Coding Theory I INF 723

Lecture: Coding Theory

Stephen F. Bush

GE Global Research

March 6, 2008

Lecture Outline

1. What is Information?
   - Entropy
   - Kolmogorov Complexity
2. Source Coding
   - Huffman Coding
   - Network Coding
3. Channel Coding
   - Noise and Channel Capacity
   - Hamming Codes
   - CRC
   - Linear Codes
   - Zero-error Information Theory
   - Rate Distortion
4. Video Coding
   - Motion Estimation

Computer Communication Networks CSI 416/516

Instructor:	Dr. Stephen F. Bush, GE Global Research
Phone:	387-6827
Email:	bushsf@research.ge.com
Course Website:	http://www.cs.albany.edu/~bushsf

TODO

- add exercises from:
    - Graph Theory books
    - Network book
    - KC book
    - QC book

Scribe System

- Two different students will be assigned to take notes for each class, starting today
- Your notes will be in LaTeX; notes are due before the next class begins
- A LaTeXtemplate will be available on the class web page: `www.cs.rpi.edu/~bushs`; follow the instructions embedded inside the template
- The goal is to form a shared resource for all class participants; you will be depending on others' taking good notes
- Your grade will focus on content of your notes, however, if too much improper syntax results in the inability to process your LaTeX, then points will be deducted
- Please email me (`bushsf@research.ge.com`) your contact information

Networks

- Send information
  - Quickly
  - Low cost
- What does this mean?

What Is Information?

- Cost and delay are easily measurable
- What about information?
    - Amount?
    - Correctness?

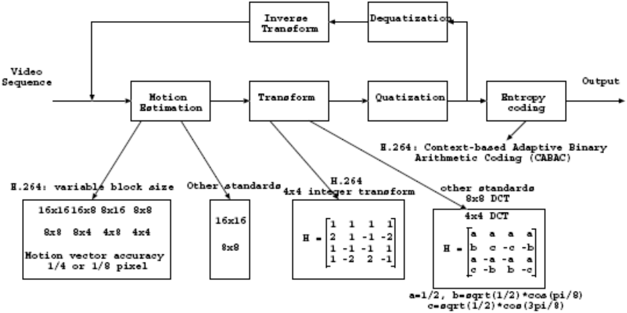
Information Representation

(show example from JPEG webpage)

- Choosing how to represent information impacts storage and transmission
- Consider an image stored as a two dimensional array of pixels
  - Chroma subsampling involves encoding images by implementing more resolution for luminance information than for color information
  - The human visual system is much more sensitive to variations in brightness than color
  - A video system can be optimized by devoting more bandwidth to Y' than the color difference components Cb and Cr
  - Y' is the luma component and Cb and Cr are the blue and red chroma components
  - The 4:2:2 Y'CbCr scheme for example requires two-thirds the bandwidth of (4:4:4) R'G'B'

Video Coding Representations

- Consider the impact of transmitting the output of each function without further processing...


Inverse Transform
Dequatization
Video Sequence
Motion Estimation
Transform
Quatization
Entropy coding
Output
H.264: Context-based Adaptive Binary Arithmetic Coding (CABAC)
H.264: variable block size
16x16 16x8 8x16 8x8
8x8 8x4 4x8 4x4
Motion vector accuracy 1/4 or 1/8 pixel
Other standards
16x16
8x8
H.264 4x4 integer transform
$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$
other standards 8x8 DCT
4x4 DCT
$H = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix}$
a=1/2, b=sqrt(1/2)*cos(pi/8)
c=sqrt(1/2)*cos(3pi/8)

Measuring Information

- Claude Shannon developed information theory to systematically measure the efficiency of communication
- Prior Art:
  - Fourier Transform
  - Nyquist telegraph efficiency
  - Hartley measured information content of messages
- Nyquist and Hartley were both at Bell Labs when Shannon arrived
- Kolmogorov and Kolmogorov Complexity

Some basics...

- A **channel code** contains redundancy to allow more reliable communication in the presence of noise. This redundancy means that only a limited set of signals is allowed: this set is the code.
- A **source code** is used to compress words (or phrases or data) by mapping common words into shorter words (e.g. Huffman Code).
- Note: A **code word** is an element of a code. Each code word is a sequence of symbols assembled in accordance with the specific rules of the code and assigned a unique meaning.

Nyquist modeled telegraphs and digital networks

Given:

C The rate at which information flows across a noiseless channel

$S(t)$ A signal which is a linear combination of periodic signals, so

$$S(t) = \sum_{i=0}^{n} S_i(t)$$

m The number of levels which S(t) can assume (correspond to number of symbols in an alphabet), each symbol has $\log_2$ m bits of information.

F The frequency of the highest frequency S(t) component. Nyquist showed the sampling rate must be at least 2F for accurate signal reconstruction. The rate which information can be sent across a noiseless channel is: $C = 2F \log_2 m$

Hartley's Work

- Hartley used the notation:

h	the amount of information in a message
n	the amount of randomly selected symbols
s	the number of symbols

- To derive: $h = n \log s$
- Assumes that the individual symbols are selected with equal probability
- This was just before the dawn of the digital age and assumes decimal digits

Hartley's 1928 paper, called simply *Transmission of Information*, made explicitly clear that information was a measurable quantity, reflecting only the receiver's ability to distinguish that one sequence of symbols had been intended by the sender rather than any other. The Hartley information, H_0, is still used as a quantity for the logarithm of the total number of possibilities.

What is Information?

- Consider a stopped clock. Does looking at that clock provide any information?
- What about flipping an unfair coin that always comes up heads? Would such a coin toss provide new information?
- A comprehensive quantitative analysis of information is illusive, often ultimately depending upon the subjective intelligence of the receiver.
- Shannon and Kolmogorov have been the best innovators in this field.

Information and Predictability

- A totally predictable process does not provide information.
- Consider the coin toss example again, but with a fair coin this time.
- Each time we flip the coin, we get one bit of information
- Bit is coincidentally (haha) the same word as in digital logic and has a similar (but not exactly the same) meaning.

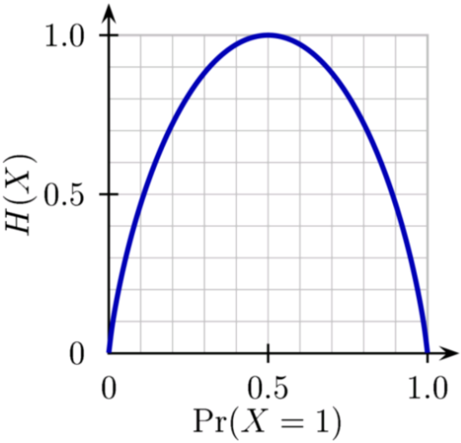
Introduction to Entropy

- Shannon defined the entropy of an information source.
- Informally, entropy refers to the unpredictability of the information from physics and thermodynamics.
- More formally, for an information source sending messages composed of symbols from an alphabet, $X = x_1, x_2, x_3, \ldots$ where p_i is the probability of transmitting the symbol x_i, the entropy of that information source is:

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i$$

- (Note when $p_i = 0$, we use $p_i \log_2 p_i = 0$)

Entropy Is Convex



An Entropy Example

- Suppose that an information source sends H for heads and T for tails of a toss of a coin, with the probabilities p_h and p_t respectively.
- In class, solve for the per symbol entropy of this information source for the following cases:
  - 1. $p_h = p_t = 1/2$ (a fair coin)
  - 2. $p_h = 1, p_h = 0$ (a certain event)
  - 3. $p_h = 3/4, p_t = 1 - p_h = 1/4$ (an unfair coin)
  - Hint: recall that entropy measure as $H = -\sum_{i=1}^{n} p_i \log_2 p_i$:

    ...and when $p_i = 0$, we use $p_i \log_2 p_i = 0$.

Solution

- Applying our definition and substituting we get:
  $H = -p_h \log(p_h) - p_t \log(p_t)$
- Solving for each case by substituting in the values given we get:
  1. $p_h = p_t = 1/2$ (A fair coin).
  $H = -1/2 \log_2(1/2) - 1/2 \log_2(1/2)$
  $H = -1/2 \times (-1) - 1/2 \times (-1)$
  $H = 1/2 + 1/2 = 1$

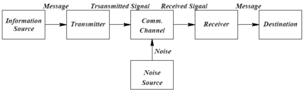
Solution

- 2. $p_h = 1, p_t = 0$ (A certain event).
  $H = -1\log_2(1) - 0\log_2(0)$
  $H = (-1 \times 0) + 0 = 0$
- 3. $p_t = 3/4, p_t = 1 - p_h = 1/4$ (An unfair coin)
  $H = -3/4\log_2(3/4) - 1/4\log_2(1/4)$
  $H \approx -(3/4 \times -0.415) - (1/4 \times -2)$
  $H = 0.811$

Noise

- Noise in information theory is a corruption of a signal across a communication channel (so the receiver does not get what was sent)

Shannon modeled the following system



Noise and Entropy

- Suppose that an information source sends a symbol A_s and a destination received a symbol A_r. Assume the symbols are selected from an alphabet $A = A_1, \ldots, A_m$.
- The entropy of the sender of the sender and the receiver are:

$$H(S) = \sum_{S=1}^{m} Pr(S) \log Pr(S)$$

$$H(R) = \sum_{R=1}^{m} Pr(R) \log Pr(R)$$

- The entropy of S being sent and R being received is:

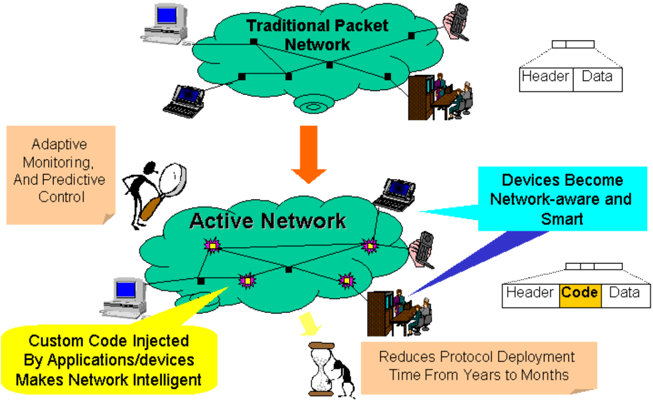
$$H(S,R) = \sum_{S=1}^{m} \sum_{R=1}^{m} Pr(S,R) \log Pr(S,R)$$

- Where $Pr(S,R)$ is the probability of S Being sent and R being received.

