16.548 Notes II
More Ways To Measure Information
How to Data Mine for Fun and Profit

Jay Weitzen

University of Massachusetts Lowell

Electrical Electrical and Computer Engineering

Module Contents

- Conditional Entropy
- Mutual Information and Information Gain (loss)
  - Introduction to Information theory and communication
- Shannon's Channel Coding Theorem

Comment

- Information theory discussed today applies to applications of data mining, data compression, and communication

Specific Conditional Entropy H(Y|X=v)

Suppose I'm trying to predict output Y and I have input X

X = College Major

Y = Likes "Gladiator"

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Let's assume this reflects the true probabilities

E.G. From this data we estimate

- *P(LikeG = Yes) = 0.5*
- *P(Major = Math & LikeG = No) = 0.25*
- *P(Major = Math) = 0.5*
- *P(LikeG = Yes | Major = History) = 0*

Note:

- *H(X) = 1.5*
- *H(Y) = 1*

Specific Conditional Entropy H(Y|X=v)

X = College Major

Y = Likes "Gladiator"

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Definition of Specific Conditional Entropy:

$H(Y|X=v)$ = **The entropy of** Y **among only those records in which** X **has value** v

Specific Conditional Entropy H(Y|X=v)

X = College Major

Y = Likes "Gladiator"

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Definition of Specific Conditional Entropy:

$H(Y|X=v)$ = **The entropy of** Y **among only those records in which** X **has value** v

Example:

- $H(Y|X=Math) = 1$
- $H(Y|X=History) = 0$
- $H(Y|X=CS) = 0$

Definition: Conditional Entropy

$$
\begin{aligned}
H(Y|X) &\stackrel{\text{def}}{=} \sum_{x\in\mathcal{X}} p(x)\, H(Y|X=x) \\
&= -\sum_{x\in\mathcal{X}} p(x) \sum_{y\in\mathcal{Y}} p(y|x)\, \log\, p(y|x) \\
&= -\sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p(x,y)\, \log\, p(y|x) \\
&= -\sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p(x,y) \log\, p(y|x) \\
&= -\sum_{x\in\mathcal{X}, y\in\mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)}.
\end{aligned}
$$

Conditional Entropy H(Y|X)

X = College Major

Y = Likes "Gladiator"

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Definition of Conditional Entropy:

$H(Y|X)$ = The average specific conditional entropy of Y

= if you choose a record at random what will be the conditional entropy of Y, conditioned on that row's value of X

= Expected number of bits to transmit Y if both sides will know the value of X

$= \Sigma_j \, Prob(X=v_j) \, H(Y \mid X = v_j)$

Conditional Entropy

X = College Major

Y = Likes "Gladiator"

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Definition of Conditional Entropy:

$H(Y|X)$ = The average conditional entropy of Y

$= \Sigma_j Prob(X=v_j)\, H(Y \mid X = v_j)$

Example:

v_j	$Prob(X=v_j)$	$H(Y \mid X = v_j)$
Math	0.5	1
History	0.25	0
CS	0.25	0

$H(Y|X) = 0.5 * 1 + 0.25 * 0 + 0.25 * 0 = 0.5$

Information Gain (loss) (aka Mutual Information)

X = College Major

Y = Likes "Gladiator"

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Definition of Information Gain:

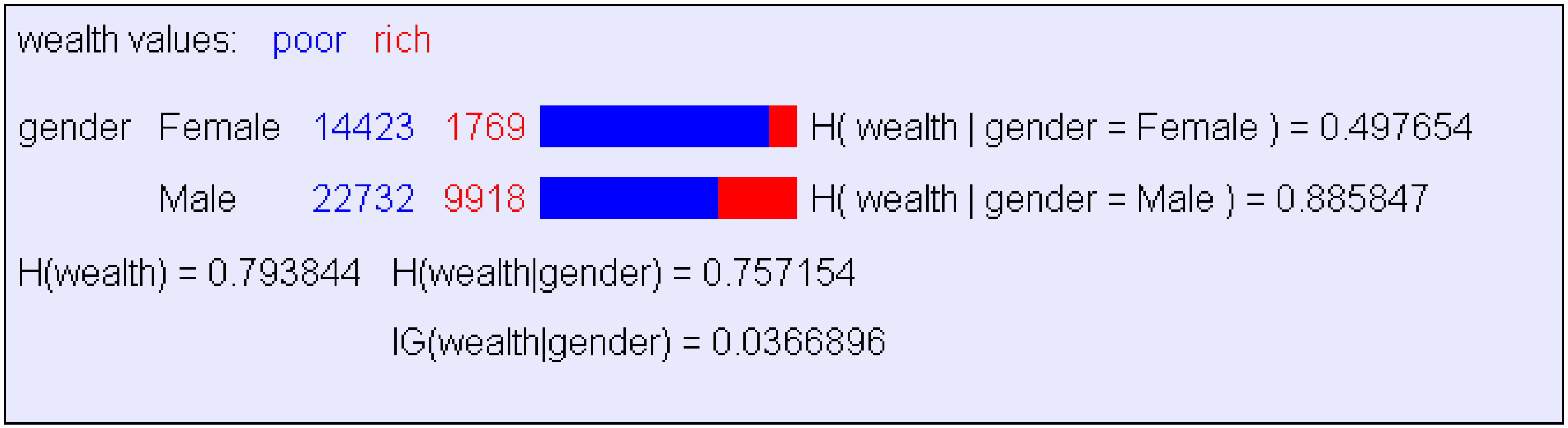
IG(Y|X) = **I must transmit** *Y.* **How many bits on average would it save me if both ends of the line knew** ***X*****?**

$IG(Y|X) = H(Y) - H(Y \mid X)$

Example:

