How much can we compress? - Shannon's Source Coding Theorem

On Probability and Entropy

Probability

- An ensemble X is a random variable x with a set of possible outcomes $\mathbf{A_x}$ with probabilities $\mathbf{P_x}$
- Probability of a subset T of $\mathbf{A_x}$

$$P(T) = \sum_{a_i \in T} P(x = a_i)$$

- A **joint ensemble** XY is an ensemble for which the outcomes are ordered pairs x,y where $x \in \mathbf{A_x}$ and $y \in \mathbf{A_y}$

Probability continued

- Marginal probability (from the joint probability P(x,y))

$$P(y) = \sum_{x \in A_x} P(x,y)$$

- Conditional probability

$$P(x = a_i \mid y = b_j) \equiv \frac{P(x = a_i, y = b_j)}{P(y = b_j)}$$

Probability continued

- Product rule

$$P(x, y \mid H) = P(x \mid y, H) P(y \mid H)$$

- Sum rule

$$P(x \mid H) = \sum_y P(x, y \mid H)$$
$$= \sum_y P(x \mid y, H) P(y \mid H)$$

Bayes's theorem

$$P(y \mid x, H) = \frac{P(x \mid y, H) P(y \mid H)}{P(x \mid H)}$$
$$= \frac{P(x \mid y, H) P(y \mid H)}{\sum_{y'} P(x \mid y', H) P(y' \mid H)}$$

Bayesian view of probability!

Entropy

- The entropy of X is a measure of the information content or "uncertainty" of x
  - ✓H(X) ≥ 0 (= iff p_i=1 for one i)
  - ✓H(X) ≤ log (|X|) (= iff p_i=1 /|X| for all i)

$$H(X) \equiv \sum_{x \in A_x} P(x) \log \frac{1}{P(x)}$$

Binary entropy

$$H(X) \equiv \sum_i p_i \log_2 \frac{1}{p_i}$$

Information measure?

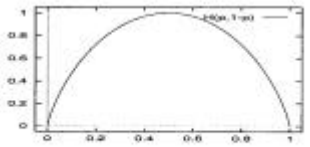


Figure 2.1. The binary entropy function $H_2(p) = H(p, 1-p) = p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}$ as a function of p.

Information content

- First attempt: number of possible outcomes $|A_x|$
  - ✓not additive: for xy we have $|A_x||A_y|$
- Perfect information content
  - ✓additive, but no probabilistic element

$$H_0(X) = \log_2 |A_X|$$

Shannon information

- looking for an information content of the event x=a_i

$$h(x) = \log_2 \frac{1}{p_i}$$

Example: letter distribution

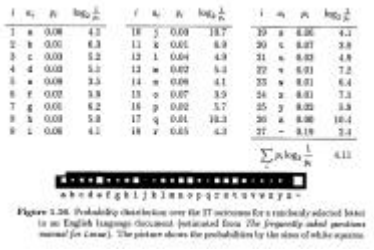


Figure 1.26. Probability distribution over the 27 outcomes for a randomly selected letter in an English language document (estimated from *The Frequently Asked Questions Manual for Linux*). The picture shows the probabilities by the areas of white squares.

Entropy continued

- The joint entropy of X,Y

$$H(X,Y) \equiv \sum_{xy \in A_x A_y} P(x,y) \log \frac{1}{P(x,y)}$$

- The conditional entropy of X given Y

$$H(X|Y) \equiv \sum_{y \in A_y} P(y) \left[\sum_{x \in A_x} P(x|y) \log \frac{1}{P(x|y)} \right]$$

$$= \sum_{xy \in A_x A_y} P(x,y) \log \frac{1}{P(x|y)}$$

"Average uncertainty that remains about x when y is known"

Entropy continued

- Chain rule for entropy

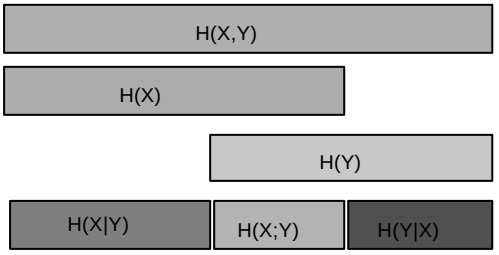
$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$

