Mathematics in Compact Storage Devices

Coding Theory applied to CDs and DVDs

References

- [1] www.engineersgarage.com/mygarage/how-cd-works
- [2] J.P. Nguyen, *Applications of Reed-Solomon Codes on Optical Media Storage*, Master's Thesis (Mathematics), San Diego State University, Fall 2011.
- [3] LexInnova, Holographic Versatile Disc, March 2014 (available on the web)
- [4] ECMA Standard ECMA-377, Information Interchange on Holographic Versatile Disc (HVD) Recordable Cartridges – Capacity: 200 Gbytes per Cartridge, May 2007.

Acronyms

- CD: Compact Disc
- DVD: Digital Versatile Disc
- Blu-ray: The name refers to the laser used to read the disc
  - The reader uses a shorter wavelength blue laser as opposed to the red laser used for CDs and DVDs
- HVD: Holographic Versatile Disc
  - Researched for years but is currently not mass produced for consumers
    - Was initially envisioned for high-end uses as the device to read such discs was expected to cost in the tens of thousands of dollars

Introduction

- In modern day mass storage devices such as CDs and DVDs the data is protected from read errors by applying error correction codes, specifically Reed-Solomon Codes
  - These codes are a clever way of adding overhead (extra data) to the information of interest in such a way to allow errors to be corrected up to a certain extent
  - The price you pay for doing this is more (non-information) data has to be stored which takes up extra space but it is well worth paying this price to obtain high reliability when listening to CDs or watching DVDs

Introduction (cont.)

- In these slides we will present the overall concept of coding theory in CDs and DVDs without getting into specific details of the actual encoding/decoding algorithms
  - The codes which we will illustrate are
    - a simple Repeat Code (which is not really a mathematical code at all) and
    - A Hamming code (the first mathematical code)
    - Neither of these codes are used on CDs/DVDs (much more sophisticated Reed-Solomon codes are used) but nevertheless illustrates the basic principle that all error correction codes exploit

- Error correction coding is also widely applicable to other areas such as digital communication systems

CD/DVD Overview

- From [1]: "The surface of a CD is made of a polycarbonate layer with molded tracks on the top. The data are stored on the CD as a series of minute groves which are known as 'pits' encoded on these spiral tracks. The areas between the 'pits' are known as 'lands.' These pits and lands do not represent the 1s and 0s, rather each change from pit to land or land to pit is interpreted as 0 while no change is read as 1. The burning process of a CD is nothing more than creating a pattern of pits and lands over the polycarbonate layer."

- Note that 'lands' is also referred to as 'bumps' in the literature.
- A similar description as above applies to DVDs and Blu-rays (but they compact the data closer together for more storage)

Typical Information Storage Capacity

CD*	DVD*	Blu-ray*	HVD**
737 MB	4.7 GB	25 GB	> 1 TB
1.2 Mbits/sec	11.1 Mbits/sec	36.0 Mbits/sec	8.8 Gbits/sec

* Single sided, Single layer

** The HVD standard [4] specifies a very powerful error correction scheme consisting of a Reed-Solomon code combined with a Low Density Parity Check (LDPC) code

Notes:

1. The storage capacity given in the table indicate information storage capacity and does not count the extra data stored due to the error correction coding applied.

2. The transfer rates quoted are 1x raw rates including the overhead data. The actual information transfer rates are a little less. Blu-ray movies operate at 2x minimum. CDs/DVDs/Blu-rays can operate at several multiples of this 1x speed.

3. For HVD > 1 TB is expected information capacity [3] but the standard in [4] is for 200 GB.

CD/DVD Error Sources

- From [2] we have
  - "Channel errors can come from several areas. They can come from foreign particles, air bubbles in the plastic material, inaccuracies in the pits due to stamper errors [17] or in the coating and cutting of the disc [8]. These will cause errors when the information is optically read out. These commonly occur as random errors and smaller length error patterns.
  - However, burst errors caused by fingerprints and scratches on the surface of the disc result in errors affecting several consecutive bits [17]."
    - These consecutive types of bit errors of why interleaving is important so that the errors are spread to several different codewords so that any particular codeword is not overwhelmed with errors.