- **H(Y) = 1**
- **H(Y|X) = 0.5**
- **Thus IG(Y|X) = 1 – 0.5 = 0.5**

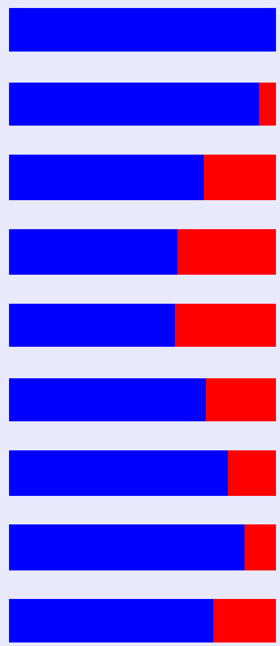
Information Gain Example



Another example

wealth values: poor rich

		poor	rich	
agegroup	10s	2507	3	H(wealth | agegroup = 10s) = 0.0133271
	20s	11262	743	H(wealth | agegroup = 20s) = 0.334906
	30s	9468	3461	H(wealth | agegroup = 30s) = 0.838134
	40s	6738	3986	H(wealth | agegroup = 40s) = 0.951961
	50s	4110	2509	H(wealth | agegroup = 50s) = 0.957376
	60s	2245	809	H(wealth | agegroup = 60s) = 0.834049
	70s	668	147	H(wealth | agegroup = 70s) = 0.680882
	80s	115	16	H(wealth | agegroup = 80s) = 0.535474
	90s	42	13	H(wealth | agegroup = 90s) = 0.788941



H(wealth) = 0.793844 H(wealth|agegroup) = 0.709463

IG(wealth|agegroup) = 0.0843813

Relative Information Gain

X = College Major

Y = Likes "Gladiator"

X	Y
Math	Yes
History	No
CS	Yes
Math	No
Math	No
CS	Yes
History	No
Math	Yes

Definition of Relative Information Gain:

$RIG(Y|X)$ = **I must transmit** Y**, what fraction of the bits on average would it save me if both ends of the line knew** X**?**

$$RIG(Y|X) = (H(Y) - H(Y | X))/ H(Y)$$

Example:

- $H(Y|X) = 0.5$
- $H(Y) = 1$
- Thus $IG(Y|X) = (1 - 0.5)/1 = 0.5$

What is Information Gain used for?

Suppose you are trying to predict whether someone is going live past 80 years. From historical data you might find…

- IG(LongLife | HairColor) = 0.01
- IG(LongLife | Smoker) = 0.2
- IG(LongLife | Gender) = 0.25
- IG(LongLife | LastDigitOfSSN) = 0.00001

IG tells you how interesting a 2-d contingency table is going to be.

Searching for High Info Gains

- Given something (e.g. wealth) you are trying to predict, it is easy to ask the computer to find which attribute has highest information gain for it.

wealth values: poor rich

relation	Husband	10870	8846	H(wealth | relation = Husband) = 0.992385
	Not_in_family	11307	1276	H(wealth | relation = Not_in_family) = 0.473439
	Other_relative	1454	52	H(wealth | relation = Other_relative) = 0.216617
	Own_child	7470	111	H(wealth | relation = Own_child) = 0.110192
	Unmarried	4816	309	H(wealth | relation = Unmarried) = 0.328606
	Wife	1238	1093	H(wealth | relation = Wife) = 0.997207

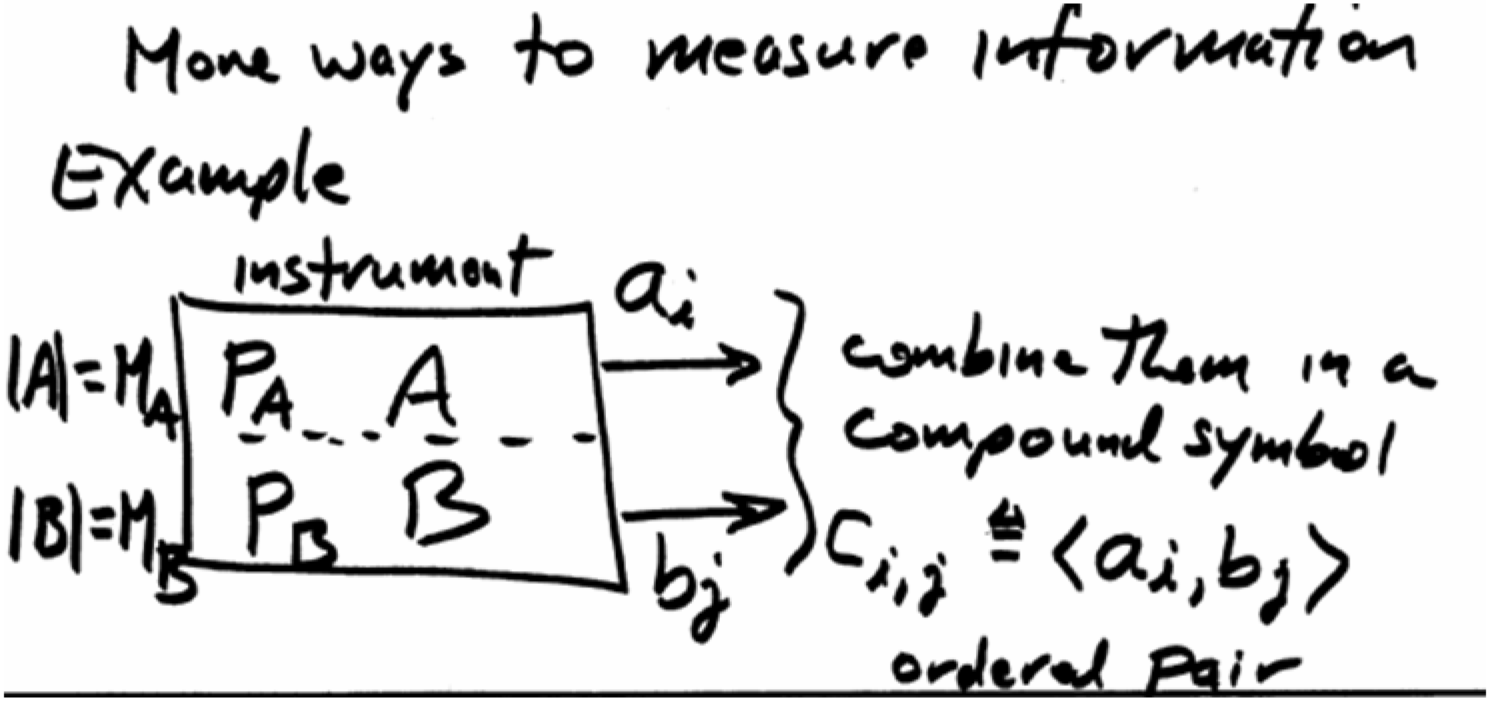
H(wealth) = 0.793844 H(wealth|relation) = 0.628421

IG(wealth|relation) = 0.165423

What Else is Conditional Entropy Used For

- It is used as a measure of uncertainty (noise) introduced by the channel
- To be derived over next few minutes