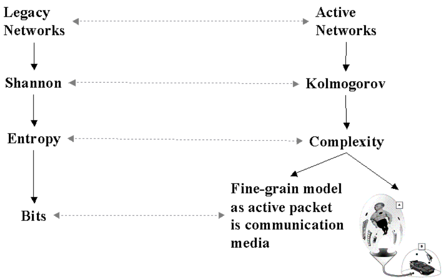
Conditional Entropy

- Recall that the conditional probability of X given that Y was observed is denoted $Pr(X|Y)$.
- Bayes theorem states that if X and Y are independent events then: $Pr(X|Y) = \frac{Pr(X,Y)}{Pr(y)}$
- The entropy (uncertainty) of R being received given that S was sent is:
$$H(R|S) = -\sum_{R=1}^{m}\sum_{S=1}^{m} Pr(S)Pr(R|S)\log_2 Pr(R|S)$$
- The entropy (uncertainty) of S being sent given that R was received is:
$$H(S|R) = -\sum_{R=1}^{m}\sum_{S=1}^{m} Pr(R)Pr(S|R)\log_2 Pr(S|R)$$

Active Networking: A Natural Evolution


Traditional Packet Network
Header | Data
Adaptive Monitoring, And Predictive Control
Active Network
Devices Become Network-aware and Smart
Header | Code | Data
Custom Code Injected By Applications/devices Makes Network Intelligent
Reduces Protocol Deployment Time From Years to Months

Shannon versus Kolmogorov



Legacy Networks

Active Networks

Shannon

Kolmogorov

Entropy

Complexity

Bits

Fine-grain model as active packet is communication media

[1]

[1] *Active Virtual Network Management Prediction: Complexity as a Framework for Prediction, Optimization, and Assurance* Stephen F. Bush, Proceedings of the 2002 DARPA Active Networks Conference and Exposition (DANCE 2002), IEEE Computer Society Press, pp. 534-553, ISBN 0-7695-1564-9, May 29-30, 2002, San Francisco, California, USA.

Kolmogorov Complexity

- Consider an Active Network: Data can be sent as executable code.

$$K_{\phi}(x) = \min_{\phi(p)=x} l(p)$$

- Which is the length (l) of the smallest program (p) on a Universal Turing machine (ϕ) that generates a of given string of bits (x).

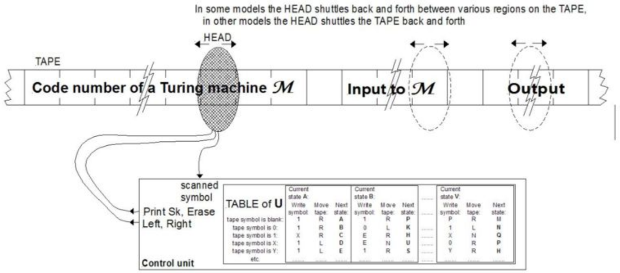
Turing Machine

Recall

- Q is a finite set of states
- Γ is a finite set of the tape alphabet
- $b \in \Gamma$ is the blank symbol (the only symbol allowed to occur on the tape infinitely often at any step during the computation)
- Σ, a subset of Γ not including b is the set of input symbols
- $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R\}$ is a partial function called the transition function, where L is left shift, R is right shift.
- $q_0 \in Q$ is the initial state
- $F \in Q$ is the set of final or accepting states

Universal Turing Machine

- Turing first universal Turing machine
- Marvin Minsky in the early 1960s (7 states)



In some models the HEAD shuttles back and forth between various regions on the TAPE, in other models the HEAD shuttles the TAPE back and forth

TABLE of U	Current state A:			Current state B:				Current state V:		
	Write symbol:	Move tape:	Next state:	Write symbol:	Move tape:	Next state:		Write symbol:	Move tape:	Next state:
tape symbol is blank:	1	R	A	1	R	P		P	R	M
tape symbol is 0:	1	R	B	0	L	K		1	L	N
tape symbol is 1:	X	R	C	E	R	H		X	N	Q
tape symbol is X:	1	L	D	E	N	U		0	R	P
tape symbol is Y:	1	L	E	1	R	S		Y	R	H
etc.										

The **Universal machine U** consists of a set of instructions in the TABLE that can "execute" the correctly-formulated "code number" of any arbitrary Turing machine $\mathcal{M}$ on its TAPE.
(Entries in the TABLE are fictitious; drawing partially after Davis (2000), p. 164.

Kolmogorov Complexity

(Show some examples from KC book)

Kolmogorov Complexity

Kolmogorov Complexity and Active Networks

$K(x)$ is the complexity or alternative view of the amount of information. In active network: send smallest program that generates bits to be transmitted [a].

[a] Active Virtual Network Management Prediction: Complexity as a Framework for Prediction, Optimization, and Assurance S. F. Bush, Proceedings of the 2002 DARPA Active Networks Conference and Exposition (DANCE 2002), IEEE Computer Society Press, pp. 534-553, ISBN 0-7695-1564-9, May 29-30, 2002, San Francisco, California, USA.

Open Problem: Cannot derive or prove that smallest program has been found in the general case (opportunity?).

Data Compression and Information Theory

Data compression: replace longer low-entropy data representations with equivalent shorter high entropy data representations

Compression Creates the shorter/higher entropy representation given the longer low entropy source.

Lossy or lossless Sometimes data can be lost (sound/video) without major degradation.

- Statistical compression replaces high frequency symbols with shorter representations(e.g. Huffman Encoding)

Substitutional compression Replaces sequences of symbols with short patterns (LZW compression)

Decompression Restores data to its original format (or an approximation if lossy techniques are used).

Huffman Coding

- Huffman codes are an optimal statistical compression on a per character basis (the old UNIX *compact* command).
  - Compression typically requires two passes
    - Count the frequency of each character in the input and then construct the Huffman encoding tree.
    - Emit the dictionary mapping each encoding symbol to the original character, and for each character in the input append its encoded representation.
  - Decompression-single pass, read the dictionary and invert the encoding.

Building a Hoffmann Coding Tree

1. Take each character and create a node containing the character and its frequency with no children.
2. While there are two or more nodes in the set.
   - (a) remove the 2 nodes with the lowest frequency
   - (b) make a new node having these 2 nodes as its children, and set its frequency to the sum of its child frequencies.
   - (c) insert the new node back into the set of nodes
3. Assign the nodes in the tree and coding as follows:
   - (a) for each leaf, traverse the tree from root to leaf
     - (i) if the sub-tree is the left sub-tree, append a zero to the encoding
     - (ii) else append a 1 to the encoding

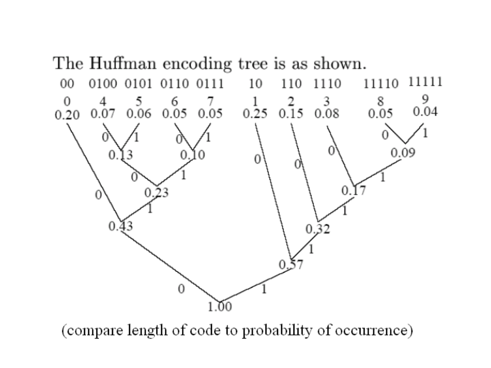
In-class Exercise 1 of 2

Consider a file composed of ASCII digits, where each digit has the following frequency

Symbol	0	1	2	3	4	5	6	7	8	9
Frequency	0.20	0.25	0.15	0.08	0.07	0.06	0.05	0.05	0.0	0.04

1. Draw Hoffman encoding tree(derive the encoding)
2. Suppose that these frequencies occurred in 10^6 bytes of data, how many bytes would it take to code this data(ignoring the dictionary).

In-class Exercise 1 of 2



The Huffman encoding tree is as shown.

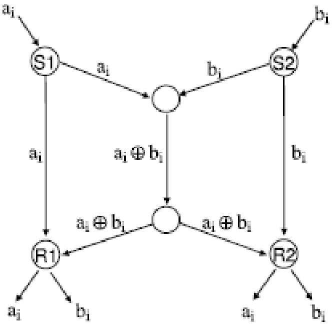
(compare length of code to probability of occurrence)

In-class Exercise 1 of 2

- Space needed = $\sum_{i=0}^{9} 10^6 \text{characters} \times \text{frequency(i)} \times \text{BitsToEncode(i)} \times \frac{1 byte}{8 bits}$
- Space needed $= 10^6 \times$ Mean Bits Per Character $\times \frac{\text{8bits}}{\text{byte}}$
- Space needed $= 10^6 \times$ Mean Bits Per Character $\times \frac{\text{1byte}}{\text{8bits}}$
- Space needed $= 10^6 \times$ 3.04bits per character $\times \frac{\text{8bits}}{\text{byte}}$
- Space needed $= 380,000$bytes
- Since ASCII coding requires 10^6 bytes and BCD requires 500,000 bytes, we get compression ratios of 2.63 and 1.32 respectively.

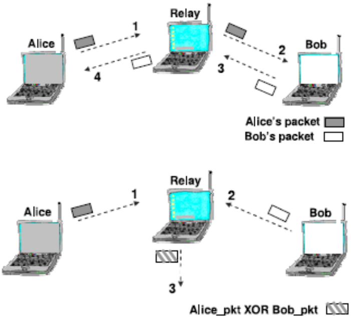
Network Coding: The Concept

- Both messages are received by both receivers due to xor ($\oplus$) operation
- Maximum flow through the network links is achieved
- Recall (and contrast) with our maximum flow optimization homework problem



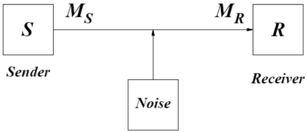
Network Coding: The Application

- Consider a simple wireless routing scenario (with and without network coding)
- 4 separate transmissions required without network coding
- 3 separate transmissions utilizing coding (same xor operation as in previous slide)



The Error Detection Problem

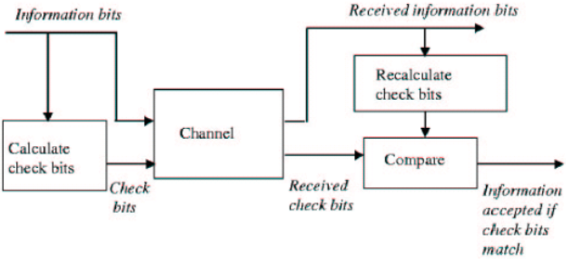
- Sometimes errors occur during communication (we will investigate how and why later).
- Suppose a sender S sends a message M_s to a receiver R which gets message M_r. The transmission is successful if $M_s = M_r$, otherwise the message was garbled during transmission.
- How could we know if and $M_s = M_r$ when S and R are different machines?