Error Correction Coding in CDs/DVDs

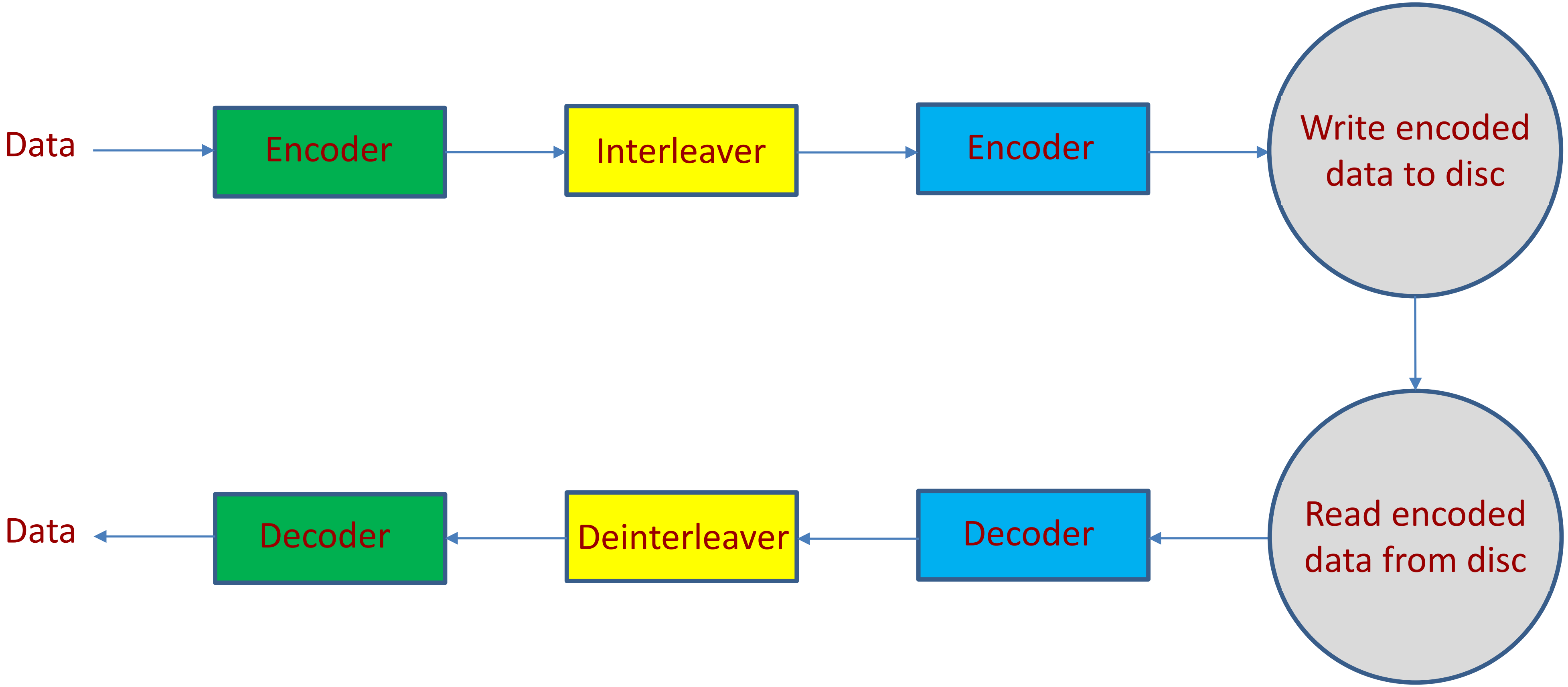
- At the basic level all data on a CD or DVD is represented logically as 0s and 1s
- Your CD/DVD player reads these 0s and 1s from the disc and translates them to their original meaning suitable for the human observer
- If the data becomes corrupted then the data may be misread (a 0 mistaken for a 1, or vice-versa)
  - In this case the powerful error correction coding used in storing the data allows for the errors to be perfectly corrected in many cases since the corrupted codeword is still closer to the correct codeword than it is to other incorrect codewords

What is Coding?

