Error Detection and Correction Codes

Dr. Chandan Karfa

CSE, IIT Guwahati

- Source:

  Chapter 1 of Z. Kohavi and N. Jha, Switching and Finite Automata Theory, 3rd Ed., Cambridge University Press, 2010

Error detection and correction

- Transmission errors may occur because of equipment failure or noise in the transmission channel.
- In any practical system there is always a finite probability of occurrence of a single error.
- **Error-detecting codes**
  If a code possesses the property that the occurrence of any single error transforms a valid code word into an invalid code word, it is said to be a (single-)error-detecting code.

- **Error-correcting codes**
  In general, a code is said to be error-correcting if the correct code word can always be deduced from the erroneous word.

Error-detecting codes

- A code is an error-detecting code if and only if its minimum distance is two or more.
- If a single error occurs it transforms the valid code word into an invalid one, thus making the detection of the error straightforward.
- In general, to obtain an n-bit error-detecting code, no more than half the possible 2^n combinations of digits can be used. The code words are chosen in such a manner that, in order to change one valid code word into another valid code word, at least two digits must be complemented.

Cont'd..

- Some Error Detecting code such as Parity Check and 2-out-of-5 code are generally used.

Decimal	Even-parity BCD					2-out-of-5				
digit	8	4	2	1	*p*	0	1	2	4	7
0	0	0	0	0	0	0	0	0	1	1
1	0	0	0	1	1	1	1	0	0	0
2	0	0	1	0	1	1	0	1	0	0
3	0	0	1	1	0	0	1	1	0	0
4	0	1	0	0	1	1	0	0	1	0
5	0	1	0	1	0	0	1	0	1	0
6	0	1	1	0	0	0	0	1	1	0
7	0	1	1	1	1	1	0	0	0	1
8	1	0	0	0	1	0	1	0	0	1
9	1	0	0	1	0	0	0	1	0	1

Parity check and 2-out-of-5 code

- The basic idea in a parity check is to add an extra digit to each code word of a given code so as to make the number of 1's in each code word either odd or even.
- The added bit, denoted p, is called the parity bit.
- The 2-out-of-5 code consists of all 10 possible combinations of two 1's in a five-bit code word.
- The 2-out-of-5 code is a weighted code and can be derived from the (1, 2, 4, 7) code (except 0)

Error-correcting codes

- For a code to be error-correcting, its minimum distance must be further increased.
- If a code possesses the property that the occurrence of any single error transforms a valid code word into an invalid code word, it is said to be a (single-)error-detecting code.
- If the minimum distance of a code is *three*, then any single error changes a valid code word into an invalid one, which is distance *one away from the original code* word and *distance two from any other valid code word*.
- Similarly, a code whose minimum distance is four maybe used for either single-error correction and double-error detection or triple-error detection.

Hamming Codes for error correcting

- The basic principles in constructing a Hamming error-correcting code are as follows.
  - To each group of m information or message digits, k parity-checking digits, denoted p1 , p2 , . . . , pk , are added to form an (m + k)-digit code.
  - The location of each of the m + k digits within a code word is assigned a decimal value; one starts by assigning a 1 to the most significant digit and m+k to the least significant digit.
  - Then k parity checks are performed on selected digits of each code word. The result of each parity check is recorded as 1 or 0, depending, respectively, on whether an error has or has not been detected.
  - If all parity checks indicate even parities then the message is correct.

Hamming Codes

- The number k of digits in the position number must be large enough to describe the location of any of the $m + k$ possible single errors, and must in addition take on the value zero to describe the “no error” condition.
- Consequently, k must satisfy the inequality $2^k \geq m + k + 1$.
  - Original message is in BCD where $m = 4$ then $k = 3$
  - At least three parity checking digits must be added to the BCD code.
- In order to be able to specify the checking digits by means of only message digits and independently of each other, they are placed in positions 1, 2, 4, . . . , 2^{k-1}.
  - BCD: the parity bits are in positions 1, 2, and 4 while the remaining positions contain the original (BCD)