Introduction to Error Detecting Codes 1 of 2

- The answer is that it is impossible to know if $M_r = M_s$ with complete certainty, but we can check with high probability as follows:
  - The sender and receiver use an agreed upon function F which operates on a message as its input.
  - The sender computes $G_s = F(M_s)$.
  - The sender transmits a message containing M_s and G_s.
  - The receiver decodes the received message Mr and received function evaluation G_r.
  - The receiver accepts the message as valid if and only if $F(M_r) = G_r$, otherwise an error is detected.

Intro to Error Detecting Codes 1 of 2



Noiseless Channel Capacity

Shannon's Fundamental Theorem for a Noiseless Channel states:

Theorem 2

Given a source with entropy H bits per symbol and a noiseless channel with capacity C bits per second, then it is possible to transmit information at a rate of:

- $\frac{C}{H} + \epsilon$ for arbitrarily small values of ϵ but not faster.
- In practice this implies that no channel can transmit data faster than:
  $$\lim_{\epsilon \to 0} \frac{C}{H} + \epsilon = \frac{C}{H} \text{ (symbols/second)}$$

Noise and Channel Capacity

- Shannon was able to show the (surprising) result that a communication channel's capacity gradually degraded as noise is added.

Theorem 3

The capacity of a channel with bandwidth B, signal power S, and noise $N > 0$ is:

$$C = B \log_2 \frac{S+N}{N}.$$

So channel capacity decreases gradually in response to noise.

Thermal Noise

- From physics it can be shown that thermal noise is $N = kTB$, where k is Boltzmans constant and T is temperature measured in Kelvin, implying that:

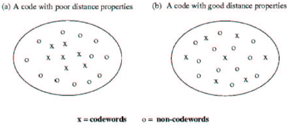
$$C = B \log_2 \frac{S+N}{N}$$

$$C = B \log_2 \frac{S+kTB}{kTB}$$

- Pierce states that as a bandwidth B increases it can be shown the limiting capacity of the channel is approached: $C = \frac{BS}{kT} \ln 2$ bits per second

Hamming Distance an Error Detection

- Consider to messages X and Y composed of symbols from the same alphabet.
- The Hamming Distance between x and y is defined as the number of positions where the symbol in X does not match the corresponding symbol in Y.
- Error detection and correction require that code words (valid messages) requires a large Hamming Distance (or a small number of errors becomes undetectable).

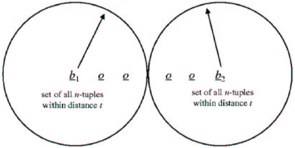


(a) A code with poor distance properties

(b) A code with good distance properties

Distance and Error Detection

- Distance (Hamming Distance) between code words allows invalid messages to be detected (since the fall in the gaps).
- Note: Distance may not necessarily be Euclidean and maybe in a many dimensioned space.
- Typically error correction assumes that the nearest code word to received a message is the intended message.



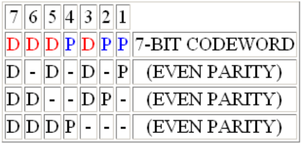
If $d_{\min} = 2t+1$, non-overlapping spheres of radius t can be drawn around each codeword; t=2 in the figure

Hamming Example I

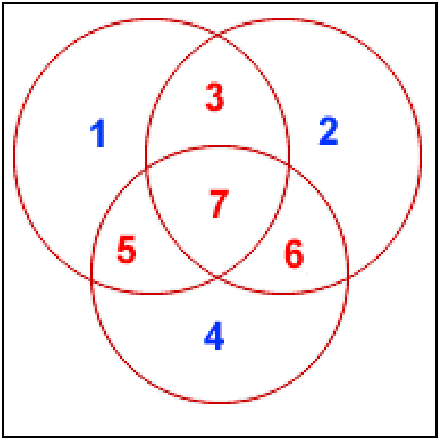
- Notice how parity is distributed among the data bits

D Data bit

P Parity bit

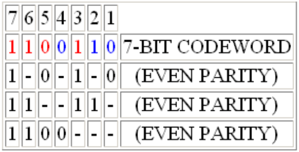


7	6	5	4	3	2	1	
D	D	D	P	D	P	P	7-BIT CODEWORD
D	-	D	-	D	-	P	(EVEN PARITY)
D	D	-	-	D	P	-	(EVEN PARITY)
D	D	D	P	-	-	-	(EVEN PARITY)

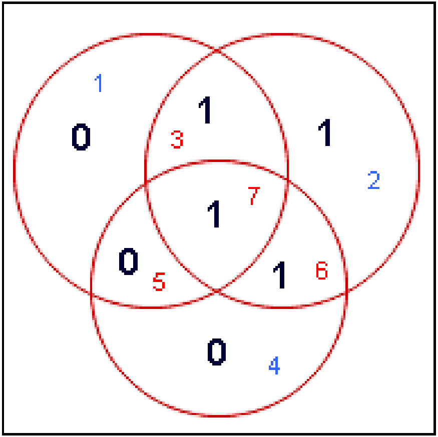


Hamming Example II

- With a valid code, parity in each circle is properly maintained.

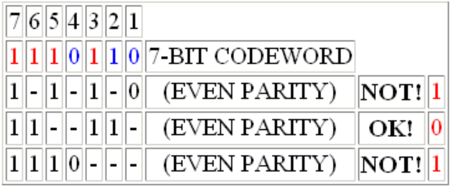


7	6	5	4	3	2	1	
1	1	0	0	1	1	0	7-BIT CODEWORD
1	-	0	-	1	-	0	(EVEN PARITY)
1	1	-	-	1	1	-	(EVEN PARITY)
1	1	0	0	-	-	-	(EVEN PARITY)



Hamming Example III

- An example with an erroneous bit...

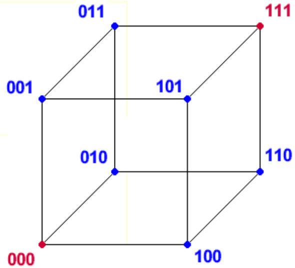


7	6	5	4	3	2	1			
1	1	1	0	1	1	0	7-BIT CODEWORD		
1	-	1	-	1	-	0	(EVEN PARITY)	NOT!	1
1	1	-	-	1	1	-	(EVEN PARITY)	OK!	0
1	1	1	0	-	-	-	(EVEN PARITY)	NOT!	1

```
      transmitted message                            received message
      1 1 0 0 1 1 0                                    1 1 1 0 1 1 0
BIT:  7 6 5 4 3 2 1           ------------->     BIT:  7 6 5 4 3 2 1
```

Hamming Example IV

- Hamming distance of three
- Only 000 and 111 are valid codewords
- The correct codeword is near any single erroneous bit-flip



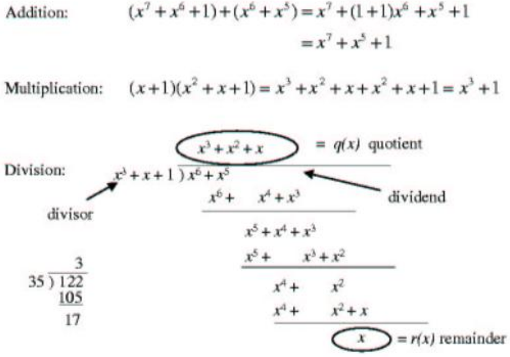
Introduction to CRC

- Cyclic redundancy codes (CRC) are among the most powerful methods for checking errors (as we will discover later).
- A CRC uses an nth degree generator polynomial $G(x)$ with coefficients of either 0 or 1 using modulo 2 arithmetic. Typically the high and low order coefficients must be 1.
- We represent a polynomial as the bit string $B = b_n, b_n - 1, ..., b_0$ such that:
- $G(x) = \sum_{i=0}^{n} b_i x^i$
- So for if $G(x) = x^4 + x + 1$, then $B = 10011$

CRC Computation

- In modulo to arithmetic:
    - Additions (and subtraction) do not carry or borrow, and are equivalent to exclusive-or operator ($\oplus$).
    - To divide, the traditional long division methods are performed using modulo-2 subtraction.
- To compute the CRC we:
    - Append n zeros to our message $M(x)$.
    - Using the modulo to arithmetic, we compute:
      $C(x) = M(x) \mod G(x)$
      where "mod" is the remainder of division.

CRC Computation



Addition: $(x^7+x^6+1)+(x^6+x^5)=x^7+(1+1)x^6+x^5+1$

$=x^7+x^5+1$

Multiplication: $(x+1)(x^2+x+1)=x^3+x^2+x+x^2+x+1=x^3+1$

Division:

$$\begin{array}{r l}
 & \boxed{x^3+x^2+x} \; = q(x) \text{ quotient} \\
x^3+x+1 \;) & x^6+x^5 \\
 & x^6+\quad x^4+x^3 \\
\hline
 & \quad x^5+x^4+x^3 \\
 & \quad x^5+\quad x^3+x^2 \\
\hline
 & \qquad x^4+\quad x^2 \\
 & \qquad x^4+\quad x^2+x \\
\hline
 & \qquad\qquad \boxed{x} \; = r(x) \text{ remainder}
\end{array}$$

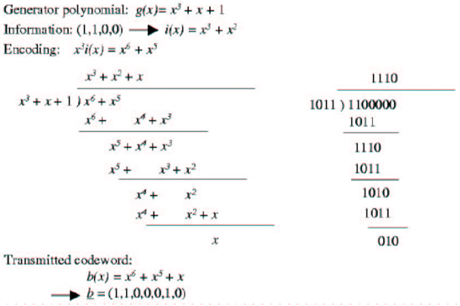
divisor

dividend

$$\begin{array}{r l}
 & \;\;3 \\
35 \;) & 122 \\
 & \underline{105} \\
 & \;\;17
\end{array}$$

The Computation Example Revisited

Computers store bits, and networks transmit bits, not polynomials. To convert, we treat the bit string as the coefficient of the system and then evaluate the CRC.