- By coding we do NOT mean
  - Source coding where the data is compressed in a lossless coding scheme (to remove redundancy)
  - Encryption where the data is mapped in a certain way in order to disguise it from an unauthorized receiver

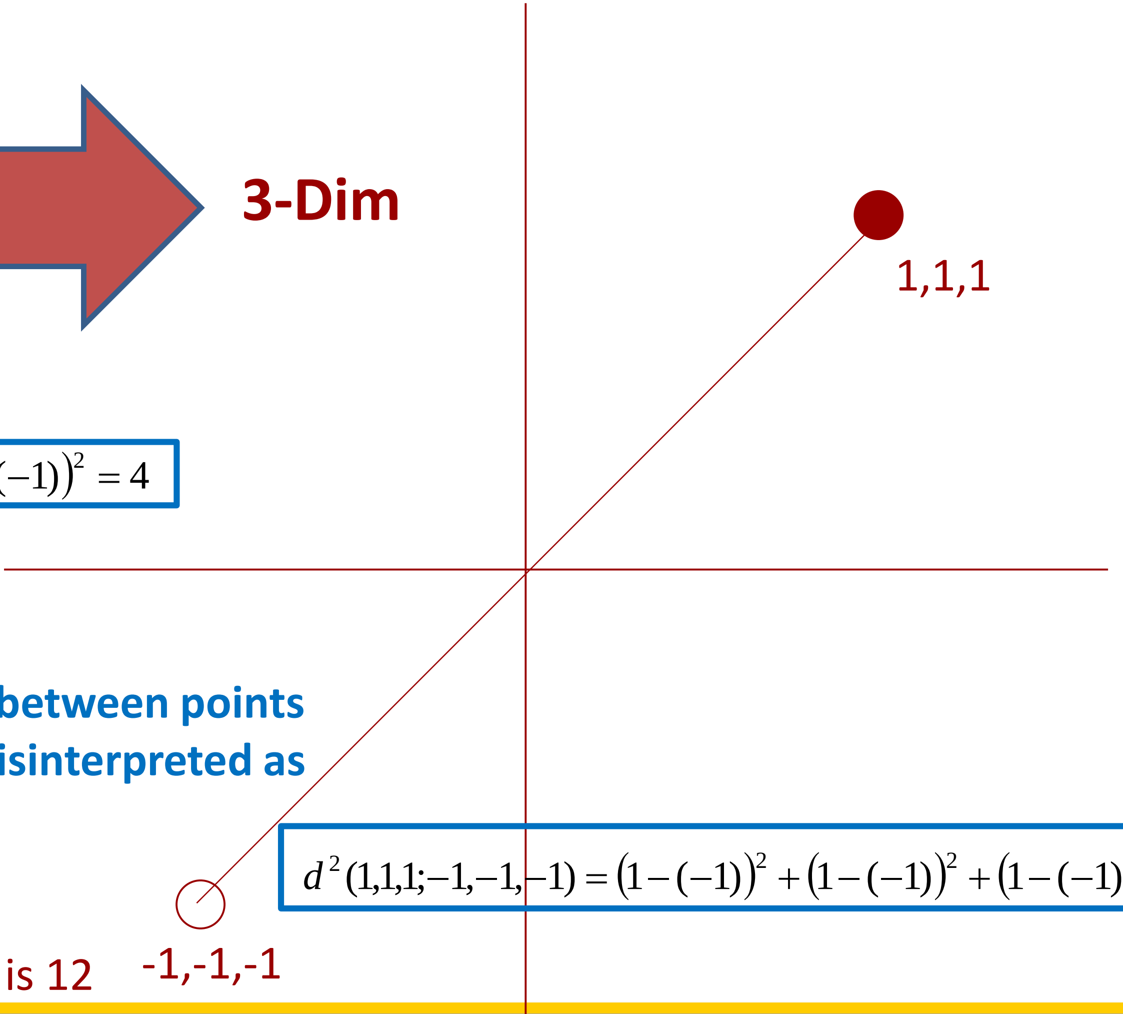
- By coding we do mean to add extra data to an information stream so that if that if the resulting code word becomes corrupted we can recover the original data with a reasonable probability

Block Diagram of CD/DVD Writing and Reading



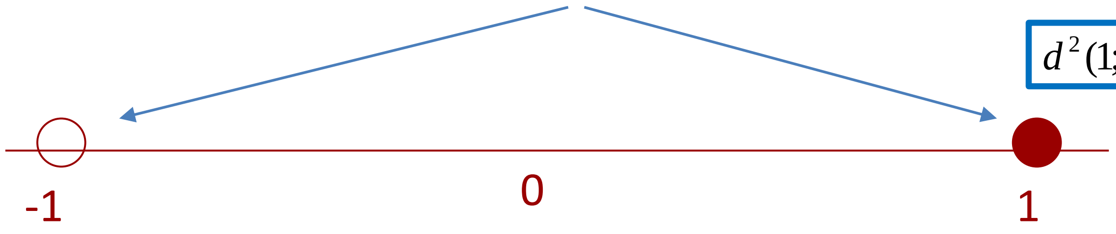
Simple Repeat Code Mapping Example

1-Dim → **3-Dim**



Data to be sent is 1 or -1 (logically a 1 or 0 on a computer)

