Information Theory (1)

- Self-Information : a quantitative measure of information
  - An event A_1
  - $P(A_1)$: the probability that A_1 will occur
  - $i(A_1)$: self-information of A_1 $\quad i(A_1) = \log_b \frac{1}{P(A_1)} = -\log_b P(A_1)$ $\quad P(A_1)\uparrow \Rightarrow i(A_1)\downarrow$
    - log(1)=0 $\quad P(A_1)\downarrow \Rightarrow i(A_1)\uparrow$
    - -log(x) increases as x decreases from one to zero
    - for independent event A_1 & A_2

$$i(A_1A_2) = -\log_b P(A_1A_2) = -\log_b P(A_1)P(A_2) = -\log_b P(A_1) - \log_b P(A_2) = i(A_1) + i(A_2)$$

- Entropy (熵,平均訊息量) : the average self-information
  - A set of independent events A_j, $j=1\sim m$, which is the set of outcomes of some experiment S. ($\cup A_j = A$)
  - The average self-information given by

$$H(S) = \sum_{j=1}^{m} P(A_j)\, i(A_j) = -\sum_{j=1}^{m} P(A_j) \log_b P(A_j)$$

is called the entropy associated with this experiment.

Base	Unit
2	bits
e	nats
10	hartley

Information Theory (2)

- Example: Flipping a coin (a source with A={H,T})
  - $P(H) = P(T) = ½$, $i(H) = i(T) = 1$ bit, $H(S) = 1$ bit
  - $P(H) = 1/8$, $P(T) = 7/8$; $i(H) = 3$ bits, $i(T) = 0.193$ bits; $\Rightarrow H(S) = 0.544$ bit
- Shannon
  - If the experiment is a source that puts out symbols A_i from a set A, then the entropy is a measure of the average number of binary symbols needed to code the output of the source.
  - The best that a lossless compression scheme can do is to encode the output of a source with an average number of bits equal to the entropy of the source.

Information Theory (3)

- Entropy of the source
  - The set of symbols *A* is called the alphabet for the source
  - For a general source *S* with *A*={1,2,…,m} that generates a sequence {x_1, x_2,…}, the entropy of the source *S* is given by

$$H(S) = \lim_{n\to\infty} \frac{1}{n} G_n, \quad \text{where}$$

Entropy

$$G_n = -\sum_{i_1=1}^{m}\sum_{i_2=1}^{m}\cdots\sum_{i_n=1}^{m} P(x_1=i_1, x_2=i_2, \ldots, x_n=i_n)\log P(x_1=i_1, x_2=i_2, \ldots, x_n=i_n)$$

Suppose X_i independent identically distributed (iid)

First –order Entropy

$$G_n = -\sum_{i_1=1}^{m}\sum_{i_2=1}^{m}\cdots\sum_{i_n=1}^{m} P(x_1=i_1)P(x_2=i_2),\ldots,P(x_n=i_n)\times$$

$$[\log P(x_1=i_1)+\log P(x_2=i_2)+\ldots+\log P(x_n=i_n)]$$

$$= -\sum_{i_1=1}^{m} P(x_1=i_1)\log P(x_1=i_1)\sum_{i_2} P(x_2=i_2)\sum_{i_3} P(x_3=i_3)\ldots\sum_{i_n} P(x_n=i_n)$$

$$-\sum_{i_2=1}^{m} P(x_2=i_2)\log P(x_2=i_2)\sum_{i_1} P(x_1=i_1)\sum_{i_3} P(x_3=i_3)\ldots\sum_{i_n} P(x_n=i_n)-\ldots$$

$$= -n\sum_{i=1}^{m} P(x_1=i_1)\log P(x_1=i_1) \Rightarrow H(s) = -\sum_{i_1=1}^{m} P(x_1=i_1)\log P(x_1=i_1)$$

Information Theory (4-1)

- Example : Given the source sequence {12323454567898910}

① iid assumption (single character)

P(1)=P(6)= P(7)=P(10)=1/16, P(2)= P(3)=P(4)=P(5)=P(8)=P(9)=2/16

$$H = -\sum_{i=1}^{10} P(i)\log P(i) = 3.25 \ {}^{bits}\!/_{symbol}$$

② – remove the correlation by taking differences of neighboring sample values as the following residual sequence

1 1 1 -1 1 1 1 -1 1 1 1 1 1 -1 1 1

– static modeling $x_n=x_{n-1}+r_n$ and iid assumption (single character)

P(1) =13/16, P(-1)=3/16 H=0.7 bits/symbol

!! Usually, it's not possible to know the entropy for a physical source

Information Theory (4-2)

- Example : Given the source sequence {12123333123333123312}
  - ① iid assumption (single character)
    P(1)=P(2)= 1/4, P(3) =1/2 H=1.5 bits/symbol total bits = 1.5×20=30 bits
  - ② iid assumption (single block)
    P(12) =1/2, P(33) =1/2 H = 1 bits/block total bits = 0.5×20=10 bits
  - ③ Limitation for larger block size ⇒ accurate model is necessary

!! $H(S) = \lim_{n\to\infty} \frac{1}{n} G_n$ says we can always extract the structure of the data by taking larger and larger block size.