Generator polynomial: $g(x)= x^3 + x + 1$

Information: (1,1,0,0) $\longrightarrow$ $i(x) = x^3 + x^2$

Encoding: $x^3 i(x) = x^6 + x^5$

$$\begin{array}{r|l}
 & x^3 + x^2 + x \\
\hline
x^3 + x + 1 & x^6 + x^5 \\
 & \underline{x^6 + \quad x^4 + x^3} \\
 & \quad x^5 + x^4 + x^3 \\
 & \quad \underline{x^5 + \quad x^3 + x^2} \\
 & \qquad x^4 + \quad x^2 \\
 & \qquad \underline{x^4 + \quad x^2 + x} \\
 & \qquad\qquad\quad x
\end{array}
\qquad
\begin{array}{r|l}
 & 1110 \\
\hline
1011 & 1100000 \\
 & \underline{1011} \\
 & \ 1110 \\
 & \ \underline{1011} \\
 & \ \ 1010 \\
 & \ \ \underline{1011} \\
 & \ \ \ 010
\end{array}$$

Transmitted codeword:

$b(x) = x^6 + x^5 + x$

$\longrightarrow$ $\underline{b} = (1,1,0,0,0,1,0)$

Key Point

- Encoding: adds the remainder after dividing by g(x), this is the same as subtracting the remainder, making the result evenly divisible by g(x):
$$b(x) = g(x)q(x) + r(x) + r(x) = g(x)q(x)$$
- Decoding: because result is evenly divisible by g(x), remainder must be zero if valid, otherwise error.

Linear Codes – Sender

- Append $n - k$ check bits that are functions of the k information bits
- Example: $n = 7, k = 4$

$$\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$Hb^T = 0$$

Linear Codes – Receiver

(need more examples)

- Repeat the operation at the sender (assume H is known):

$$\overset{\text{H}}{\begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}} \overset{\text{Received data}}{\begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix}} = \overset{\text{syndrome}}{\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}}$$

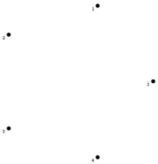
- If not 0 then an error has occurred. Ideally, the precise location of the error can be found via: $e = H^{-1}s$
- Unfortunately, H not invertible so work-arounds are explained in the text
- (How could we make H invertible and what would be the implications?)

Joint Source-Channel Coding

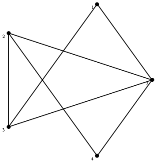
Note competing goals between source and channel coding: source coding removes bits to squeeze information into only the most necessary bits, resulting in short, but delicate information, while channel coding adds bits, yielding fat but robust information.

Basics

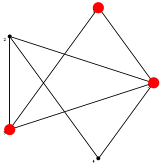
A graph has nodes:



A graph has edges:



A graph has cliques (sets of pairwise adjacent vertices):



Basics

- $\omega(H)$ is the size of the maximum clique of graph H
- In a coloring, adjacent vertices receive different colors
- $\chi(H)$ is the minimum number of colors needed
- Clearly, $\chi(H) \geq \omega(H)$

Perfect graph

- A graph G is perfect if $\chi(H) = \omega(H)$ for every induced subgraph of H

Definition: hole

A hole is a cycle of length at least four; its complement is an antihole. A hole/antihole in G is an induced subgraph that is a hole/antihole.

- Graphs that are not perfect
  - Odd holes
  - Odd antiholes
  - Graphs that have an odd hole or odd antihole

Examples of perfect graphs

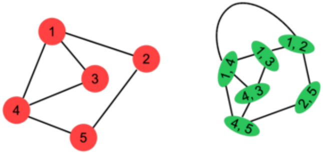
- Bipartite graphs ($\omega = 2 = \chi$)
  - and their complements
- Line graphs of bipartite graphs
  - and their complements
- 96 known classes of perfect graphs

Line graph

Definition: Line graph

The line graph $L(G)$ of an undirected graph G is a graph such that

- each vertex of $L(G)$ represents an edge of G; and
- any two vertices of $L(G)$ are adjacent if and only if their corresponding edges share a common endpoint in G



Perfect graph theorems

Theorem (The Perfect Graph Theorem (Lovász 1972))

graph is perfect ⇔ *its complement is perfect.*

Theorem (The Strong Perfect Graph Conjecture (Berge 1960))

A graph is perfect ⇔ *it has no odd hole and no odd antihole ("Berge graph")*

Relation to zero-error information theory

- Consider a discrete memoryless channel.
- Elements of a finite alphabet Σ are transmitted, some pairs of elements may be confused

Example

Let $\Sigma = a, b, c, d, e$. Assume that ab, bc, cd, de, ea may be confused.

So a, c may be sent without confusion $\Rightarrow 2^n$ n-symbol error-free messages.

But, ab, bd, ca, dc, ee are pairwise unconfoundable $\Rightarrow$ $5^{n/2} = 2^{(\frac{1}{2}\log 5)n}$ n-symbol error-free messages.

Shannon capacity via graphs

- Let $V(G) = \Sigma$, where a, b are adjacent if unconfoundable

Definition:Shannon Capacity

$$C(G) \equiv \lim_{n\to\infty} \frac{1}{n} \log \omega(G^n)$$

- G^n is the graph cartesian product of the symbols
- largest cliques is the largest unconfoundable group of symbols
- We have $\omega^n(G) \leq \omega(G^n) \leq \chi(G^n) \leq \chi^n(G)$ and so if $\omega(G) = \chi(G)$, then they determine $C(G)$
- Lovász proved that $C(C_5) = \frac{1}{2}\log 5$, using geometric representations of graphs (θ function)
- $C(G)$ is unknown in general

Perfect graphs are beautiful

- Communication theory (related to Shannon capacity and entropy)
- Sorting
- Polyhedral combinatorics
- Radio channel assignment
- Fundamental and deceptively simple-looking unsolved problems

Zero-error information theory

- Traditional information theory provides coding theorems in which a small, but greater than zero, probability of error is tolerated
- Zero-error information theory is concerned with asymptotically achievable rates and capacities with a probability of error strictly equal to zero
- "The zero-error capacity of a noisy channel", Shannon, 1956.
- How many bits can we send over a discrete memoryless channel with zero probability of error?

Shannon capacity: A definition (1)

- A discrete memoryless channel (DMC) is defined by a conditional probability distribution of the form $W(y|x)$
- It is given as a matrix, where the rows are indexed by elements of X, and the columns by elements of Y
- The definition is extended to n-vectors using the notation $W^n(y|x)$
- Two sequences x' and x'' of size n of input variables are distinguishable by a receiver iff the vectors $W^n(.|x')$ and $W^n(.|x'')$ are orthogonal

Shannon capacity: A definition (2)

- $N(W, n)$ is the maximum cardinality of a set of mutually orthogonal vectors among the $W^n(.|x), x \in X^n$
- The zero-error capacity of W is:

$$C_0(W) = \limsup_{n \to \infty} \frac{1}{n} \log N(W, n)$$

Characteristic graph

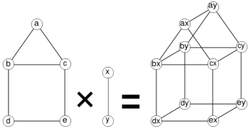
- $G = G(W)$ is defined by a set of vertices $V(G) = X$ and a set of edges $E(G)$ such that two symbols x' and x'' are adjacent if they are distinguishable (i.e. there does not exist a y such that $W(y|x') > 0$ and $W(y|x'') > 0$)
- The definition is extended to vectors using the nth OR-power $G^n = G(W^n)$ of G
- More precisely, $x', x'' \in E(G^n)$ if for at least one i the ith coordinates of x' and x'' satisfy $\{x'_i, x''_i\} \in E(G)$

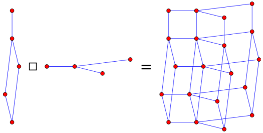
Graph product confusion

- The tensor product $G \times H$ of graphs G and H is a graph such that the vertex set of $G \times H$ is the Cartesian product $V(G) \times V(H)$ and any two vertices (u, u') and (v, v') are adjacent in $G \times H$ if and only if u' is adjacent with v' **and** u is adjacent with v
- The tensor product is also called the **direct product**, **categorical product**, **cardinal product**, or **Kronecker product**. It is also equivalent to the Kronecker product of the adjacency matrices of the graphs
- The notation $G \times H$ is also sometimes used to refer to the Cartesian product of graphs, but more commonly refers to the tensor product. The cross symbol shows visually the two edges resulting from the tensor product of two edges

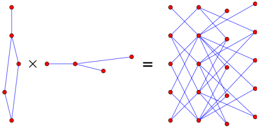
Graph products

Cartesian



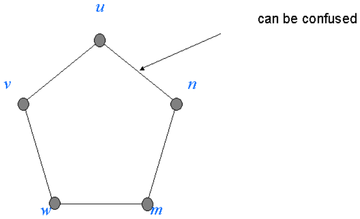


Tensor



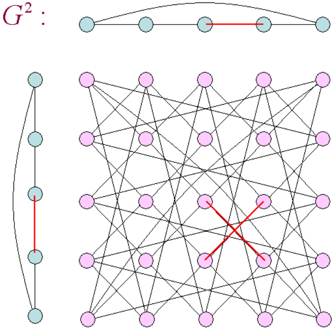
Noisy channels

Alphabet $\{u, v, w, m, n\}$
Largest safe subset: $\{u, m\}$



But if we allow words...

- Safe subset: $\{uu, nm, mv, wn, vw\}$
- Shannon capacity of G: $C(G) = \lim_{k\to\infty} \sqrt[k]{\alpha(G^k)}$



Reaching perfection...

- $C(G) \geq \alpha(G)$
- For which graphs does $C(G) = \alpha(G)$ hold?
- Which are the minimal graphs for which $C(G) > \alpha(G)$?
- Sufficient for equality: G can be covered by $\alpha(G)$ cliques.
- $\alpha(G) = \chi(\bar{G})$

Convex hull

- The convex hull of X can be described as the set of convex combinations of points from X: that is, the set of points of the form $\sum_{j=1}^{n} t_j x_j$, where n is an arbitrary natural number, the numbers t_j are non-negative and sum to 1, and the points x_j are in X
- So the convex hull $H_{\mathrm{convex}}(X)$ of set X is:

$$H_{\mathrm{convex}}(X) = \left\{ \sum_{i=1}^{k} \alpha_i x_i \;\middle|\; x_i \in X,\ \alpha_i \in \mathbb{R},\ \alpha_i \geq 0,\ \sum_{i=1}^{k} \alpha_k = 1,\ k = 1, 2, \ldots \right\}. \tag{1}$$

- The convex hull is defined for any kind of objects made up of points in a vector space, which may have any number of dimensions. The convex hull of finite sets of points and other geometrical objects in a two-dimensional plane or three-dimensional space are special cases of practical importance.