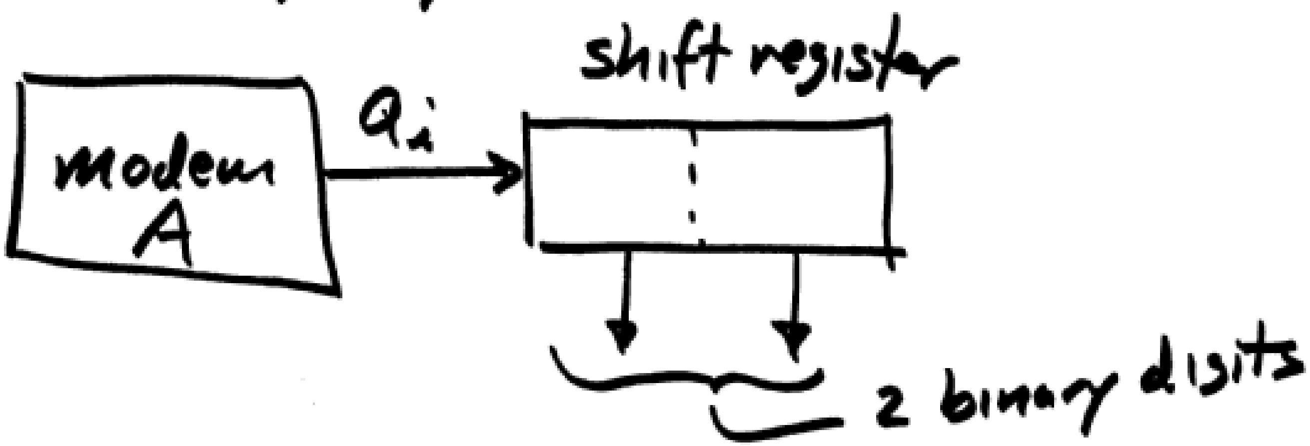
Joint Information or Dependency



University of Idaho

(2)

another example



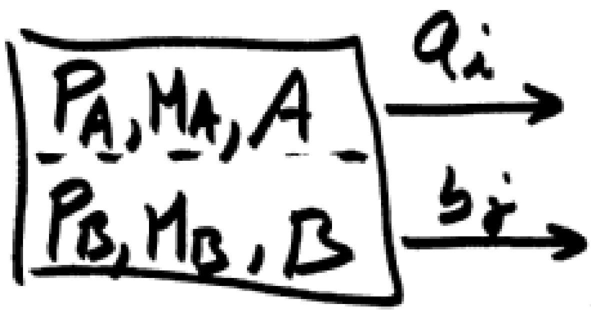
let a_i represent the digit at "even" times

b_j represent the digit at "odd" times

$$C_{i,j} = \langle a_i, b_j \rangle \quad \text{a "word"}$$

University of Idaho

(3)



$$C_{i,j} = \langle a_i, b_j \rangle$$

if $C = \{C_{i,j}\}$, how much info (on the average) does a compound symbol "carry"?

$$H(C) = ?$$

University of Idaho $c_{i,j} = \langle a_i, b_j \rangle$

④

$$\Pr[c_{i,j}] \triangleq p_{i,j} = \Pr[a_i, b_j]$$

joint probability

recall

$$\sum_{\text{all } c_{i,j} \in C} p_{i,j} = 1 = \sum_{a_i \in A} \sum_{b_j \in B} p_{i,j}$$

$$\Pr[a_i, b_j] = \Pr[b_j, a_i]$$

$$p_{i,j} = \Pr[b_j | a_i] \cdot p(a_i) \triangleq p_{j|i} \cdot p_i$$

University of Idaho

(5)

Since $p_{i,j} = p_{j,i} = Pr[b_j, a_i]$

$$p_{i,j} = Pr[a_i | b_j] \cdot p(b_j)$$

$$\downarrow \qquad \downarrow$$

$$p_{i|j} \cdot p_j$$

University of Idaho

(6)

Probability example

2 fair coins : heads, tails

$Pr[H,H] = 1/4$

$Pr[H,T] = 1/4$

x $Pr[T,H] = 1/4$

x $Pr[T,T] = 1/4$

Suppose I toss the first coin and it comes up heads H_1, what about the Prob. for 2nd coin?

$p_2 = 1/2$

$$Pr[H_2 | H_1] \cdot Pr[H_1] = Pr[H,H] = 1/4$$

$$\underset{1/2}{} \quad \underset{1/2}{} \quad = 1/4$$

University of Idaho

⑦

$c_{i,j} = \langle a_i, b_j \rangle$

goal: find $H(C)$

use definition

$$H(C) = \sum_{\text{all } c_{i,j} \in C} p_{i,j} \cdot \log_2\left(\frac{1}{p_{i,j}}\right)$$

$$= \sum_{a_i \in A} \sum_{b_j \in B} p_{i,j} \log_2\left(\frac{1}{p_{i,j}}\right) \triangleq H(A,B)$$

↑ joint entropy

use $p_{i,j} = p_{j|i} \cdot p_i$

University of Idaho

(8)

$$H(C) = \sum_{a_i \in A} \sum_{b_j \in B} p_{i,j} \log_2\left(\frac{1}{p_{j|i} \cdot p_i}\right)$$

$\shortparallel$

$H(A,B)$

$$\log_2\left(\frac{1}{p_{j|i} \cdot p_i}\right) = \log_2\left(\frac{1}{p_{j|i}}\right) + \log_2\left(\frac{1}{p_i}\right)$$

University of Idaho

(9)

so

$$H(C) = \sum_{a_i \in A} \sum_{b_j \in B} p_{i,j} \log_2\left(\frac{1}{p_{j|i}}\right)$$

$$+ \sum_{a_i \in A} \sum_{b_j \in B} p_{i,j} \log_2\left(\frac{1}{p_i}\right) \Big\} H(A)$$

University of Idaho

(10)

$$p_{i,j} = p_{j|i} \cdot p_i$$

2nd term on pg 9 can be written

$$\sum_{a_i \in A} \sum_{b_j \in B} p_{j|i} \cdot \left(p_i \cdot \log_2 \left(\frac{1}{p_i} \right) \right)$$

$$= \underbrace{\sum_{a_i \in A} p_i \log_2 \left(\frac{1}{p_i} \right)}_{H(A)} \underbrace{\sum_{b_j \in B} p_{j|i}}_{= 1}$$

University of Idaho (11)