Models(1)

- Physical Models
  - Based on the physics of the data generation process
  - e.g. speech-related applications
- Probability Models
  - independence assumption : ignorance model
  - $A=\{a_1, a_2, \ldots a_M\}$ $P=\{P(a_1), P(a_2),\ldots, P(a_M)\}$, $H(S) = \sum_{i=1}^{M} P(a_i)\ i(a_i)$
- Markov Models
  - Kth-order Markov model
    $P(X_n|X_{n-1},\ldots,X_{n-k})=P(X_n|X_{n-1}, X_{n-2}, \ldots, X_{n-k}, \ldots)$
  - 1st-order Markov model
    $P(X_n|X_{n-1})=P(X_n|X_{n-1}, X_{n-2}, \ldots, X_{n-k},\ldots)$

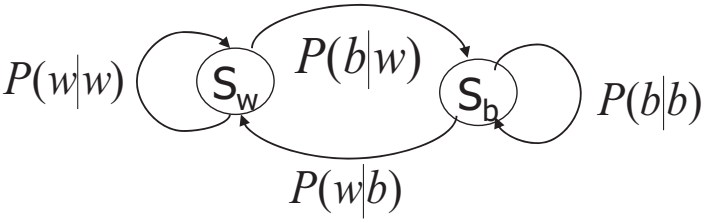
Ex. 1 $x_n = \rho x_{n-1} + \varepsilon_n$ ε_n : white noise process

Ex. 2 discrete time Markov chain

Models(2)

- Example : binary image with $P(S_w)=30/31$, $P(S_b)=1/31$, $P(w|w)=0.99$, $P(b|w)=0.01$, $P(b|b)=0.7$, and $P(w|b)=0.3$
  - Simple probability model (iid assumption)
    $H = -P(S_w)\log P(S_w) - P(S_b)\log(S_b) = 0.206$ bits
  - Finite state model (1st-order discrete time Markov chain)

S_w: the current pixel is a white pixel



$$\begin{cases} P(S_w) = 0.99P(S_w) + 0.3P(S_b) \\ P(S_b) = 0.7P(S_b) + 0.01P(S_w) \end{cases}$$

$$with\ P(S_w) + P(S_b) = 1$$

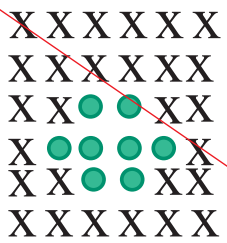
$\Rightarrow$ $P(S_w)=30/31$, $P(S_b)=1/31$

$$H(S_w) = -P(b|w)\log P(b|w) - P(w|w)\log p(w|w) = 0.081\ bits$$

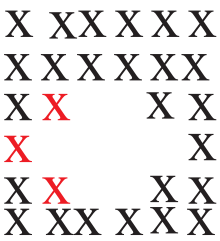
$$H(S_b) = -P(w|b)\log P(w|b) - P(b|b)\log p(b|b) = 0.881\ bits$$

$$H(S) = P(S_w)\cdot H(S_w) + P(S_b)\cdot H(S_b) = 0.107\ \text{bits}$$

Q: Exact entropy for the bi-level image = ?



(a)



S_w: current pixel is white

$P(S_w)=28/36$

P(w|w): current pixel is white, next pixel is white.

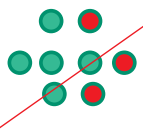
P(b|w): current pixel is white, next pixel is black.

P(w/w)=25/28
P(b/w)=3/28

28/36=25/28*28/36+3/8*8/36

S_b: current pixel is black

$P(S_b)=8/36$

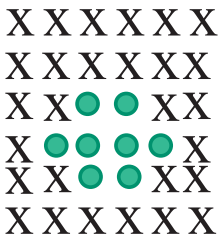


P(w/b)=3/8
P(b/b)=5/8

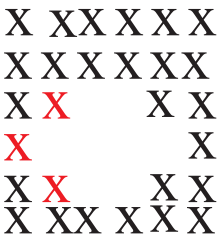
P(w|b): current pixel is black, next pixel is white.

P(b|b): current pixel is black, next pixel is black.

8/36=5/8*8/36+3/28*28/36



(a)



S_w: the set of pixels of which each pixel is white.

$P(S_w)=28/36$

$P(w|w)$: current pixel is in S_w, next pixel is in S_w.

$P(b|w)$: current pixel is in S_w, next pixel is in S_b.

$P(w/w)=25/28$

$P(b/w)=3/28$

$28/36=25/28*28/36+3/8*8/36$

$P(S_b)$: the probability of a pixel which belongs to the set S_b

S_b: the set of pixels of which each pixel is black.