Graph entropy

- The characteristic vector of a stable set S of G is a vector $x \in \{0,1\}^{|G|}$, for which $x_v = 1$ iff $v \in S$
- The vertex packing polytope $VP(G)$ of graph G is the convex hull of the characteristic vectors of G

Theorem (Graph entropy)

The graph entropy of G with respect to a distribution $P = (p_1, p_2, ..., p_n)$ on $V(G)$ is

$$H(G,P) \equiv \min_{a \in VP(G)} \sum_{i=1}^{n} p_i \log \frac{1}{a_i}.$$

Rate Distortion

- Ratedistortion theory gives theoretical bounds for how much compression can be achieved (using lossy compression)
- Many of the existing audio, speech, image, and video compression techniques have transforms, quantization, and bit-rate allocation procedures that capitalize on the general shape of ratedistortion functions
- Ratedistortion theory was created by Claude Shannon in his foundational work on information theory
- Rate is usually understood as the number of bits per data sample to be stored or transmitted
- In the most simple case (which is actually used in most cases), distortion is defined as the variance of the difference between input and output signal (i.e., the mean squared error of the difference)
- Lossy compression techniques operate on data that will be perceived

Rate Distortion

- The functions that relate the rate and distortion are found as the solution of the following minimization problem:

  $\inf_{Q_{Y|X}(y|x)} I_Q(Y;X)$ subject to $D_Q \leq D^*$
- $Q_{Y|X}\ X_(y|x)$, sometimes called a test channel, is the conditional probability density function (PDF) of the communication channel output (compressed signal) X for a given input (original signal) X, and $I_{Q_(Y|X)}$ is the mutual information between Y and X defined as

  $I(Y;X) = H(Y) - H(Y|X)$

  where $H(Y)$ and $H(Y|X)$ are the entropy of the output signal Y and the conditional entropy of the output signal given the input signal, respectively:

  $$H(Y) = \int_{-\infty}^{\infty} P_Y(y) \log_2(P_Y(y))\, dy$$

  $$H(Y|X) = \int^{\infty} \int^{\infty} Q_{Y|X}(y|x) P_X(x) \log_2(Q_{Y|X}(y|x))\, dx\, dy$$

Rate Distortion

- Can also be formulated as Distortion-Rate function, where we find the supremum over achievable distortions for given rate constraint. The relevant expression is:

  $\inf_{Q_{Y|X}(y|x)} E[D_Q[X,Y]]$subject to $I_Q(Y;X) \leq R$
- The two formulations lead to functions which are inverses of each other
- The mutual information can be understood as a measure for prior uncertainty the receiver has about the sender's signal $(H(Y))$, diminished by the uncertainty that is left after receiving information about the sender's signal $(H(Y|X))$
- The decrease in uncertainty is due to the communicated amount of information, which is $I(Y;X)$
- As an example, in case there is no communication at all, then $H(Y|X) = H(Y)$ and $I(Y;X) = 0$

Rate Distortion

- In the definition of the ratedistortion function, DQ and D^* are the distortion between X and Y for a given $QY|X(y|x)$ and the prescribed maximum distortion, respectively
- When we use the mean squared error as distortion measure, we have (for amplitude-continuous signals):

$$D_Q = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P_{X,Y}(x,y)(x-y)^2 \, dx \, dy =$$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} Q_{Y|X}(y|x) P_X(x)(x-y)^2 \, dx \, dy$$

- As the above equations show, calculating a ratedistortion function requires the stochastic description of the input X in terms of the PDF $P_X(x)$, and then aims at finding the conditional PDF $QY|X(y|x)$ that minimize rate for a given distortion D^*

Rate Distortion

- These definitions can be formulated measure-theoretically to account for discrete and mixed random variables as well
- An analytical solution to this minimization problem is often difficult to obtain except in some instances for which we next offer two of the best known examples
- The ratedistortion function of any source is known to obey several fundamental properties, the most important ones being that it is a continuous, monotonically decreasing convex (U) function and thus the shape for the function in the examples is typical (even measured ratedistortion functions in real life tend to have very similar forms)

Rate Distortion

- Although analytical solutions to this problem are scarce, there are upper and lower bounds to these functions including the famous Shannon lower bound (SLB), which in the case of squared error and memoryless sources, states that for arbitrary sources with finite differential entropy,

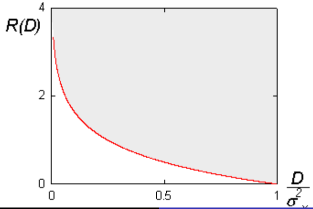
  $R(D) \geq h(X) - h(D)$

  where $h(D)$ is the entropy of a Gaussian random variable with variance D. This lower bound is extensible to sources with memory and other distortion measures
- One important feature of the SLB is that it is asymptotically tight in the high distortion regime for a wide class of sources and in some occasions, it actually coincides with the ratedistortion function
- Shannon Lower Bounds can generally be found if the distortion between any two numbers can be expressed as a function of the difference between the value of these two numbers

Rate Distortion: Memoryless (independent) Gaussian source

If we assume that $P_X(x)$ is Gaussian with variance σ^2, and if we assume that successive samples of the signal X are stochastically independent (or, if your like, the source is memoryless, or the signal is uncorrelated), we find the following analytical expression for the ratedistortion function:

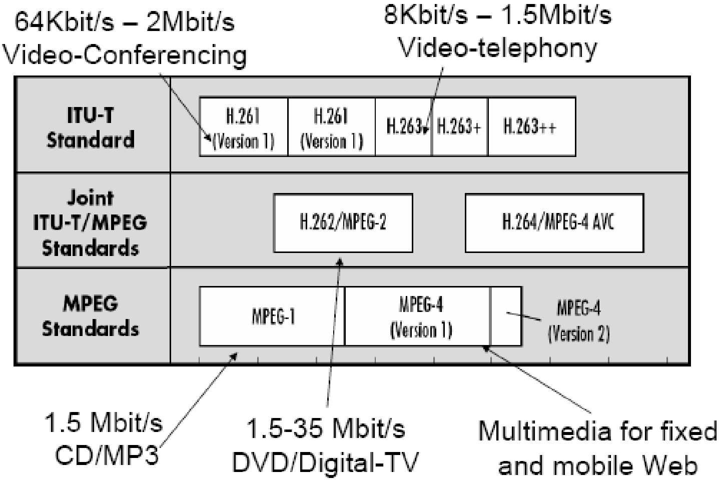
$$R(D) = \begin{cases} \frac{1}{2}\log_2(\sigma_x^2/D), & \text{if } D \leq \sigma_x^2 \\ 0, & \text{if } D > \sigma_x^2 \end{cases}$$



Rate Distortion

- Ratedistortion theory tell us that no compression system exists that performs outside the gray area. The closer a practical compression system is to the red (lower) bound, the better it performs
- As a general rule, this bound can only be attained by increasing the coding block length parameter. Nevertheless, even at unit blocklengths one can often find good (scalar) quantizers that operate at distances from the ratedistortion function that are practically relevant
- This ratedistortion function holds only for Gaussian memoryless sources
- It is known that the Gaussian source is the most “difficult” source to encode: for a given mean square error, it requires the greatest number of bits
- The performance of a practical compression system working

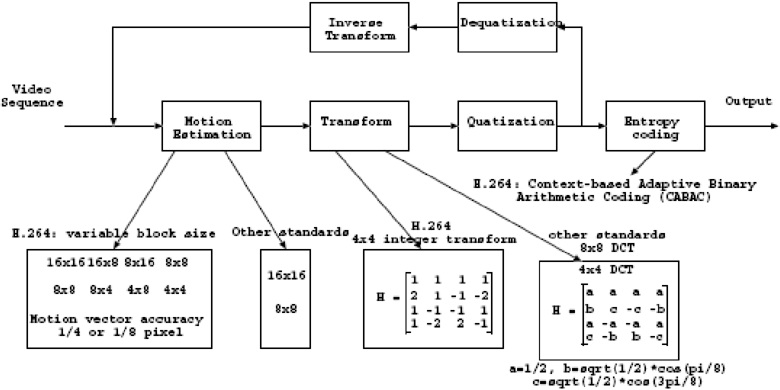
Joint Video Standards Overview



Video Coding Outline

- Motion Estimation (skim)
- DCT and Integer Transform (skim)
- Quantization (skim)
- Entropy Coding (important)

Video Coding Framework



Video Sequence

Motion Estimation

Transform

Quatization

Entropy coding

Output

Inverse Transform

Dequatization

H.264: variable block size

16x16 16x8 8x16 8x8

8x8 8x4 4x8 4x4

Motion vector accuracy 1/4 or 1/8 pixel

Other standards

16x16

8x8

H.264 4x4 integer transform

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

H.264: Context-based Adaptive Binary Arithmetic Coding (CABAC)

other standards 8x8 DCT

4x4 DCT

$$H = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix}$$

a=1/2, b=sqrt(1/2)*cos(pi/8)
c=sqrt(1/2)*cos(3pi/8)

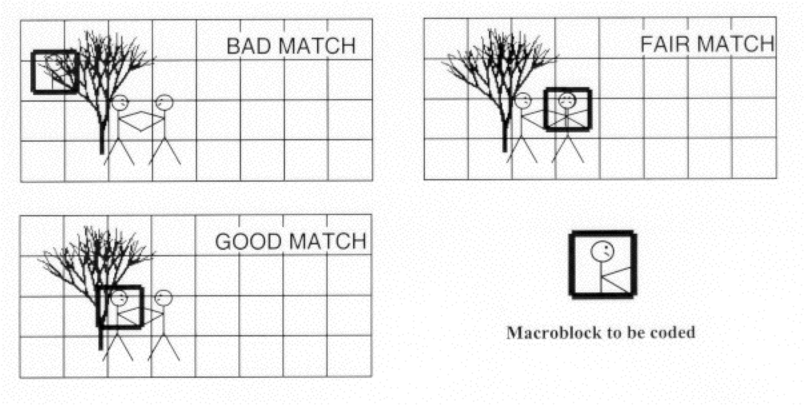
Motion Estimation I

- Motion Estimation (ME) is important since it can achieve significant compression by exploiting temporal redundancy in a video sequence
- Unfortunately it is also the most computationally intensive function of the encoding process
- The image is divided into Macro-Blocks (MB) and for each MB, a similar one is chosen in a reference frame, minimizing a distortion measure
- The best match found represents the predicted MB; displacement from the original MB to the best match gives the so-called Motion Vector (MV)
- Only the MV and the residual (i.e. the difference between the original MB and the predicted MB) need to be encoded and transmitted