So

$$H(C) = H(A) + \underbrace{\sum_{a_i \in A} \sum_{b_j \in B} p_{i,j} \log_2\left(\frac{1}{p_{j|i}}\right)}$$

H(B|A) is the uncertainty remaining in B given knowledge of A

defined to be the "conditional entropy" H(B|A)

aka "equivocation"

$$0 \leq H(B|A) \leq H(B)$$

University of Idaho (13)

gathering together

$$H(C) = H(A,B) = H(A) + H(B|A)$$

since $Pr[a_i, b_j] = Pr[b_j, a_i]$

$$H(A,B) = H(B,A) = H(B) + H(A|B)$$

usually, $H(A) \neq H(B)$

$$H(A,B) = H(B,A) \Rightarrow H(B|A) \neq H(A|B)$$

University of Idaho

(13)

since $0 \leq H(B|A) \leq H(B)$

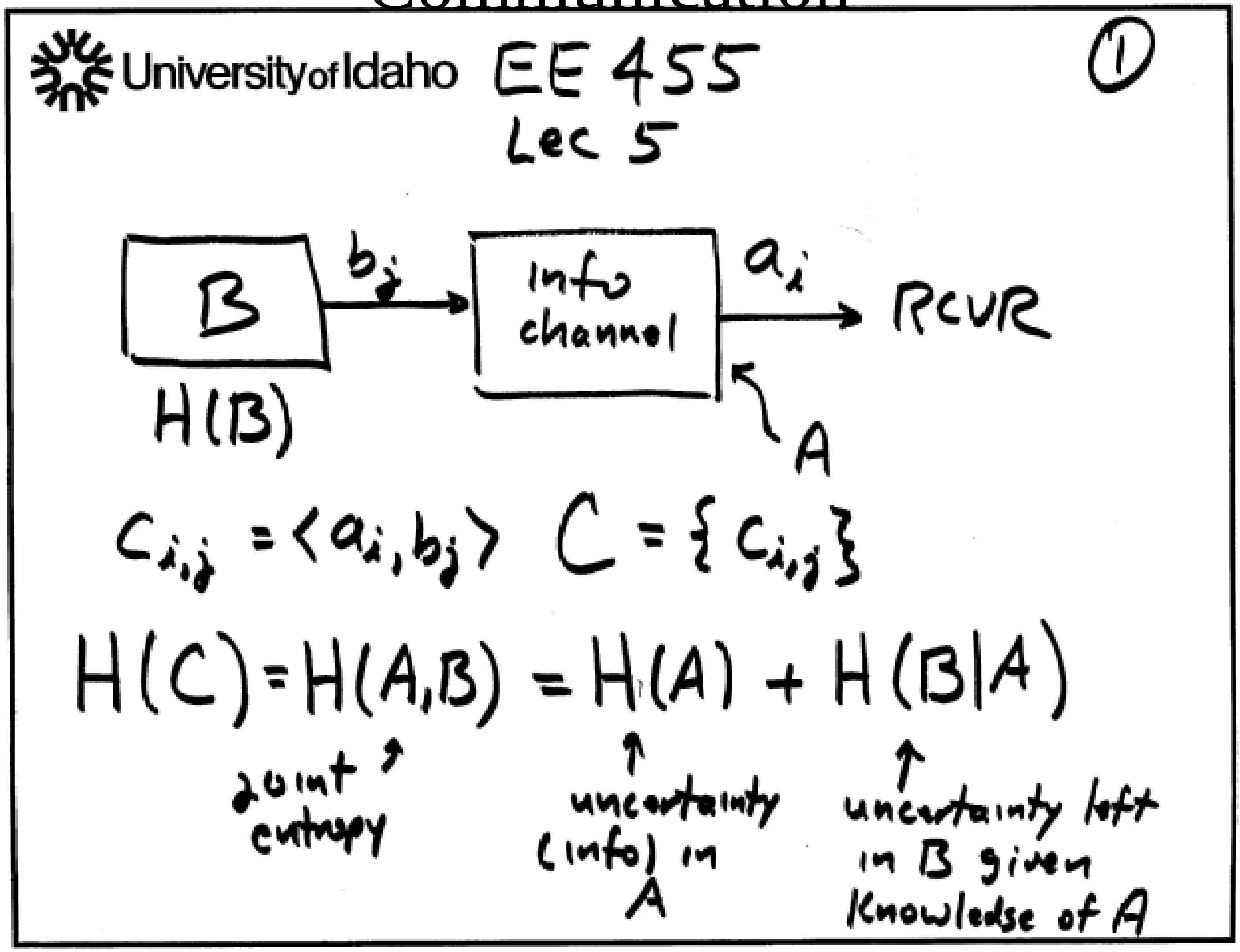
$$H(C) = H(A,B) \leq H(A) + H(B)$$

equality in this limit only occurs if

$H(B|A) = H(B) \Rightarrow$ A tells us nothing about B

$\Rightarrow$ A and B are statistically independent

Information Theory Applied to Communication



Key Slide: Definition of Mutual Information

University of Idaho

②

recall that $0 \leq H(B|A) \leq H(B)$

decrease in uncertainty in B given knowledge of A is

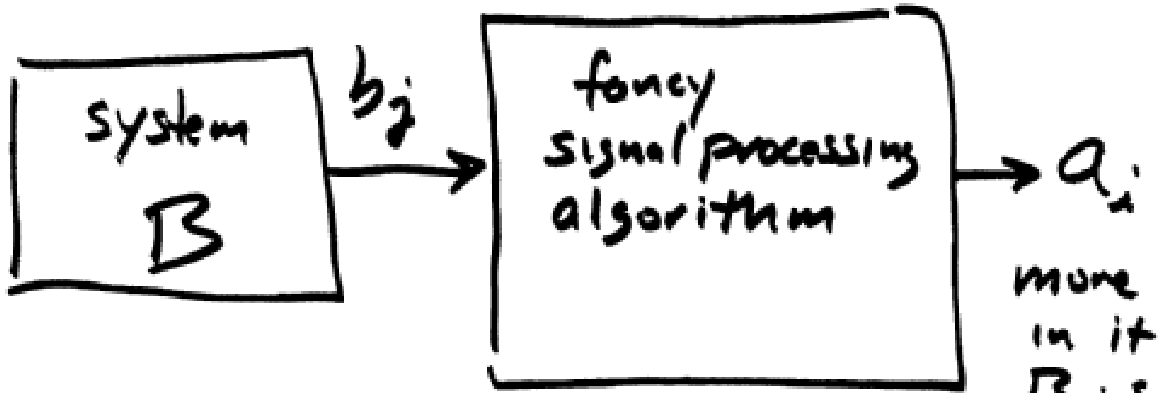
$$H(B) - H(B|A) \triangleq I(B;A)$$

↑ "mutual information"

if $I(B;A) < H(B) \Rightarrow$ info was <u>lost</u> in transmission

University of Idaho

(3)



more info in it than B; specific.

can you recover lost info?

info ≠ data

$$0 \leq H(B|A) \leq H(B)$$

side info theorem: fancy signal processing can not increase the information – once info is lost, it's gone for good.