$$d^2(1;-1) = \left(1-(-1)\right)^2 = 4$$



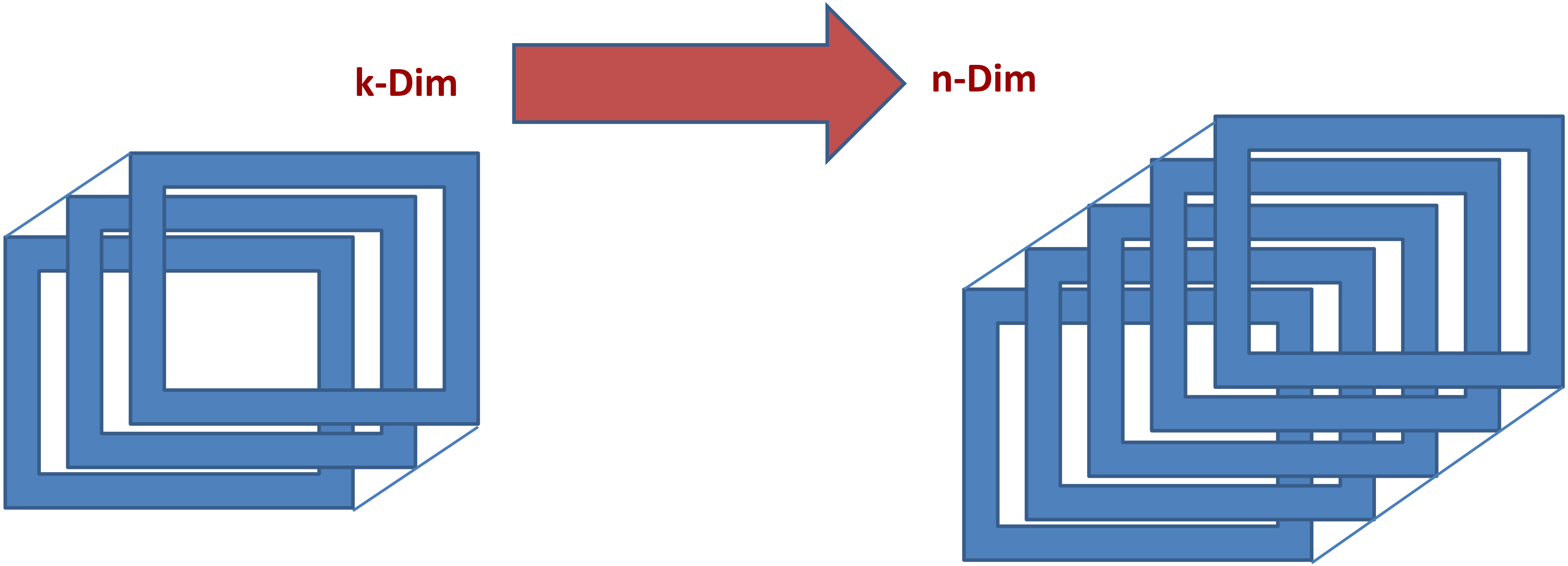
The performance of the code is related to the squared distance between points (the greater the distance the more likely the point will not be misinterpreted as another point due to corruption)

$$d^2(1,1,1;-1,-1,-1) = \left(1-(-1)\right)^2 + \left(1-(-1)\right)^2 + \left(1-(-1)\right)^2 = 12$$

