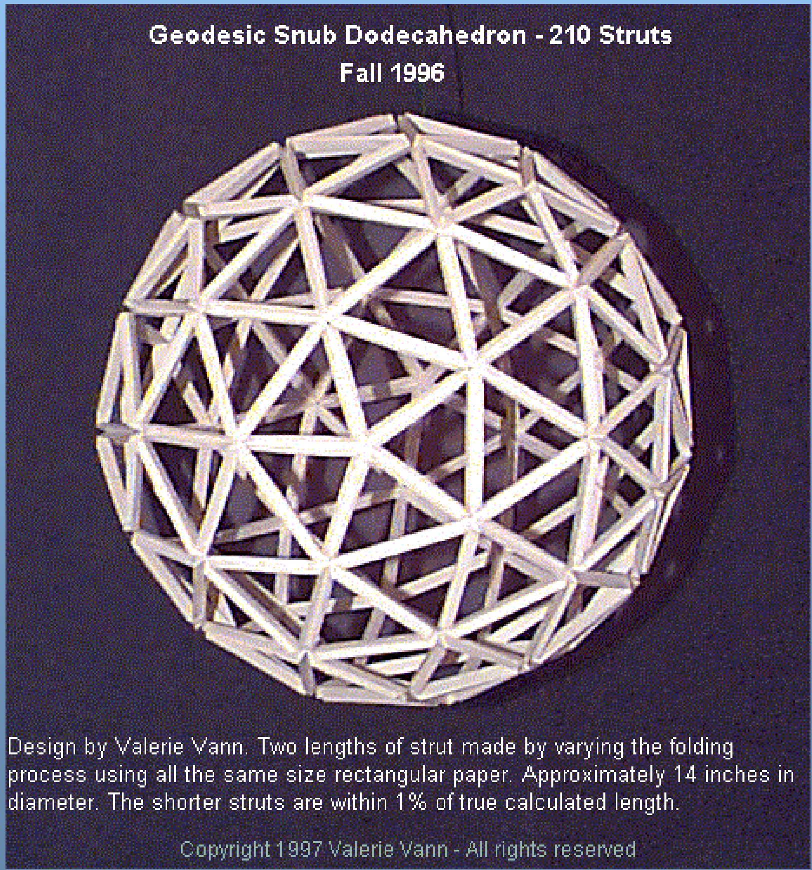
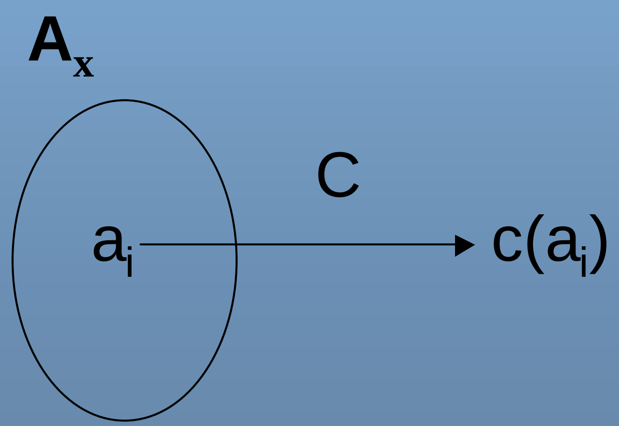
The Revenge of a Student - Symbol Codes



Symbol codes

- Notation: $\{0,1\}^+=\{0,1,00,01,10,11,000,\ldots\}$
- A symbol code C is a mapping from $\mathbf{A_x}$ to $\{0,1\}^+$

$$c^+(x_1x_2x_3\ldots x_N) = c(x_1)c(x_2)c(x_3)\ldots\ c(x_N)$$



$$l(x) = |x|$$

Decoding of symbol codes

- A code C(X) is uniquely decodable if

$$\forall \mathbf{x}, \mathbf{y} \in A_X^+, \mathbf{x} \neq \mathbf{y} \Rightarrow c^+(\mathbf{x}) \neq c^+(\mathbf{y})$$

