bit can result in having C' as the set of failed parity checks.

How to choose P_j's:

There is an ingenious way devised by Hamming for the choice of C and P_j for $j \in C$. Set

$$C := \{1, 10, 100, \ldots\}$$

where the numbers are in radix 2. Set

$$P_j := \{i \in P \mid i \,\text{AND}\, j \neq 0\} = \{\text{number of the form } * \cdots * 1 * \cdots *, \text{ with the 1 at location } j\}$$

where AND is the bitwise AND operator the numbers are in radix 2.

Then if C' is the set of failed parity checks, then $\sum_{a \in C'} a$ is the position of the bit that is corrupted. (If the sum is 0, this means that there is no corruption.)

Example 0.3. Data bits: 1101.
It suffices to have 3 parity check bits.
Now all numbers below are in radix 2.

Position of the parity check bits: $001, 010, 100$. ($C = \{001, 010, 100\}$.)

$$P_{001} = \{**1\} = \{001, 011, 101, 111\}$$
$$P_{010} = \{*1*\} = \{010, 011, 110, 111\}$$
$$P_{100} = \{1**\} = \{100, 101, 110, 111\}.$$

Position	001	010	011	100	101	110	111
Value of data bits	_	_	1	_	1	0	1
Value of all bits (Sender sends)	1	0	1	0	1	0	1
Value of all bits (Receiver sees)	1	0	1	0	1	1	1
P_{001}	•		•		•		•
P_{010}		•	•			•	•
P_{100}				•	•	•	•

Then the sender sends 1010101. Assume the receiver receives 1010111. From an all-seeing point of view, we see that the bit at position 110 is flipped.

The receiver checks that the bits with positions in P_{001} sum to 0 modulo 2; the bits with positions in P_{010} sum to 1 modulo 2; the with positions in P_{001} sum to 1 modulo 2. The three observations mean that the position of the bit that gets flipped is not of the form $**1$; it is of the form $*1*$; it is of the form $1**$. Thus it must be at position 110.