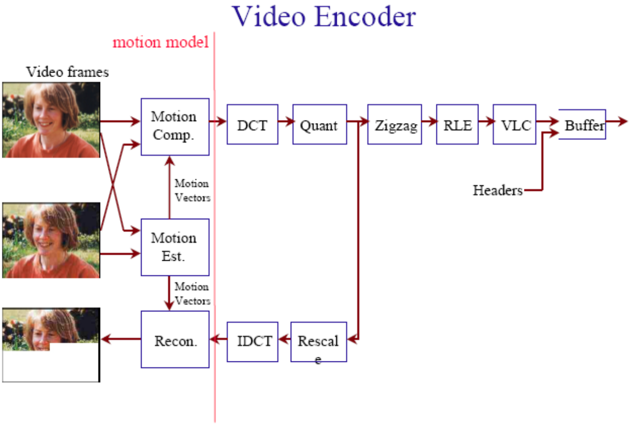
Motion Estimation II

- The distortion measure is the Sum of Absolute Differences: $SAD(d_x, d_y) = \sum_{m,n=0}^{N-1} |I_t(x+m, y+n) - I_{t-k}(x+d_x+m, y+d_y+n)|$
- (d_x, d_y) represents the MV components, $I_t(x, y)$ the luminance value in frame t at coordinates (x, y)

Motion Prediction

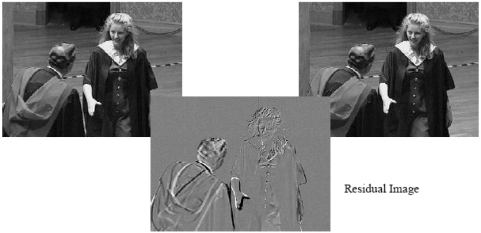


Motion Estimation



Motion Estimation

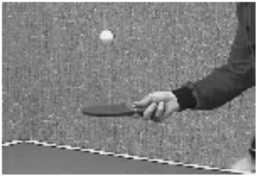
- The amount of data to be coded can be reduced significantly if the previous frame is subtracted from the current frame:



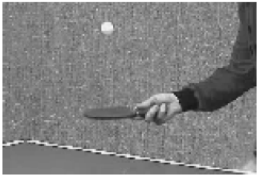
Motion Estimation

- Process 16x16 luminance samples at a time (“macroblock”)
  - Compare with neighboring areas in previous frame
  - Find closest matching area
  - prediction reference
- Calculate offset between current macroblock and prediction reference area
- motion vector

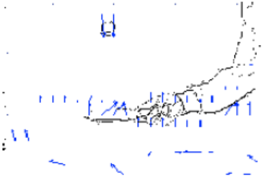
Motion Estimation



frame 1



frame 2



motion vectors

Motion Compensation

- Subtract the reference area from the current macroblock yielding a difference macroblock
- Encode the difference macroblock with an image encoder
- If motion estimation was effective little data left in difference macroblock and more efficient compression

DFT I

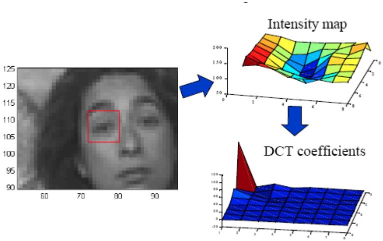
- The sequence of N complex numbers $x_0, ..., x_{N-1}$ is transformed into the sequence of N complex numbers $X_0, ..., X_{N-1}$ by the DFT according to the formula:
  $$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} \qquad k = 0, \dots, N-1$$
  - e is the base of the natural logarithm
  - i is the imaginary unit ($i^2 = -1$)
- Many of the properties of the DFT only depend on the fact that $e^{-\frac{2\pi i}{N}}$ is a primitive root of unity, sometimes denoted ω_N or W_N (so that $\omega_N^N = 1$)
  - Such properties include the completeness, orthogonality, Plancherel/Parseval, periodicity, shift, convolution, and unitarity properties below, as well as many FFT algorithms

DFT II

- A simple description of these equations is that the complex numbers X_k represent the amplitude and phase of the different sinusoidal components of the input "signal" x_n
- The DFT computes the X_k from the x_n, while the IDFT shows how to compute the x_n as a sum of sinusoidal components $X_k \exp(2\pi ikn/N)/N$ with frequency k/N cycles per sample
- By writing the equations in this form, we are making extensive use of Euler's formula to express sinusoids in terms of complex exponentials, which are much easier to manipulate

DCT and Integer Transform

- Transform each block of 8x8 samples into a block of 8 × 8 spatial frequency coefficients
  - energy tends to be concentrated into a few significant coefficients
  - other coefficients are close to zero and insignificant



DCT and Integer Transform

- Transform each block of 8x8 samples into a block of 8×8 spatial frequency coefficients
  - Energy tends to be concentrated into a few significant coefficients
  - Other coefficients are close to zero and insignificant
- $X_{k_1,k_2} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1,n_2} \cos\left[\frac{\pi}{N_1}\left(n_1+\frac{1}{2}\right)k_1\right] \cos\left[\frac{\pi}{N_2}\left(n_2+\frac{1}{2}\right)k_2\right]$

H.264 Transform

- 4x4 DCT of an input array X is given by:

$$Y = AXA^T = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} X \begin{bmatrix} a & b & a & c \\ a & c & -a & -b \\ a & -c & -a & b \\ a & -b & a & -c \end{bmatrix}$$

where $a = 1/2$, $b = \sqrt{1/2}\cos(\pi/8)$,$c = \sqrt{1/2}\cos(3\pi/8)$

- $Y = AXA^T =$

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{bmatrix} X \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -1 & -1 \\ 1 & -d & -1 & 1 \\ 1 & -1 & 1 & -d \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix}$$

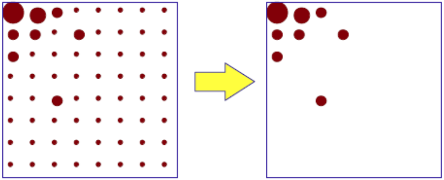
where a, b area already defined $d = c/b \approx 0.414 \rightarrow -.5$ and $\otimes$ indicates simple element-by-element

multiplication

- To ensure the orthogonality, b is modified so that: $a = 1/2$, $b = \sqrt{2/5}$, $d = 1/2$

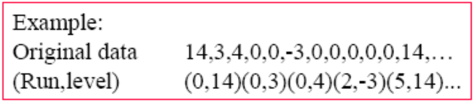
Quantization

- Divide each DCT coefficient by an integer, discard remainder
- Result: loss of precision
- Typically, a few non-zero coefficients are left



Prepare for entropy coding

- Encode each coefficient value as a (run,level) pair
  - run = number of zeros preceding value
  - level = non-zero value
- Usually, the block data is reduced to a short sequence of (run,level) pairs.
- This is now easy to compress using an entropy encoder

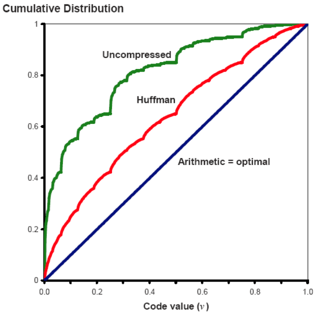


Example:

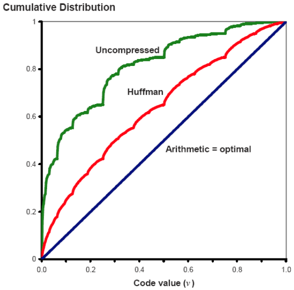
Original data	14,3,4,0,0,-3,0,0,0,0,0,14,…
(Run,level)	(0,14)(0,3)(0,4)(2,-3)(5,14)...

Entropy Coding

- The efficiency of a compression method may be analyzed by considering the distribution of the code values it produces
- From Shannon's information theory, we know that, if a coding method is optimal, then the cumulative distribution of its code values has to be a straight line from point (0, 0) to point (1, 1)



Entropy Coding



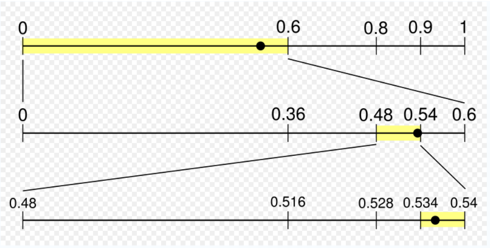
- The straight-line distribution means that there is no statistical dependence or redundancy left in the compressed sequences, and consequently its code values are uniformly distributed on the interval [0, 1)
- Essential for understanding of how arithmetic coding works
- Code values are an integral part of the arithmetic encoding/decoding

Arithmetic Coding

- The code value ν of a compressed data sequence is the real number with fractional digits equal to the sequence's symbols. We can convert sequences to code values by simply adding "0." to the beginning of a coded sequence, and then interpreting the result as a number in base-D notation, where D is the number of symbols in the coded sequence alphabet
- For example, if a coding method generates the sequence of bits 0011000101100, then we have
  - Code sequence $d = [0011000101100]$
  - Code value $\nu = 0.0011000101100_2 = 0.19287109375$
- where the "2" subscript denotes base-2 notation

Arithmetic Coding

- This construction creates a convenient mapping between infinite sequences of symbols from a D-symbol alphabet and real numbers in the interval [0, 1), where any data sequence can be represented by a real number, and vice-versa

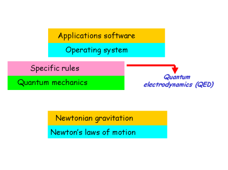


- Fundamentally, the arithmetic encoding process consists of creating a sequence of nested intervals in the form $\Phi_k(S) = [\alpha_k, \beta_k), k = 0, 1, ..., N$, where S is the source data sequence, α_k and β_k are real numbers such that $0 \leq \alpha_k < \alpha_{k+1}$, and $\beta_{k+1} \leq \beta_k \leq 1$

What is Quantum Mechanics?

- It is a framework for the development of physical theories
- It is not a complete physical theory in its own right

Consider the analogy



- Quantum Electrodynamics (QED) is an example of “Specific Rules”
    - Describes interaction of electrons and photons
- Quantum Mechanics (QM) consists of four mathematical postulates which lay the ground rules for our description of the world.

How Successful is Quantum Mechanics?