The squared distance between data points in 1-Dim is 4

The squared distance between data points after coding in 3-Dim is 12

More General Encoding Mapping Pictorial



$k =$ number of information bits

$2^k =$ number of possible codewords

$n =$ size of codeword

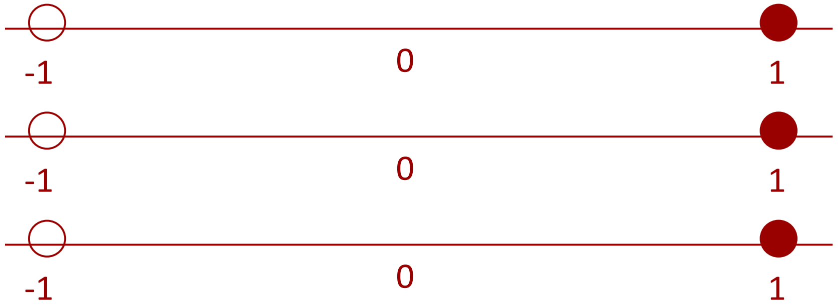
$n - k =$ number of parity bits (overhead)

$2^n =$ number of possible vectors received (or read)

2^n is much greater than 2^k so the legitimate codewords can look quite different from each other since there can be so much "room" or distance between them if the encoding (mapping) is done right

Example of Repeat Code

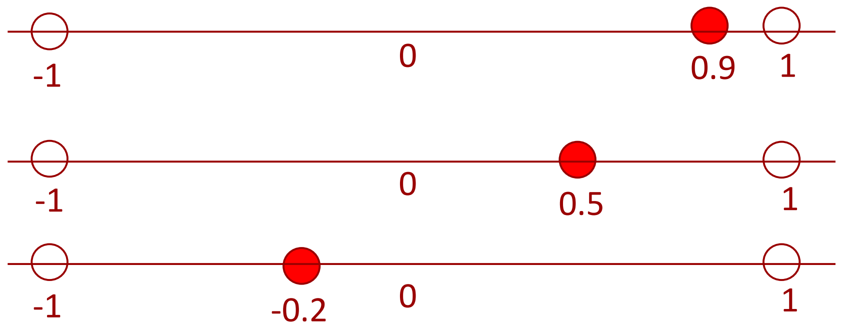
- Suppose we use a simple repeat code (repeat the bit 3 times) and transmit (or store) 1 1 1



Example of Repeat Code (cont.)

A hard decision means we make a decision on a 1 or 0 for each bit instead of considering all the bits before making decisions using squared metrics (which would be the case for soft decisions)

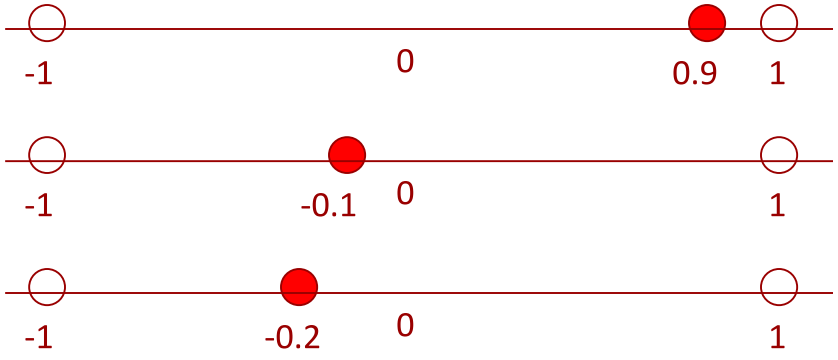
- Say we receive (or read) the following



- Making hard decisions we get 1 1 0 (1 1 -1) so we decide 1 was sent since 1 1 0 is closer to 111 than it is to 0 0 0

Example of Repeat Code (cont.)

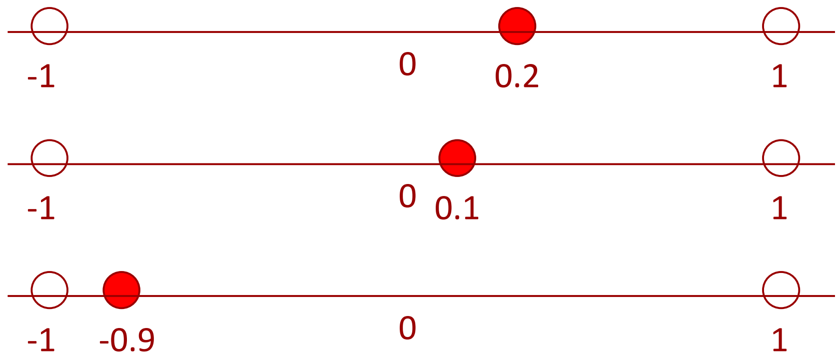
- But if we receive (or read)



- Making hard decisions we get 1 0 0 so we decide 0 was sent and thus we make an error
- But if we use soft decisions in this example we get the (squared) metrics 2.66 (for 1) and 5.06 (for –1) so we decide correctly that 1 was sent

Example of Repeat Code (cont.)