University of Idaho

(5)

$$H(B|A) \leq H(B)$$

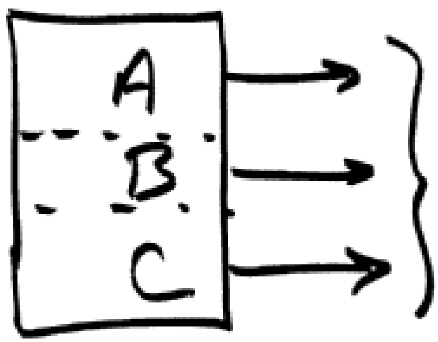
what about the "<" condition?

messed up by

- noise
- lossy data compression
- computer roundoff/truncation error

Hard decision vs Soft Decision

entropy algebra



$$d_{i,j,k} = \langle a_i, b_j, c_k \rangle$$

entropy (joint entropy) obeys a chain rule

$$\frac{df}{dt} = \frac{df}{dx} \cdot \frac{dx}{dt}$$

$$H(D) = H(A,B,C) = H(A) + H(B|A) + H(C|A,B)$$

University of Idaho

(7)

$$H(A,B,C,D) = H(A) + H(B|A) + H(C|A,B) + H(D|A,B,C)$$

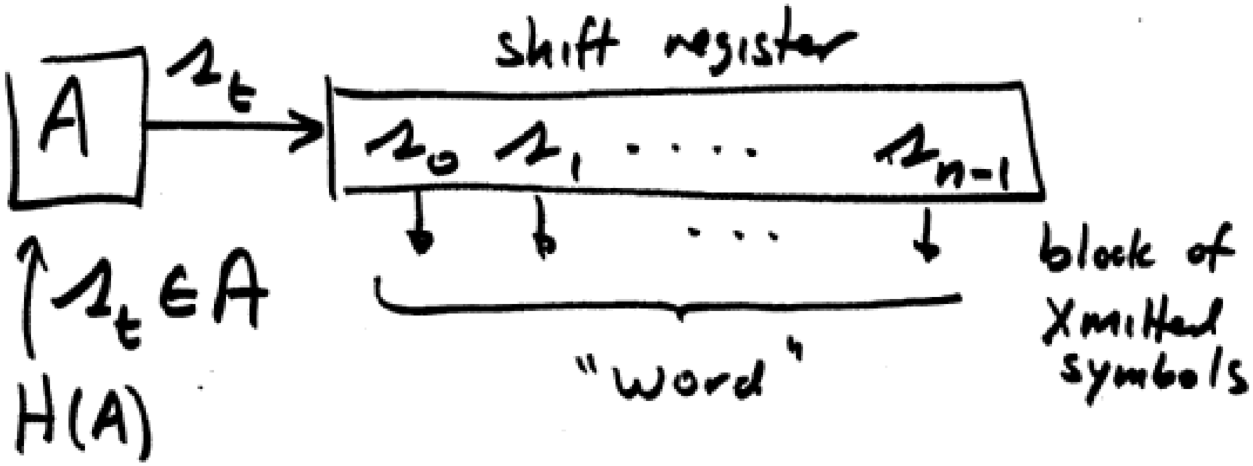
$$\|$$

$$H(B,C,A,D) \rightarrow H(B) + H(C|B) + H(A|B,C) + H(D|B,C,A)$$

University of Idaho

⑧

Let's get practical



The word is only a compound symbol, C

$$H(C) = H(A_0, A_1, \ldots, A_{n-1})$$

University of Idaho

(9)

from chain rule

$$H(C) = H(A_0, A_1, A_2, \ldots, A_{n-1})$$

$$= H(A_0) \leftarrow = H(A)$$

$$+ H(A_1 | A_0) \leftarrow \leq H(A)$$

$$+ H(A_2 | A_0, A_1) \leftarrow \leq H(A)$$

$$\vdots$$

$$+ \cdots$$

$$+ H(A_{n-1} | A_0, A_1, \ldots, A_{n-2}) \leftarrow \leq H(A)$$

$$H(C) \leq n \cdot H(A)$$

University of Idaho

(10)

when does $H(C) = n \cdot H(A)$?

This requires $H(A_1|A_0) = H(A)$

This requires that A_1, A_0 be statistically independent

So $H(C) = n H(A)$ iff the source is DMS

$$H(C) \leq n H(A) \Rightarrow \frac{H(C)}{n} \leq H(A)$$

University of Idaho

(11)

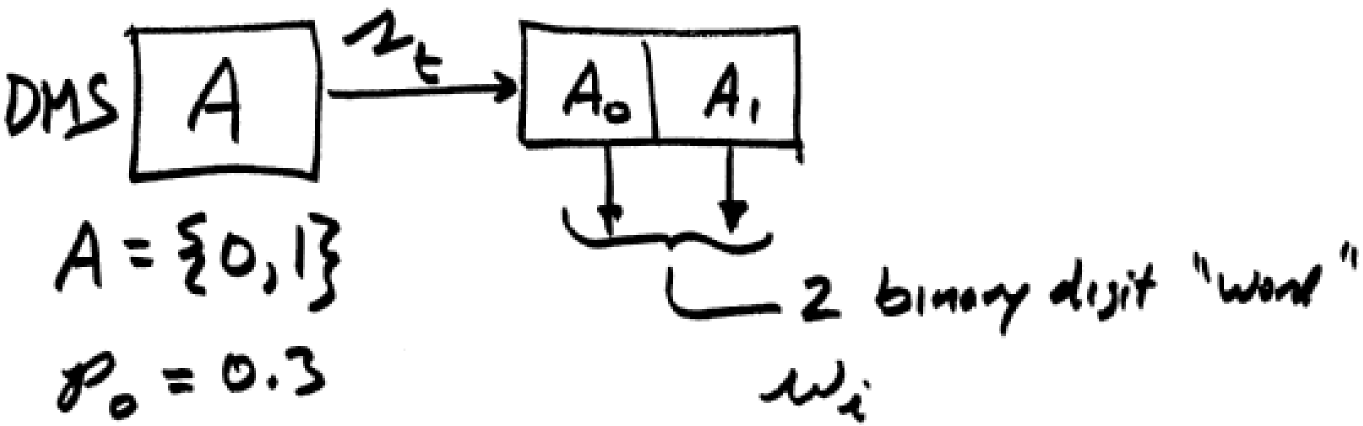
in English "q" is almost always followed by the letter "u"