- It is unbelievably successful
- Not just for the small stuff
- QM crucial to explain why stars shine, how the Universe formed, and the stability of matter
- No deviations from quantum mechanics are known
- Most physicists believe that any "theory of everything" will be a quantum mechanical theory
- The "measurement problem" remains to be clarified
- Attempts to describe gravitation in the framework of quantum mechanics have (so far) failed

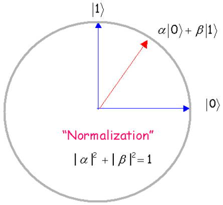
The Structure of Quantum Mechanics

- Linear Algebra
  - We are all matrices
- Dirac notation $\langle\psi|$,$\langle\phi|$,$\langle A\rangle$
- Four postulates of quantum mechanics
  1. How to describe quantum states of a closed system; "State vectors" and "state space"
  2. How to describe quantum dynamics; "unitary evolution"
  3. How to describe measurements of a quantum system; "projective measurements"
  4. How to describe quantum state of a composite system; "tensor products"

Example: qubits

- Two level quantum systems
- Photons, electron spin, nuclear spin, etc...
- $|0\rangle$ and $|1\rangle$ are the computational basis states

Two Level Quantum System


$|1\rangle$
$\alpha|0\rangle + \beta|1\rangle$
$|0\rangle$
"Normalization"
$|\alpha|^2 + |\beta|^2 = 1$

Cheat Sheet

z^*	Complex conjugate of the complex number z, $(1+i)^* = 1 - i$
$\|\psi\rangle$	Vector AKA *ket*
$\langle\psi\|$	Vector dual to $\|\psi\rangle$ AKA *bra*
$\langle\phi\|\psi\rangle$	Inner product between the vectors $\|\phi\rangle$ and $\|\psi\rangle$ (vector product)
$\|\phi\rangle \otimes \|\psi\rangle$	Tensor product of the vectors $\|\phi\rangle$ and $\|\psi\rangle$
$\|\phi\rangle\|\psi\rangle$	Alternative notation for tensor product
A^*	Complex conjugate of the A matrix
A^T	Transpose of the A matrix
$A^\dagger$	Hermitian conjugate or adjoint of the A matrix, $A^\dagger = (A^T)^*$
	$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^\dagger = \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix}$
$\langle\phi\|A\|\psi\rangle$	Inner product between $\|\phi\rangle$ and $A\|\psi\rangle$
	Also, inner product between $A^\dagger\|\phi\rangle$ and $\|\psi\rangle$

Postulate 1: Rough Form

- Associated to any quantum system is a complex vector space known as state space.
- The state of a closed quantum system is a unit vector in state space.
- Example: we will work mainly with qubits, which have state space C^2.

  $$\alpha\,|0\rangle + \beta\,|1\rangle \equiv \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$
- Quantum mechanics does not prescribe the state spaces of specific systems, such as electrons. Thats the job of a physical theory like quantum electrodynamics.

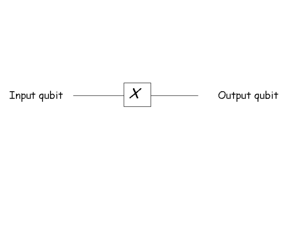
Conventions

- Vectors are written as $|\psi\rangle \equiv \bar{\psi}$
- This is the ket notation
- We will assume that our physical systems have finite dimensional state spaces
- $|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle + \alpha_2 |2\rangle \ldots \alpha_{d-1} |d-1\rangle = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{d-1} \end{bmatrix}$
- Qudit in C^d

Dynamics: Quantum Logic Gates

$$X|0\rangle = |1\rangle$$
$$X|1\rangle = |0\rangle$$

Quantum NOT Gate



$$\alpha|0\rangle + \beta|1\rangle \rightarrow ?$$
$$\alpha|0\rangle + \beta|1\rangle \rightarrow \alpha|1\rangle + \beta|0\rangle$$

Matrix representation

$$X = \begin{array}{ccc} & |0\rangle & |1\rangle \\ |0\rangle & 0 & 1 \\ |1\rangle & 1 & 0 \end{array}$$

General dynamics of a closed quantum system (including logic gates) can be represented as a unitary matrix.

Unitary Matrices

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Hermitian conjugation; taking the adjoint

$$A^\dagger = (A^*)^T = \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix}$$

A is said to be unitary if $AA^\dagger = A^\dagger A = I$

We usually write unitary matrices as U

Example

$$XX^\dagger = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

Nomenclature

matrix = (linear) operator = (linear) transformation = (linear) map = quantum gate (modulo unitarity)

Postulate 2

The evolution of a closed quantum system is described by a unitary transformation

$$|\psi'\rangle = U|\psi\rangle$$

Why Unitaries?

Unitary maps are the only linear maps that preserve normalization $|\psi'\rangle = U|\psi\rangle$ implies $\| \, |\psi'\rangle \, \| = \| U |\psi\rangle \, \| = \| \, |\psi\rangle \, \| = 1$

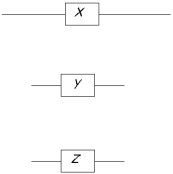
Exercise

Prove for yourself that unitary evolution preserves normalization

Pauli Gates

X gate (AKA σ_x or σ_1)
$X\left|0\right\rangle = \left|1\right\rangle$, $X\left|1\right\rangle = \left|0\right\rangle$,
$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$
Y gate (AKA σ_y or σ_2)
$Y\left|0\right\rangle = i\left|1\right\rangle$, $Y\left|1\right\rangle = -i\left|0\right\rangle$,
$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$
Z gate (AKA σ_z or σ_3)
$Z\left|0\right\rangle = \left|0\right\rangle$, $Z\left|1\right\rangle = -\left|1\right\rangle$,
$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Notation: $\sigma_0 = I$

Pauli Gates



Exercise

Prove that $XY = iZ$

Exercise

Prove that $X^2 = Y^2 = Z^2 = I$

Goals

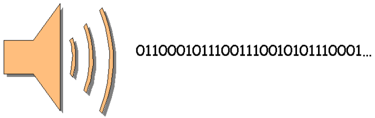
- Goal 1: Definitions and basic examples
  Give definitions of the entropy, both classical and quantum, and to work through some examples
- Goal 2: To explain data compression
  Explain data compression, both classical and quantum, and its connection with entropy (data compression has some extremely interesting connections with physics)
- Goal 3: Other properties of entropy
  Explain some of the basic properties of entropy, which has application to entanglement, quantum error-correction, and quantum communication (in last few lectures)

What is an information source?

011000101110011100101011100011101001011101000

- We need a simple “toy model” of an information source
- The model might not be realistic, but it should give rise to a theory of information that can be applied to realistic situations

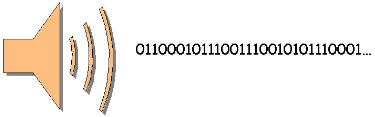
Discrete iid sources



- Definition: Each output from a discrete information source comes from a finite set
- We will mostly be concerned with the case where the alphabet consists of 0 and 1

- More generally, there is no loss of generality in supposing that the alphabet is $0, \ldots, n-1$
- Independent and identically distributed – if each output from the source is independent of other outputs from the source, and furthermore each output has the same distribution

Discrete iid sources



- We will model sources using a probability distribution for the output of the source

Example

A sequence of coin tosses of a biased coin with probability p of heads, and $1 - p$ of tails.

More generally, the distribution on alphabet of symbols is denoted

$p_0, p_1, \ldots, p_n$

What other sources are discrete iid?

- Most interesting sources are not:

  > *"Sing, goddess, the rage of Achilles the son of Peleus, the destructive rage that sent countless pains on the Achaeans..."*

- Reason: correlations
  The reason is that most sources of information show correlations between different outputs of the source. In English text, for example, certain letter combinations, like "th" and "wh" appear far more frequently than you would expect if the letters were all independent of one another
- However, lots of sources can be approximated as iid - even with English text this is not a bad approximation
- Many sources can be described as stationary, ergodic sequences of random variables, and similar results apply

Research problem

Find a good quantum analogue of "stationary, ergodic sources" for, and extend quantum information theory to those sources. (Quantum Shannon-Macmillan-Breiman theorem?)

Shannon-Macmillan-Breiman

- For an iid source X_k with random variable X and distribution $p(x)$:

$$-\frac{1}{n}\log p(X) \rightarrow H(X) \tag{2}$$

as $n \rightarrow \infty$ where X_k must be stationary

- The Shannon-McMillan-Breiman Theorem:

$$Pr(-\lim_{n\rightarrow\infty}\log Pr(X) = H) = 1 \tag{3}$$

if X_k is stationary and ergodic

A process is ergodic iff 'time averages' over a single realization of the process converge in mean square to the corresponding 'ensemble averages' over many realizations.

How can we quantify the rate at which information is being produced by a source?

- Two broad approaches
  - Axiomatic approach: Write down desirable axioms which a measure of information "should" obey and find such a measure (unfruitful)
  - Operational approach: Based on the "fundamental program" of information science (more promising)

Recall "the fundamental program"

The first step The first step of the program is to identify a physical process, like energy, time, bits, space, or perhaps shared entangled pairs, that can be used to do information processing

The second step The second step is to identify an information processing task. In a classical context that might be something like data compression. In both classical and quantum contexts it could be information transmission

The third step The third step is to identify a criterion for successful completion of the information processing task

The question Once we've done all three of these things we can ask the basic question of information science, how much of 1 is needed to do 2, while satisfying 3?

How many bits are needed to store the output of the source so the output can be reliably recovered?

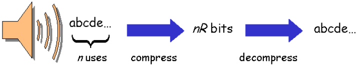
Historical origin of data compression

*"He can compress
the most words into
the smallest ideas of
any man I ever met."*

How a similar debate is playing out now with entanglement measures

- These philosophical issues regarding a definition of the measure of information are similar to the debate now going on in the research community about how to define measures of the amount of entanglement present in a quantum state
- Some people advocate an axiomatic approach, while others advocate an operational approach, and still others are advocating a combination of the approaches

Data compression



- What is the minimal value of R that allows reliable decompression?
- We will define the minimal value to be the information content of the source

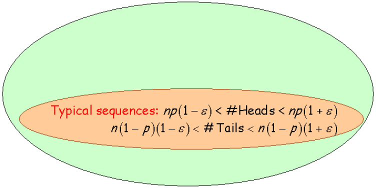
Theorem (Shannon's noiseless channel coding theorem)

The minimal achievable value of R is given by the Shannon entropy of the source distribution, $H(X) \equiv H(p_x) \equiv -\sum_x p_x \log(p_x)$

where logarithms are taken to base two.