Hamming Codes

- Position number: A binary number, $c1c2 \cdots ck$, whose value is equal to the decimal value assigned to the location of the erroneous digit when an error occurs and is equal to zero if no error occurs

Table 1.7 Position numbers $c_1c_2c_3$

	Position number		
Error position	c_1	c_2	c_3
0 (no error)	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Hamming Code Construction

- If an error occurs in position 1, or 3, or 5, or 7, the least significant digit, i.e., $c3$, of the position number must be equal to 1.
- If the code is constructed so that in every code word the digits in positions 1, 3, 5, and 7 have even parity, then the occurrence of a single error in any of these positions will cause an odd parity.

p1 is selected so as to establish even parity in positions 1, 3, 5, 7;
p2 is selected so as to establish even parity in positions 2, 3, 6, 7;
p3 is selected so as to establish even parity in positions 4, 5, 6, 7.

Hamming Code Construction

- Consider, for example, the message 0100 (i.e., decimal 4)

Digit position:	1	2	3	4	5	6	7
Digit symbol:	p_1	p_2	m_1	p_3	m_2	m_3	m_4
Original BCD message:			0		1	0	0
Parity check in positions 1, 3, 5, 7 requires $p_1 = 1$:	**1**		**0**		**1**	0	**0**
Parity check in positions 2, 3, 6, 7 requires $p_2 = 0$:	1	**0**	**0**		1	**0**	**0**
Parity check in positions 4, 5, 6, 7 requires $p_3 = 1$:	1	0	0	**1**	**1**	**0**	**0**
Coded message:	1	0	0	1	1	0	0

Hamming Code

- Hamming code for decimal digits(0 to 9)

Decimal digit	Digit position and symbol						
	1 p_1	2 p_2	3 m_1	4 p_3	5 m_2	6 m_3	7 m_4
0	0	0	0	0	0	0	0
1	1	1	0	1	0	0	1
2	0	1	0	1	0	1	0
3	1	0	0	0	0	1	1
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	1
6	1	1	0	0	1	1	0
7	0	0	0	1	1	1	1
8	1	1	1	0	0	0	0
9	0	0	1	1	0	0	1

Error Correction in Hamming Code

- Perform Parity check 1-3-5-7 to get c3
- Perform parity check 2-3-6-7 to get c2
- Perform parity check 4-5-6-7 to get c1
- <c1 c2 c3> identifies the error location in the code.
- To correct the error, the digit in position <c1c2c3> is complemented

- Transmitted code: 1101001
- Received code : 1101101

Error Correction in Hamming Code

Digit position:	1	2	3	4	5	6	7	
Message received:	1	1	0	1	1	0	1	
4-5-6-7 parity check:				**1**	**1**	**0**	**1**	$c_1 = 1$ since parity is odd
2-3-6-7 parity check:		**1**	**0**			**0**	**1**	$c_2 = 0$ since parity is even
1-3-5-7 parity check:	**1**		**0**		**1**		**1**	$c_3 = 1$ since parity is odd

- <c1c2c3> = <101> = 5
- So, the error occurs in 5th bit
- To correct the error, the digit in position 5 is complemented and the correct message 1101001 is obtained

Correctness

Theorem: the Hamming code constructed is a code whose distance is **three.**

Proof:

- Since each message digit appears in at least two parity checks, the parity checks that involve the digit in which the two code words differ will result in different parities.
- Hence different checking digits will be added to the two words making the distance between them equal to three.

- The two words differ in only *m*1 (i.e., position 3). Parity checks 1-3-5-7 and 2-3-6-7 for these two words will give different results. Therefore, the parity checking digits *p*1 and *p*2 must be different for these words.

Digit position:	1	2	3	4	5	6	7
Digit symbol:	p_1	p_2	m_1	p_3	m_2	m_3	m_4
First word:			1		0	0	1
Second word:			0		0	0	1
First word with parity bits:	**0**	**0**	1		0	0	1
Second word with parity bits:	**1**	**1**	0		0	0	1