- However if we receive (or read)



- Making hard decisions we decide 1 was sent
- Making soft decisions we decide 0 was sent (an error)
- Despite this last example soft decisions give better performance on average but is more complex to implement

Hamming Codes

- Hamming codes were the first class of linear codes designed for error correction
- For any positive integer m >= 3 there is a Hamming code such that
  - Code length: $n = 2^m - 1$
  - Number of information symbols: $k = 2^m – m – 1$
  - Number of parity check symbols: $n – k = m$
  - Error-correcting capability: $t = 1$ ($d_{min} = 3$)
    - This means each codeword differs from any other codeword in at least 3 bit positions
    - t is number of correctable errors

How Hamming did it

- The syndrome results from writing a 0 for each of the parity bits that are correct and a 1 for each failure

- We can think of the syndrome as an m-bit number so it can represent at most 2^m things

- We need to represent the state of all the message symbols being correct plus the location of any single error in the n bits thus $2^m >= n+1$

Hamming Code Example

Example. Let us design for k = 4 message digits. Let m = 3. Note that 2^3 >= (4+3)+1 = 8.
In fact, 2^3 = 8.

Position	Binary Rep
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

We will use positions 1, 2, 4 of the code vector for the 3 parity checks. Every position that has a 1 in the last column of the binary rep. is the first parity check. The second parity check covers the next column over and so forth.

The 1st parity check covers positions 1, 3, 5, 7.
The 2nd parity check covers positions 2, 3, 6, 7.
The 3rd parity check covers positions 4, 5, 6, 7.

Notice that position
1 is covered in 1st parity => 001
2 is covered in 2nd parity => 010
3 is covered in 1st and 2nd parity => 011
4 is covered in 3rd parity => 100
5 is covered in 1st and 3rd parity => 101
6 is covered in 2nd and 3rd parity => 110
7 is covered in 1st, 2nd and 3rd parity => 111

The numbers on the right above are seen to be the binary representations of the bit position being covered

Each parity is set to a value (0 or 1) so that the bit positions it covers has an even number of 1's

Hamming Code Example (cont.)

Encode:

Position	1	2	3	4	5	6	7
Message	-	-	1	-	0	1	1
Encode	0	1	1	0	0	1	1
Receive	0	1	**0**	0	0	1	1
			error				

Decode:

Check 1	1	3	5	7
	0	0	0	1 **fails => 1**
Check 2	2	3	6	7
	1	0	1	1 **fails => 1**
Check 3	4	5	6	7
	0	0	1	1 correct =>0

The syndrome is 0 1 1 which is the binary rep. of 3 so position 3 is in error. We add 1 to position 3 of the received vector to get

0 1 1 0 0 1 1

as the decoded error vector.

Note: The syndrome mentioned here is the result of the error checks (a syndrome of 000 means no error was detected). Also note that all addition with these binary codes is modulo 2 (0+0=0, 0+1=1+0=1, 1+1=0)

Systematic Hamming Code

- The Hamming code can be put in systematic form so that the codeword contains the information sequence as consecutive bits
- In this case
  - The first parity check covers bit positions 1, 4, 6, 7
  - The second parity check covers bit positions 2, 4, 5, 6
  - The third parity check covers bit positions 3, 5, 6, 7
- Suppose that when a parity check is done on the data that is read the second and third parity check fails
  - We then note that only bit position 5 is covered in both the second and third parity checks (and not in the first) so we deduce that bit 5 is in error so we correct that bit

Systematic Hamming Codes

Data				Codewords						
0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	0	1	0	0	0
0	1	0	0	0	1	1	0	1	0	0
1	1	0	0	1	0	1	1	1	0	0
0	0	1	0	1	1	1	0	0	1	0
1	0	1	0	0	0	1	1	0	1	0
0	1	1	0	1	0	0	0	1	1	0
1	1	1	0	0	1	0	1	1	1	0
0	0	0	1	1	0	1	0	0	0	1
1	0	0	1	0	1	1	1	0	0	1
0	1	0	1	1	1	0	0	1	0	1
1	1	0	1	0	0	0	1	1	0	1
0	0	1	1	0	1	0	0	0	1	1
1	0	1	1	1	0	0	1	0	1	1
0	1	1	1	0	0	1	0	1	1	1
1	1	1	1	1	1	1	1	1	1	1