- Mutual information
  $H(X;Y) \equiv H(X) - H(X|Y)$

"Average reduction in uncertainty of x when learning the value of y

- Entropy distance
  $D_H(X,Y) \equiv H(X,Y) - H(X;Y)$

Entropy relationships

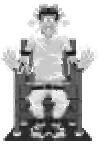


Kullback-Leibler divergence

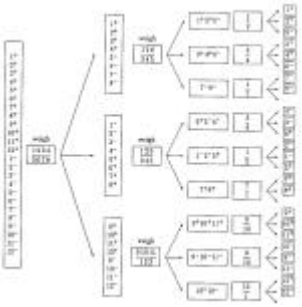
- Also known as "relative entropy"

$$D_{KL}(P \| Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

- Not strictly a "distance"



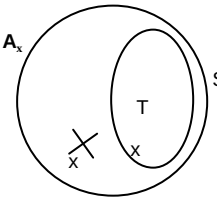
Weighting problem



Idea

- Some symbols have a smaller probability
- gamble that the rare symbols won't occur
- encode the observations in a smaller code (alphabet) C_X
- measure $\log_2|C_X|$
- the larger the risk, the smaller the alphabet

Formalize the idea



Smallest T s.t.
$P(x \notin T) < d$

Essential information

$$H_d(X) = \log_2 \min\{|T| : T \subseteq A_X, P(x \in T) \geq 1 - d\}$$

Example

x = $(x_1,\ldots,x_N)$, x = {0,1} with probabilities $p_0 = .9$, $p_1 = .1$
Let r(**x**) be the number of 1's in **x**

Probability of string **x**

$$P(\mathbf{x} \mid p_0, p_1) = p_0^{N-r(\mathbf{x})} p_1^{r(\mathbf{x})}$$

AEP and source coding

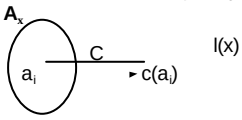
Asymptotic Equipartition Principle: for N i.i.d. random variables $X^N = \{X_1,\ldots,X_N\}$, with N sufficiently large, the outcome **x** = $\{x_1,\ldots,x_N\}$ is almost certain to belong to a subset of $\mathbf{A}_x^N$ having only $2^{NH(X)}$ members all having probability close to $2^{-NH(X)}$

The Revenge of a Student - Symbol Codes

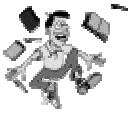
Symbol codes

- 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!

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

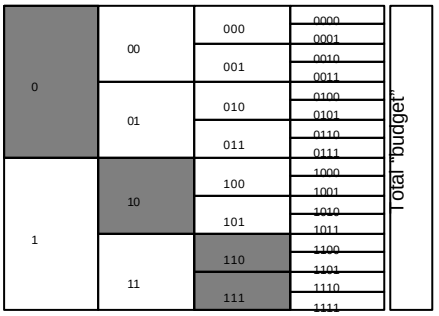
Kraft inequality

- 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: $\sum_i 2^{-l_i} \le 1$

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



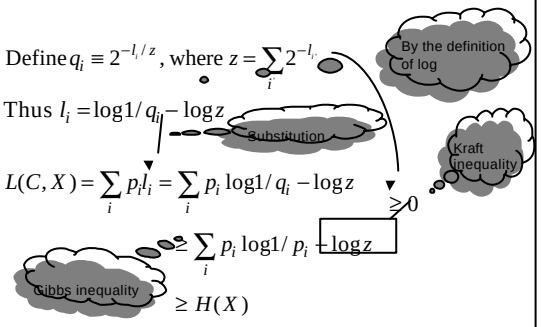
What can we hope for?

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

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

"Proof-map" of the lower bound



(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

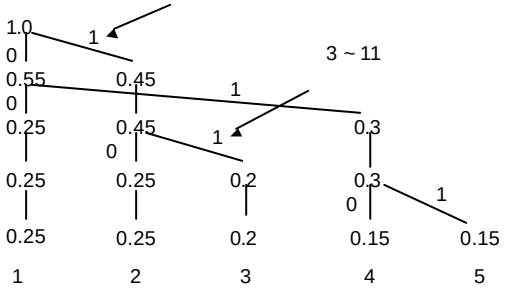
$$L(C,X) = H(X) + \sum_i p_i \log p_i / q_i$$

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)

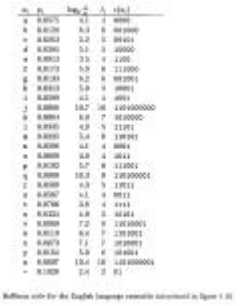
Huffman in practice



Huffman for the Linux manual

L(C,X) = 4.15 bits

H(X) = 4.11 bits



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"