Data compression

- Suppose we flip coins, getting heads with probability p, and tails with probability $1 - p$
- For large values of n, it is very likely that we will get roughly np heads, and $n(1 - p)$ tails
- A typical sequence is one such that the number of heads is between $np(1 - \epsilon)$ and $np(1 + \epsilon)$

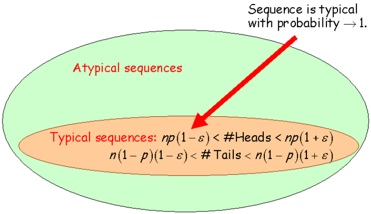


Typical sequences: $np(1-\varepsilon) < \#\text{Heads} < np(1+\varepsilon)$

$n(1-p)(1-\varepsilon) < \#\text{Tails} < n(1-p)(1+\varepsilon)$

Data compression

- x is the random variable for a bit sequence


Sequence is typical
with probability → 1.
Atypical sequences
Typical sequences: $np(1-\varepsilon) < \#\text{Heads} < np(1+\varepsilon)$
$n(1-p)(1-\varepsilon) < \#\text{Tails} < n(1-p)(1+\varepsilon)$

- $p^{np(1+\epsilon)}(1-p)^{n(1-p)(1+\epsilon)} < Pr(x) < p^{np(1-\epsilon)}(1-p)^{n(1-p)(1-\epsilon)}$
- $Pr(x) \approx 2^{np\log p + n(1-p)\log(1-p)} \approx 2^{-nH(p,1-p)}$
- $\#$ typical sequence $\approx 2^{nH(p,1-p)}$

Data compression: The algorithm

- The two critical facts
  - Sequence is typical with probability → 1
  - $\#$ typical sequences $\approx 2^{nH(p,1-p)}$
- In principle it is possible to construct a lookup table containing an indexed list of all $2^{nH(p,1-p)}$ typical sequences

```
Let $y$ be the source output
If $y$ is atypical then
send the bit 0 ($n + 1$ bits) and then the bit string $y$
else
send 1 and the index of $y$ ($nH(p, 1 - p) + 1$ bits) in the lookup
table
```

- On average, only $H(p, 1 - p)$ bits were required to store the compressed string, per use of the source

Variants on the data compression algorithm

- Our algorithm is for large n, gives variable-length output that achieves the Shannon entropy on average
- The algorithm never makes an error in recovery
- Algorithms for small n can be designed that do almost as well

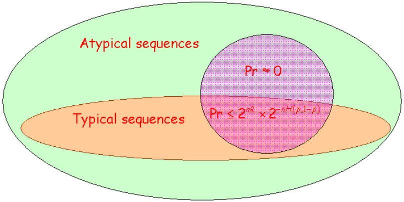
Fixed-length compression

```
Let $y$ be the source output
If $y$ is atypical then
send $(nH(p, 1-p) + 1)$ 0's
else
send 1 and the index of $y$ in the lookup table
```

- Errors must always occur in a fixed-length scheme, but it does work with probability approaching one
- Such a scheme will only be able to distinguish between 2^{nR} possible source outputs, yet a source may have more than this number of possible

Why it’s impossible to compress below the Shannon rate

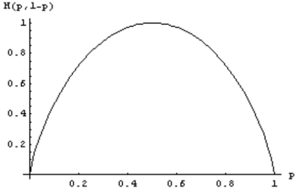
- Suppose $R < H(p, 1-p) \rightarrow Pr(\text{no loss}) \leq 2^{n(R-H(p,1-p))} \longrightarrow 0$
- At most 2^{nR} sequences can be correctly compressed and then decompressed by a fixed-length scheme of rate R



Basic properties of entropy

- $H(X) \equiv H(p_x) \equiv -\sum_x p_x \log(p_x)$
- $0 \log 0 \equiv 0$
  - If you take the limit of $x \log x$ as x goes to zero, you get zero
  - If letter of the alphabet occurs with probability zero, then clearly the information content of that source should not be affected by the presence or absence of the letter in the alphabet

- The entropy is non-negative and ranges between 0 and $\log(d)$ where d is the number of letters in the alphabet used by the source
- $H(p) \equiv H(p, 1-p)$ is known as the binary entropy



Recall Kolmogorov Complexity

When the minimum is obtained The minimum is obtained when the source is producing just a single letter, over and over again, with probability one (no need to compress this information) - this string contains no information at all, beyond its length

When the maximum is obtained The maximum is obtained when the input distribution is completely uniform, that is, we know nothing at all about potential biases in the source

Why's this notion called entropy, anyway?

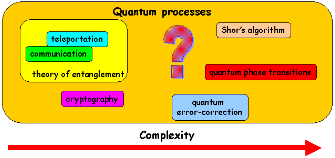
- Close mathematical correspondence for the formula for the entropy that Shannon gave, and the usual formula given in thermodynamics textbooks, based on Ludwig Boltzmanns magnificent formulation of statistical mechanics

From the American Heritage Book of English
Usage (1996):

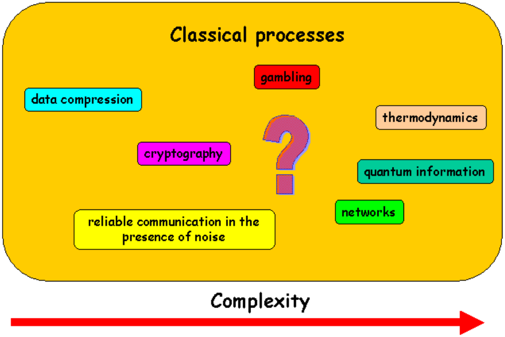
> *"When the American scientist Claude Shannon found that the mathematical formula of Boltzmann defined a useful quantity in information theory, he hesitated to name this newly discovered quantity entropy because of its philosophical baggage.*
>
> *The mathematician John Von [sic] Neumann encouraged Shannon to go ahead with the name entropy, however, since'no one knows what entropy is, so in a debate you will always have the advantage."*

What else can be done with Shannon entropy?

1. Identify a physical resource - energy, time, bits, space, entanglement
2. Identify an information processing task - data compression, information transmission, teleportation
3. Identify a criterion for success
4. How much of 1 do I need to achieve 2, while satisfying 3?

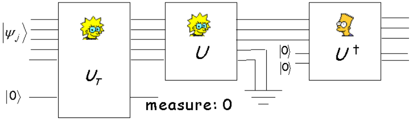


What else can be done with Shannon entropy?

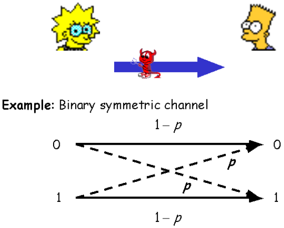


Reliable communication in the presence of noise

- A binary symmetric channel
- If a bit is input to the binary symmetric channel, then that bit is sent through correctly with probability $1 - p$, and flipped to the incorrect value with probability p



General model of a noisy channel



- The channel is described by conditional probabilities $p(y|x)$
- Example: for the binary symmetric channel:
  - $p(0|0) = 1 - p$
  - $p(1|0) = p$
  - $p(0|1) = p$
  - $p(1|1) = 1 - p$

Reliable communication in the presence of noise

- Channel capacity $\equiv$ $\dfrac{\text{maximal \# of message bits that can be reliably sent}}{\text{number of uses of channel}}$
- Mutual information
  - $X \rightarrow Y$
  - $H(X,Y) \equiv H(X) + H(Y) - H(X:Y)$ (remove information common to both)
  - $H(X:Y) \equiv H(X) + H(Y) - H(X,Y)$ (high mutual information means that we can recover the original material)

Shannon's noisy channel coding theorem

The capacity of a noisy channel is given by the expression capacity $= \max_{p_x} H(X:Y)$.

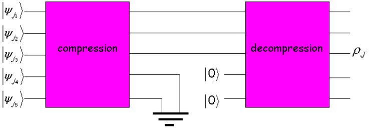
What is a quantum information source?

- Example: "Semiclassical coin toss"
  - $|0\rangle$ with probability $\frac{1}{2}$
  - $|1\rangle$ with probability $\frac{1}{2}$
- Example: "Quantum coin toss"
  - $|0\rangle$ with probability $\frac{1}{2}$
  - $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ with probability $\frac{1}{2}$

Theorem (General definition)

A quantum information source produces states $|\psi_j\rangle$ with probabilities p_j.

Quantum data compression



- $J \equiv (j_1, \ldots, j_n)$
- $p_J \equiv p_{J_1} \times \ldots \times p_{J_n}$
- $|\psi_J\rangle \equiv |\psi_{j_1}\rangle \ldots |\psi_{j_n}\rangle$
- $\bar{F} \equiv \sum_J p_J F(|\psi_J\rangle , \rho_J)$
- (Recall that $F \equiv \sqrt{\langle \psi_J \,|\, \rho_J \,|\, \psi_J \rangle}$)
- $\bar{F} \rightarrow 1$

Whats the best possible rate for quantum data compression?

(show why below... may need to introduce density matrix)

- Example: "Semiclassical coin toss"
  - $|0\rangle$ with probability $\frac{1}{2}$
  - $|1\rangle$ with probability $\frac{1}{2}$
  - Answer: $H(\frac{1}{2}) = 1$
- Example: "Quantum coin toss"
  - $|0\rangle$ with probability $\frac{1}{2}$
  - $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ with probability $\frac{1}{2}$
  - Answer: $H(\frac{1}{2}) = 1$? NO!
  - Answer: $H(\frac{1+1/\sqrt{2}}{2}) \approx 0.6$ ($|0\rangle$ appears with greater than $\frac{1}{2}$ probability)
- In general, we can do **better** than Shannon's rate $H(p_j)$

References

1 M. Nelson and J. Gailly. The Data Compression Book. M & T Books, 2nd edition, 1995.

2 J. R. Pierce. An Introduction to Information Theory: Symbols, Signals and Noise. Dover, 1980.

3 J. R. Pierce and A. M. Noll. Signals: The Science of Telecommunications. Number 32. Scientific American Library, 1990.

4 C. E. Shannon. A mathematical theory of communication. Bell System Technical Journal, 27:379 pp 423 and 623-656, July and October 1948. Originally Published in 2 parts, can be found on-line at many locations.

5 M. Li and P. Vitnyi. An Introduction to Kolmogorov Complexity and Its Applications. Springer, 2nd Edition, 1997.

6 Active Virtual Network Management Prediction: Complexity as a Framework for Prediction, Optimization, and Assurance (Introduction to Active Networks) by Stephen F. Bush, Proceedings of the 2002 DARPA Active Networks Conference and Exposition (DANCE 2002), IEEE Computer Society Press, pp. 534-553, ISBN 0-7695-1564-9, May 29-30, 2002, San Francisco, California, USA.

7 Shannon's 1956 paper

8 Lovász : "On the Shannon Capacity of a graph", IEEE Trans. Inf. Th., jan. 1979.

9 Körner and Orlitsky: "Zero-error information theory", IEEE Trans. Inf. Th., oct. 1998.

10 Iain E. G. Richardson, "Introduction to Image and Video Coding", 2002, www.vcodex.com.