Some 4-bit data packets differ by only 1 bit

All codewords differ by at least 3 bits so if one bit is received in error it is still closer to the correct codeword than any other codeword

Encoding/Decoding Comments

- Developing a mathematically elegant encoding scheme that has nice distance properties for the codewords is difficult
  - But performing the encoding algorithm on a computer is fairly simple
- Developing a mathematically elegant decoding scheme that finds the best match codeword that is practical to implement is difficult
  - Also performing the decoding algorithm on a computer can be computationally intensive
    - But this is still much better than a brute force decoder that tries to find the best codeword thru an exhaustive search comparison
    - For good code performance without too much overhead the size of the data to be encoded (*k* bits) needs to be quite large – maybe in the thousands of bits
    - This means there are 2^k possible codewords which is prohibitive to consider directly in decoding if *k* is large (this is why we need decoding algorithms that utilize mathematical properties of the code)

Reed-Solomon Codes

- Reed-Solomon (RS) Codes are used in CDs/DVDs/Blu-rays and utilize hard decisions (soft decisions logic is quite complex)
  - Can still get very strong error correction performance with hard decisions
  - Most applications of RS codes in industry actually use hard decisions
  - An RS code is defined over a finite field and is an example of an algebraic code which utilizes modern algebra

Reed-Solomon Codes in CDs/DVDs/Blu-rays

- CDs utilize two RS codes separated by an interleaver
  – The interleaver spreads the data out so that a burst of errors during a read show up in different codewords instead of overwhelming just one or a few codewords that are contiguous and thus permits more error correction
  – This is referred to as Cross-Interleaved Reed-Solomon Coding (CIRC)
- DVDs also utilize two RS codes as a product code
  – Here the data is written into a matrix (like a block interleaver) and each row is encoded by a RS code and then each column is encoded by a RS code
    - The RS codes and resulting product coding scheme is more powerful than that used for CDs but is necessary to protect the more compact data as any imperfection on the disc affects more data for DVDs than for CDs
  – This is referred to as a Reed-Solomon Product Code (RSPC)
- Blu-rays utilize an even more powerful RS error correction with interleaving scheme

Conclusion

- Many codes are in use today
  - Hamming codes are still widely used for encoding headers in data packets (powerful codes like Reed-Solomon, turbo and LDPC codes are not appropriate for small amounts of data)

- We have seen that mathematics in the form of coding theory makes possible the excellent listening/viewing quality of CDs/DVDs/Blu-rays
  - Without the strong error correction capability of this applied mathematics it would not be possible to store so much data onto these discs and read it reliably
  - For example, for a random read bit error rate of $2x10^{-2}$ the DVD decoder can reduce this error rate to 10^{-15} which is 10 times better than decoding on a CD (ref. [2])
  - As another example, a disc was sprayed with dust and a byte error rate (a byte is 8 bits and 1 or more bit errors in the byte constitutes a byte error) of $4x10^{-3}$ was reduced to $5.7x10^{-8}$ by the DVD decoder and reduced to $1.5x10^{-18}$ by the Blu-ray decoder (ref. [2])

Appendix

History of some Noteworthy Codes

Applications of Coding

History of Some Noteworthy Codes

- Coding theory and information theory were firmly established with the publication of papers by Claude Shannon and Richard Hamming (both in the Bell System Technical Journal)
  - "A Mathematical Theory of Communication" by Shannon, 1948
  - "Error Detecting and Error Correcting Codes" by Hamming, 1950

- An important nonbinary code was discovered by M. J. E. Golay
  - "Notes on Digital Coding," Proc. IEEE, 1949

History of Some Noteworthy Codes (cont.)

- Reed-Muller coding was established by Irving Reed and D. Muller
  - "A Class of Multiple-Error-Correcting Codes and the Decoding Scheme" by Reed, IEEE Trans. On Info. Theory, 1954.
  - "Application of Boolean Algebra to Switching Circuit Design and to Error Detection" by Muller, IEEE Trans. On Computers, 1954

– RM codes were also discovered by N. Mitani
  - "On the Transmission of Numbers in a Sequential Computer" delivered at the National Convention of the Institute of Electrical Engineers of Japan, 1951

History of Some Noteworthy Codes (cont.)