in English alphabet, $H(A) \approx 4.1$ bits/letter

but $H(\text{English}) \approx 1.0$ to 1.5 bits/letter

$$\frac{H(A_0, A_1, \ldots, A_{n-1})}{n} \leq H(A)$$

example



$A = \{0, 1\}$

$p_0 = 0.3$

$p_1 = 0.7$

$$H(A) = .3 \log_2\left(\frac{1}{.3}\right) + .7 \log_2\left(\frac{1}{.7}\right)$$
$$= 0.8813 \text{ bits}$$

$$W = \{(0,0), (0,1), (1,0), (1,1)\}$$

$$H(W) = ? \quad \text{DMS} \Rightarrow H(W) = H(A) + \underbrace{H(A|A)}_{= H(A)} = 2H(A)$$

i	w_i	p_i
0	00	$(.3)(.3) = .09$
1	01	$(.3)(.7) = .21$
2	10	$(.7)(.3) = .21$
3	11	$(.7)(.7) = .49$

$$H(W) = \sum_{i=0}^{3} p_i \log_2\left(\frac{1}{p_i}\right) = 1.7626 = 2H(A)$$

data is 2 binary digits

$H(W) = 1.7626 < 2$

Applications of Information Theory: Compression

Shannon’s First Theorem: A.K.A Source Coding Theorem, A.K.A Compression Theorem

Shannon's source coding theorem

University of Idaho

(8)

Shannon's first theorem (aka the source coding theorem aka the noiseless capacity theorem)

if $\omega = \langle A_0, A_1, \ldots, A_{n-1} \rangle$

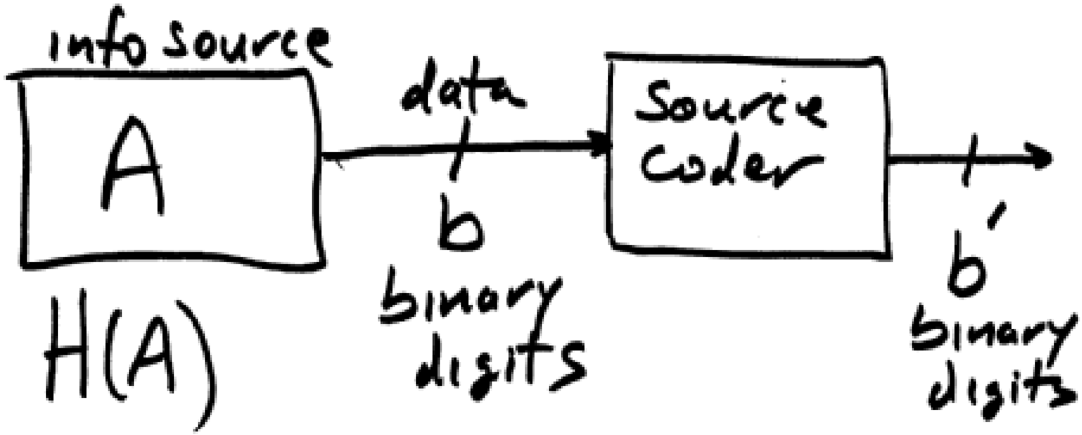
then there exists ($\exists$) an instantaneously decodable source code such

$$H(A_0, A_1, \ldots, A_{n-1}) \leq \bar{L} < H(A_0, A_1, \ldots, A_{n-1}) + 1$$

avg. codeword length as $\bar{l} \triangleq \frac{\bar{L}}{n}$ then

University of Idaho

(2)



if $b > H(A)$ then data representation is inefficient

$$\frac{H(A)}{b} \times 100\% = \%\text{ efficiency}$$

average $(b') < b$

typically, source codes have codewords that are variable length

University of Idaho

(4)

- Types of source codes

Huffman Codes
Arithmetic Codes
Dictionary Codes
- Dynamic Dictionary Codes
  - Lempel-Ziv codes

} lossless codes

JPEG
MPEG
⋮

} lossy codes

Shannon lossless source coding theorem is based on the concept of block coding. To illustrate this concept, we introduce a special information source in which the alphabet consists of only two letters:

$$\mathcal{A} = \{a, b\}.$$

Here, the letters `a' and `b' are equally likely to occur. However, given that `a' occurred in the previous character, the probability that `a' occurs again in the present character is 0.9. Similarly, given that `b' occurred in the previous character, the probability that `b' occurs again in the present character is 0.9. This is known as a binary symmetric Markov source.

An n-th order block code is just a mapping which assigns to each block of n consecutive characters a sequence of bits of varying length. The following examples illustrate this concept.

1. First-Order Block Code: Each character is mapped to a single bit.

B_1	$p(B_1)$	Codeword
a	0.5	0
b	0.5	1
R=1 bit/character		

An example:

Original Data:	a a a a a a a b b b b b b b b b b b b b a a a a
Compressed Data:	0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0

Note that 24 bits are used to represent 24 characters --- an average of 1 bit/character.

Rate of a Source Code

- The rates shown in the tables are calculated from

$$R = \frac{1}{n} \sum p(B_n) l(B_n) \quad \textbf{bits/sample},$$

where $l(B_n)$ is the length of the codeword for block B_n.

2^{nd} order Block Codes and Huffman Encoding

2. Second-Order Block Code: Pairs of characters are mapped to either one, two, or three bits.

B_2	$p(B_2)$	Codeword
aa	0.45	0
bb	0.45	10
ab	0.05	110
ba	0.05	111
R=0.825 bits/character		

An example:

Original Data:	a a	a a	a a	a b	b b	b b	b b	b b	b b	b b	a a	a a
Compressed Data:	0	0	0	110	10	10	10	10	10	10	0	0

Note that 20 bits are used to represent 24 characters --- an average of 0.83 bits/character.