- A code C(X) is a prefix code if no codeword is a prefix of any other codeword
- The expected length L(C,X) of a symbol code C for ensemble X is

$$L(C, X) = \sum_{x \in A_X} P(x) l(x)$$

Example

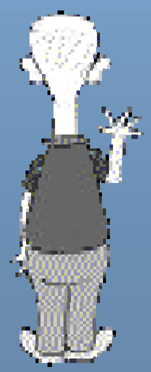
$\mathbf{A_x} = \{1,2,3,4\}$, $P_X = \{1/2,1/4,1/8,1/8\}$

C: c(1) = 0, c(2) = 10, c(3) = 110, c(4) = 111

The entropy of X is 1.75 bits: L(C,X) is also 1.75 bits

Obs!

$$l_i = \log_2(1/p_i), p_i = 2^{-l_i}$$



Kraft inequality

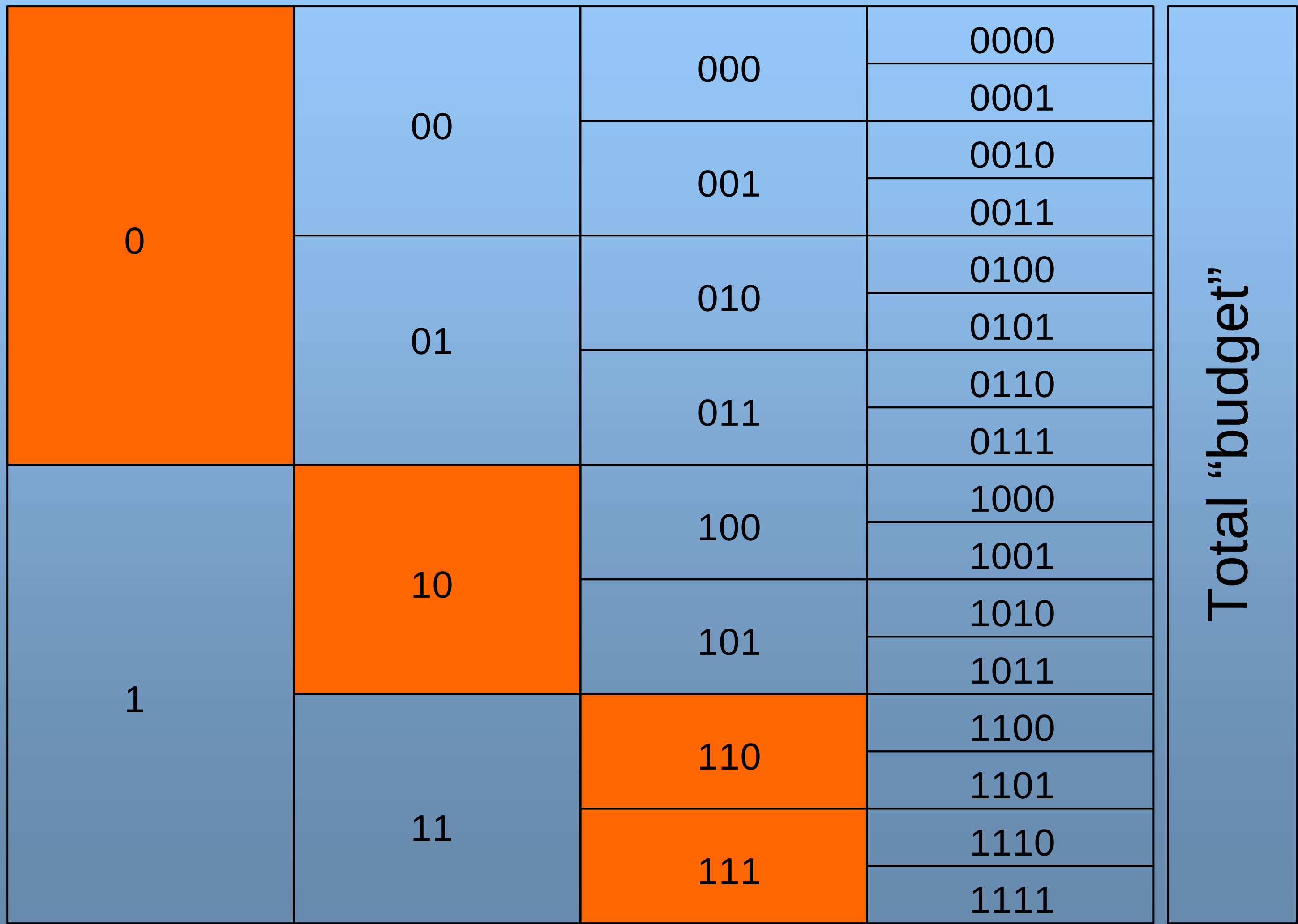
- Given a list of integer $\{l_i\}$, does there exist a uniquely decodable code with $\{l_i\}$?
- "Market model": total budget 1; cost per codeword of length *l* is 2^{-l}.

