Error Detection and Correction: Parity Check Code; Bounds Based on Hamming Distance

Greg Plaxton
Theory in Programming Practice, Spring 2005
Department of Computer Science
University of Texas at Austin

Error Detection: A Simple Example

- Suppose bits are occasionally "flipped" in transmission, e.g., the message 1110001 gets corrupted to 0110011 (two bit flips)
- By using a code with sufficient redundancy, we can hope to detect/correct such errors, assuming there aren't too many of them
- For example, suppose we just repeat each bit twice
  - If the receiver gets xx, it assumes the bit is x
  - If the receiver gets two different bits, it requests retransmission
- The above is an example of an error detecting code (that can detect one error)
- The code is not considered to be error correcting because retransmission is necessary

Error Correction: A Simple Example

- Suppose the sender codes each bit x as xxx
- Claim: The receiver can now *correct* a single error
- How?
- How many errors can be detected?

Parity Check Code

- Commonly used technique for detecting a single flip
- Define the *parity* of a bit string w as the parity (even or odd) of the number of 1's in the binary representation of w
- Assume a fixed block size of k
- A block w is encoded as wa where the value of the "parity bit" a is chosen so that wa has even parity
  - Example: If $w = 10101$, we send 101011
- If there are an even number of flips in transmission, the receiver gets a bit string with even parity
- If there are an odd number of flips in transmission, the receiver gets a bit string with odd parity

Parity Check Code: Decoding

- If the receiver gets a bit string wa with even parity, it *assumes* that there were zero flips in transmission and outputs w
  - Note that the receiver fails to decode properly if the (even) number of flips is nonzero
- If the receiver gets a bit string wa with odd parity, it *knows* that there were an odd (and hence nonzero) number of flips, so it requests retransmission
  - The receiver never makes a mistake in this case
  - Still, it is a bad case because no progress is being made
- Underlying assumption: Flips are rare, so we can tolerate the corruption of the extremely small fraction of blocks with a nonzero even number of flips

Parity Check Code: Analysis of a Simple Example

- Note that the bit-duplicating code (where bit a is transmitted as aa) we discussed earlier is a parity check code
- Suppose we are using this code in an environment where each bit transmitted is independently flipped with probability 10^{-6}
- Without the code, one bit in a million is corrupted
  - We use one bit to encode each bit
- With the code, only about one bit in a trillion is corrupted
  - The retransmission rate is negligible, so on average we use slightly over each bits to encode each bit

Two-Dimensional Parity Check Code

- Generalization of the simple parity check code just presented
- Assume each block of data to be encoded consists of mn bits
- View these bits as being arranged in an $m \times n$ array (in row-major order, say)
- Compute $m + n + 1$ parity bits
  - One for each row, one for each column, and one for the whole message
- Send $mn + m + n + 1$ bits (in some fixed order)
- How many errors can be detected?

Hamming-Distance-Based Bounds on Error Correction and Detection

- Assume we would like to encode each symbol in a given set by a distinct codeword, where all codewords have the same length k
  - For a given k, and some desired level of error correction or detection, how large a set of symbols can we support?
  - It is also interesting to consider variable-sized codewords, but we will restrict our attention to the simpler scenario of fixed-size codewords
- Theorem: Let S be a set of codewords and let h be the minimum Hamming distance between any two codewords in S. Then it is possible to detect any number of errors less than h and to correct any number of errors less than $h/2$

Error Detection Bound

- Let S be a set of codewords and let h be the minimum Hamming distance between any two codewords in S
- Why are we guaranteed to detect any number of errors less than h?
- Is there guaranteed to be a case in which we are unable to detect h errors?

Error Correction Bound

- Let S be a set of codewords and let h be the minimum Hamming distance between any two codewords in S
- Why are we guaranteed to be able to correct any number of errors less than $h/2$?
- Is there guaranteed to be a case in which we are unable to correct $\lceil h/2 \rceil$ errors?