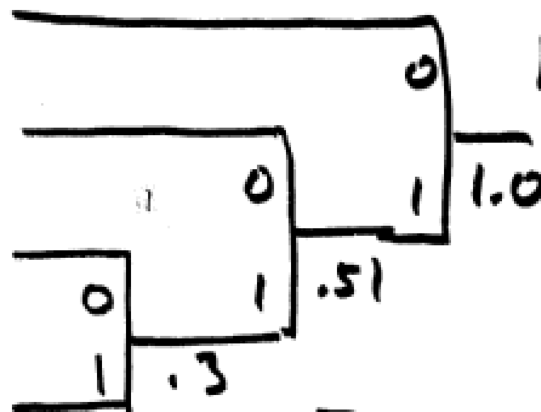
University of Idaho

(14)

data compression

for this example, suppose I coded the words before I sent them. (Huffman code)

	w_i		
0	w_3	(11)	.49
10	w_2	(10)	.21
110	w[illegible]	(01)	.21
111	w_0	(00)	.09



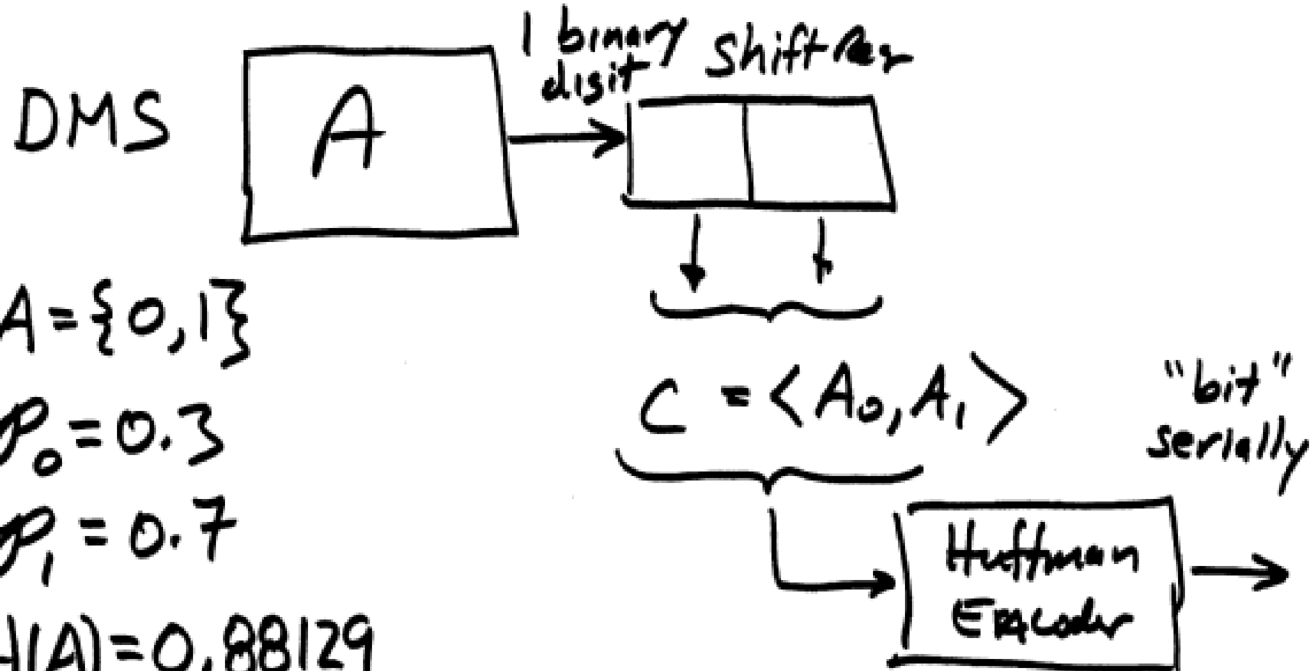
$\frac{\bar{L}}{2} = .905$

$H(A) = .881$

97% efficient

$\bar{L} = 1.81$ bits/word

Example (Huffman Code)



$A = \{0, 1\}$

$P_0 = 0.3$

$P_1 = 0.7$

$H(A) = 0.88129$

eff. = 88.1%

University of Idaho

A is DMS

⑥

$p(a_0, a_1) = p(a_0)\,p(a_1)$

A_0	A_1	$p(A_0, A_1)$
0	0	$(.3)(.3) = .09$
0	1	$(.3)(.7) = .21$
1	0	$(.7)(.3) = .21$
1	1	$(.7)(.7) = .49$

since these A_i are stat. independant

$$H(A_0, A_1) = H(A_0) + H(A_1 | A_0)$$

$$= H(A) + H(A_1) = 2H(A) = 1.7625$$

University of Idaho

(7)

$A_0 A_1$	$p(A_0, A_1)$	code word C	code word length	avg. xmit code word length
0 0	$.09 = p_0$	C_0	l_0	$p_0 l_0$
0 1	$.21 = p_1$	C_1	l_1	$p_1 l_1$
1 0	$.21 = p_2$	C_2	l_2	$p_2 l_2$
1 1	$.49 = p_3$	C_3	l_3	$p_3 l_3$

average code length is

$$\overline{L} = \sum_{m=0}^{3} p_m l_m$$

University of Idaho

(9)

Then

$= n H(A)$ for DMS

$$\frac{H(A_0, A_1, \ldots, A_{n-1})}{n} \leq \bar{l} < \frac{H(A_0, A_1, \ldots, A_{n-1}) + 1}{n}$$

$\therefore$ in the limit as $n \to \infty$

→ entropy rate

$$\mathcal{H}(A) \leq \bar{l} < \mathcal{H}(A) + \frac{1}{n}$$

$\therefore$ Shannon says it is possible to find a source code such that we can transmit an avg # of symbols per out = entropy rate

Some Definitions

A. **What is the difference between lossless and lossy compression?**

In lossless data compression, the compressed-then-decompressed data is an exact replication of the original data. On the other hand, in lossy data compression, the decompressed data may be different from the original data. Typically, there is some distortion between the original and reproduced signal.

The popular WinZip program is an example of lossless compression. JPEG is an example of lossy compression.

B. **What is the difference between compression rate and compression ratio?**

Historically, there are two main types of applications of data compression: transmission and storage. An example of the former is speech compression for real-time transmission over digital cellular networks. An example of the latter is file compression (e.g. Drivespace).