Kraft inequality: For any uniquely decodeable code C over the binary alphabet {0,1}, the codeword lengths must satisfy:

$$\sum_i 2^{-l_i} \leq 1$$

Conversely, given a set of codeword lengths that satisfy this inequality, there exists a uniquely decodable prefix code with these codelengths.

Limits of unique decodeability



What can we hope for?

Lower bound on expected length: The expected length L(C,X) of a uniquely decodable code is bounded below by H(X).

Compression limit of symbol codes: For an ensemble X there exists a prefix code

$$H(X) \leq L(C,X) < H(X) + 1.$$

"Proof-map" of the lower bound

Define $q_i \equiv 2^{-l_i / z}$, where $z = \sum_{i'} 2^{-l_{i'}}$

By the definition of log

Thus $l_i = \log 1/q_i - \log z$

Substitution

$$L(C, X) = \sum_i p_i l_i = \sum_i p_i \log 1/q_i - \log z$$

≥ 0

Kraft inequality

$$\geq \sum_i p_i \log 1/p_i \boxed{-\log z}$$

Gibbs inequality

$$\geq H(X)$$

(What happens if we use the "wrong" code?)

Assume the "true probability distribution" is $\{p_i\}$. If we use a complete code with lengths l_i, they define a probabilistic model $q_i = 2^{-l_i}$. The average length is