- BCH codes were first introduced by A. Hocquenghem
  - "Codes Correcteurs d'Erreurs (Error Correcting Codes)", Chiffres, 1959
- BCH codes were independently discovered by R. C. Bose and D. K. Ray-Chaudhuri
  - "On a Class of Error Correcting Binary Group Codes", Information and Control, 1960
- Reed-Solomon codes were discovered by Irving Reed and Gus Solomon
  - "Polynomial Codes over Certain Finite Fields", J. SIAM, 1960

History of Some Noteworthy Codes (cont.)

- Idea of Sequential Decoding was published by J. M. Wozencraft and B. Reiffen
  - *Sequential Decoding*, M.I.T. Press, 1961
- Low density parity check codes were introduced by Gallager
  - "Low-Density Parity-Check Codes," IRE Trans. Info. Theory, 1962
- G. D. Forney, Jr. looked at concatenated coding
  - *Performance of Concatenated Codes,* MIT Press, 1966
- An optimal double error correcting nonbinary code was discovered A. W. Nordstrom and J. P. Robinson
  - "An Optimal Nonlinear Code," Information and Control, 1967

History of Some Noteworthy Codes (cont.)

- An algorithm for decoding convolutional codes was developed by A. J. Viterbi
  - "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," IEEE Trans. On Information Theory, 1967

- Turbo Codes were discovered by C. Berrou, A. Glavieux and P. Thitimajshima
  - "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," Proc. 1993 International Conference on Communications

Applications of Coding

- Some applications of codes include
  - Pioneer 9 solar orbit mission (launched in 1968) was the first deep-space mission using a convolutional code
  - Pioneer 10 (1972) and Pioneer 11 (1973) also used convolutional codes
    - These codes used the Fano sequential decoding algorithm
  - Voyager spacecraft (1977) used two convolutional codes at 100 Kbps to send pictures of Jupiter and Saturn back to Earth in 1979 and 1981, respectively
    - When studying Uranus in 1986 one convolutional code was concatenated with an outer Reed-Solomon code to improve performance and operated at 30 Kbps

Applications of Coding (cont.)

– A concatenated coding scheme (convolutional inner code and Reed-Solomon outer code) was utilized in the Galileo spacecraft (launched in 1989)

– The digital color television transmission system (DITEC) using the INTELSAT IV satellite link employed a systematic convolutional code with threshold decoding

– Mobile communication such as GSM in the United Kingdom (1988)

– CCITT (International Telephone and Telegraph Consultative Committee) adopted convolutional codes for use in telephone systems (1982, 1985, 1993)

  - Uses QAM (Quadrature Amplitude Modulation) signaling

Applications of Coding (cont.)

- The CCSDS (Consultative Committee for Space Data Systems) has adopted turbo codes as a new standard for telemetry channel coding (1997)
- Reed-Solomon codes are used in CDs and DVDs
- The CDMA2000 proposal recommends turbo codes for use in CDMA cellular mobile systems
- Turbo codes were proposed for INMARSAT's new multimedia service
  - Data rate is 64 Kbps and combines turbo coding with 16-QAM modulation

Applications of Coding (cont.)

- Low Density Parity Check codes have been adopted as a digital video broadcast standard

- The CCSDS has recommended Low Density Parity Check codes for use in near-earth and deep space applications(2007)

Error Correction Capability

- Error correction capability of a code depends on the minimum distance of the code
  - Measures how close code words are to each other
  - The distance measure is Hamming distance which is the number of places where the vectors differ
  - Example: 1 1 0 0 1 vs. 1 0 0 1 0 (Hamming distance 3)
  - Minimum distance is the smallest of all distances
- For a linear code the minimum distance is simply the fewest number of 1's found in all the nonzero code vectors
- A code can correct $t = (d_{min} - 1)/2$ or fewer errors

Coding Concepts

- Channel encoder adds redundancy to data by mapping k information bits to *n* code bits (codeword)
  - resulting codewords do not look like each other
  - Code rate R = k / n
  - Input rate is r_b => output rate is $r_c = r_b / R$ bps
- In transmissions of signals an M-ary modulator maps L encoded bits to one of $M=2^L$ waveforms
  - Each waveform is of T sec duration
  - Output symbol rate $r_s = 1 / T = r_b / RL$ symbols per sec
  - Modulation is typically phase or amplitude and phase
    - Others are possible