The term ``compression rate'' comes from the transmission camp, while ``compression ratio'' comes from the storage camp.

Compression rate is the rate of the compressed data (which we imagined to be transmitted in ``real-time''). Typically, it is in units of bits/sample, bits/character, bits/pixels, or bits/second. Compression ratio is the ratio of the size or rate of the original data to the size or rate of the compressed data. For example, if a gray-scale image is originally represented by 8 bits/pixel (bpp) and it is compressed to 2 bpp, we say that the compression ratio is 4-to-1. Sometimes, it is said that the compression ratio is 75%.

Compression rate is an absolute term, while compression ratio is a relative term.

We note that there are current applications which can be considered as both transmission and storage. For example, the above photograph of Shannon is stored in JPEG format. This not only saves storage space on the local disk, it also speeds up the delivery of the image over the internet.

C. **What is the difference between ``data compression theory'' and ``source coding theory''?**

There is no difference. They both mean the same thing. The term ``coding'' is a general term which could mean either ``data compression'' or ``error control coding''.

Higher Order Codes Converge

3. Third-Order Block Code: Triplets of characters are mapped to bit sequence of lengths one through six.

B_3	$p(B_3)$	Codeword
aaa	0.405	0
bbb	0.405	10
aab	0.045	1100
abb	0.045	1101
bba	0.045	1110
baa	0.045	11110
aba	0.005	111110
bab	0.005	111111
R=0.68 bits/character		

An example:

Original Data:	a a a	a a a	a b b	b b b	b b b	b b b	b b a	a a a
Compressed Data:	0	0	1101	10	10	10	1110	0

Note that 17 bits are used to represent 24 characters --- an average of 0.71 bits/character.

University of Idaho

(11)

Huffman code		P_i	$l_m P_m$
11	0	.49	.49
10	10	.21	.42
01	110	.21	.63
00	111	.09	.27
			1.81 = $\bar{L}$

$$\bar{l} = \frac{\bar{L}}{2} = 0.905$$

$$H(A) = 0.88129$$

$$\frac{H(A)}{\bar{l}} = 97\%$$

97.38%

University of Idaho

(12)

At the Rx

Suppose we receive the following

0 | 1 0 | 1 1 1 | 1 0 | 0 | 0 | 0 | 1 0

data decoded → 11 10 00 10 11 11 11 10

Prefix condition : can decode as soon as you Rx a complete codeword.

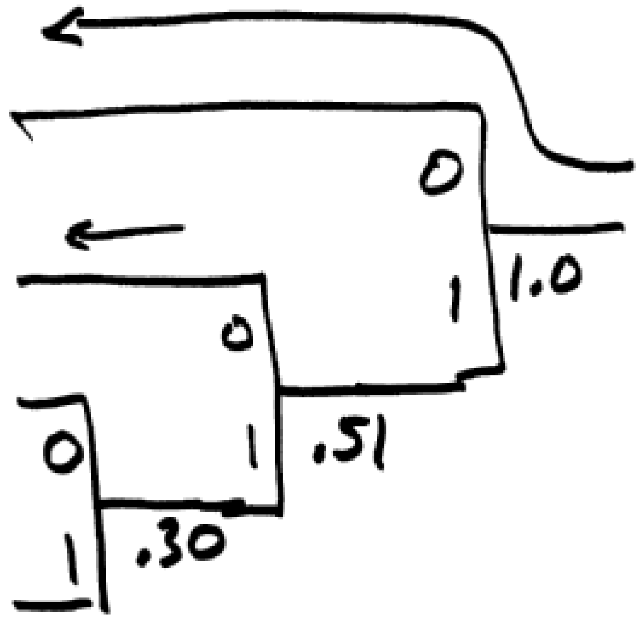
what about hardware?

University of Idaho

Huffman Algorithm

(10)

$A_0 A_1$	p_i	code word
1 1	.49	0
1 0	.21	1 0
0 1	.21	1 1 0
0 0	.09	1 1 1



first → last

Huffman tree

"prefix condition"

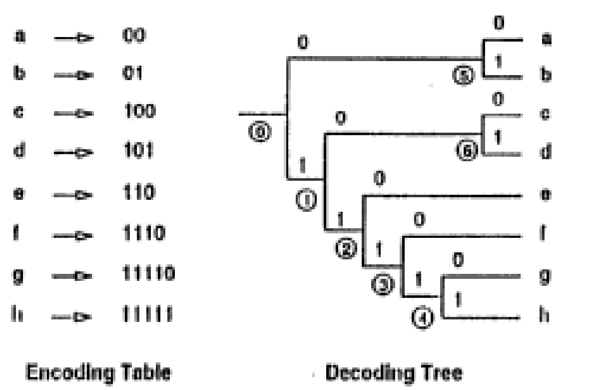


Figure 12.1 Example of a Huffman encoding table with the corresponding decoding tree.

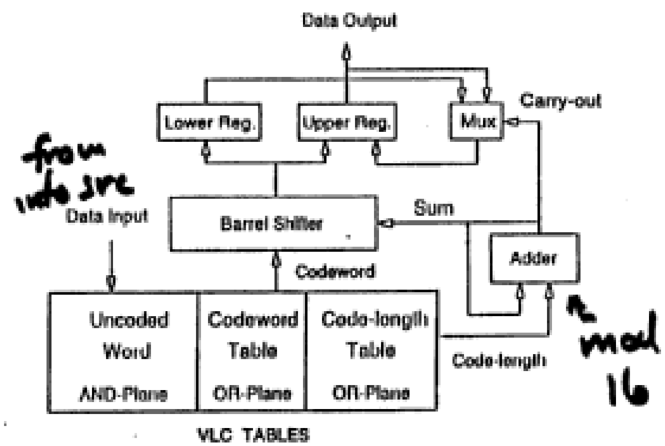


Figure 12.2 Block diagram of the Lei-Sun VLC encoder.

Input	Upper Register	Lower Register	Sum	Carry Out
g	1111000000000000	0000000000000000	5	0
b	1111001000000000	0000000000000000	7	0
a	1111001000000000	0000000000000000	9	0
c	1111001001000000	0000000000000000	12	0
b	1111001001000100	0000000000000000	14	0
f	1111001001000111	1000000000000000	2	1
d	1010100000000000	0000000000000000	5	0

Table 12.1 Example of operation of the VLC encoder.