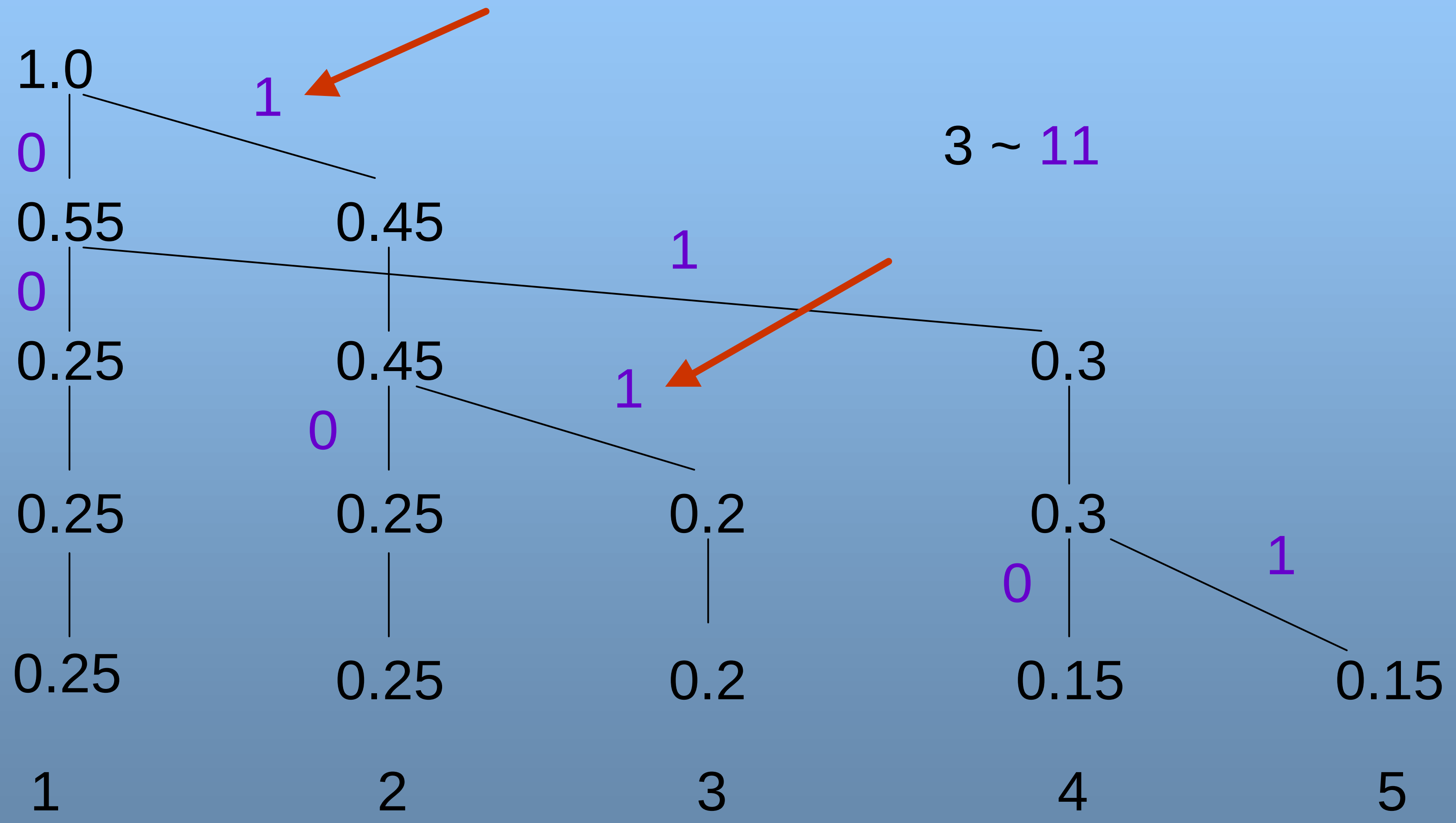
$$L(C,X) = H(X) + \boxed{\sum_i p_i \log p_i / q_i}$$

Kullback-Leibler divergence $D_{KL}(p||q)$

"Optimal" symbol code: Huffman coding

- Take two least probable symbols in the alphabet as defined by $\{p_i\}$.
- Combine these symbols into a single symbol, $p_{new} = p_1 + p_2$. Repeat (until one symbol)

Huffman in practice



Huffman for the Linux manual

$L(C,X) = 4.15$ bits

$H(X) = 4.11$ bits

a_i	p_i	$\log_2 \frac{1}{p_i}$	l_i	$c(a_i)$
a	0.0575	4.1	4	0000
b	0.0128	6.3	6	001000
c	0.0263	5.2	5	00101
d	0.0285	5.1	5	10000
e	0.0913	3.5	4	1100
f	0.0173	5.9	6	111000
g	0.0133	6.2	6	001001
h	0.0313	5.0	5	10001
i	0.0599	4.1	4	1001
j	0.0006	10.7	10	1101000000
k	0.0084	6.9	7	1010000
l	0.0335	4.9	5	11101
m	0.0235	5.4	6	110101
n	0.0596	4.1	4	0001
o	0.0689	3.9	4	1011
p	0.0192	5.7	6	111001
q	0.0008	10.3	9	110100001
r	0.0508	4.3	5	11011
s	0.0567	4.1	4	0011
t	0.0706	3.8	4	1111
u	0.0334	4.9	5	10101
v	0.0069	7.2	8	11010001
w	0.0119	6.4	7	1101001
x	0.0073	7.1	7	1010001
y	0.0164	5.9	6	101001
z	0.0007	10.4	10	1101000001
-	0.1928	2.4	2	01

Figure 3.3. Huffman code for the English language ensemble introduced in figure 1.16.

Why is this not the end of the story?

- Adaptation: what if the ensemble X changes? (as it does…)
  - ✓calculate probabilities in one pass
  - ✓communicate code + the Huffman-coded message
- "The extra bit": what if H(X) ~1 bit?
  - ✓Group symbols to blocks and design a "Huffman block code"