$P(S_b)=8/36$

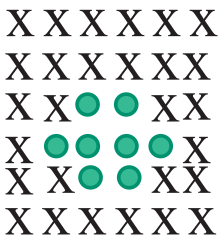
$P(b/b)=5/8$

$P(w/b)=3/8$

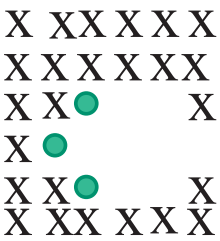
$P(w|b)$: current pixel is in S_b, next pixel is in S_w.

$P(b|b)$: current pixel is in S_b, next pixel is in S_b.

$8/36=5/8*8/36+3/28*28/36$



(b)



S_w: the set of pixels of which the previous pixel of each pixel is white.

$P(S_w)=28/36$

$P(w|w)$: current pixel is in S_w, next pixel is in S_w.

$P(b|w)$: current pixel is in S_w, next pixel is in S_b.

$P(w/w)=25/28$

$P(b/w)=3/28$

$28/36=25/28*28/36+3/8*8/36$

S_b: the set of pixels of which the previous pixel of each pixel is black.

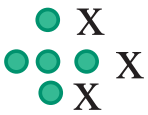
$P(w|b)$: current pixel is in S_b, next pixel is in S_w.

$P(b|b)$: current pixel is in S_b, next pixel is in S_b.

$P(S_b)=8/36$

$P(b/b)=5/8$

$P(w/b)=3/8$



$8/36=5/8*8/36+3/28*28/36$

Coding (1-1)

- codec = coder + decoder
- Redundancy *(R)*
  - l_j, $j=1\sim N$: length of the codeword of a_j
  - average codeword length $\bar{l} = \sum_{j=1}^{N} P(a_j)\, l_j$
  - $R = \bar{l} - H(S)$

- set of codewords ≡ code (C)
- set of symbols (a_j) ≡ alphabet *A*

$$H = -\sum_{j=1}^{4} P(a_j)\log P(a_j) = 1.75 \ {}^{bits}\!/_{symbol}$$

symbols	P(a_j)	Code-0	Code-1	Code-2	Code-3	Code-4
a1	**0.5**	**00**	**0**	**0**	**0**	**0**
a2	**0.25**	**01**	**0**	**1**	**10**	**01**
a3	**0.125**	**10**	**1**	**00**	**110**	**011**
a4	**0.125**	**11**	**10**	**11**	**111**	**0111**
$\bar{l}$		2.0	1.125	1.25	1.75	1.875

fixed-length code — ambiguous code — not uniquely decodable code — prefix code — not instantaneously decodable code

Coding (1-2)

Symbols	Codewords
a_1	0
a_2	01
a_3	11

Symbols	Codewords
a_1	0
a_2	01
a_3	10

011111111111111111

$a_1 a_3\ a_3\ a_3\ a_3\ a_3\ a_3\ a_3\ a_3 1$

$a_2\ a_3\ a_3\ a_3\ a_3\ a_3\ a_3\ a_3\ a_3$

not instantaneously decodable code

01010101010101010

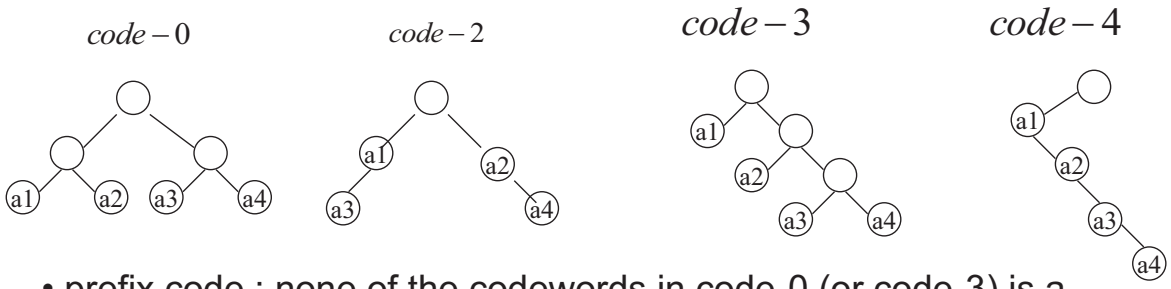
$a_1 a_3\ a_3\ a_3\ a_3\ a_3\ a_3\ a_3\ a_3$

$a_2\ a_2\ a_2\ a_2\ a_2\ a_2\ a_2\ a_2\ a_1$

not uniquely decodable code

(Note: $K(C) \le 1$)

Coding (2)



- prefix code : none of the codewords in code-0 (or code-3) is a prefix of any of the other codewords
- prefix code $\rightleftarrows$ uniquely decodable
- equivalent codes ≡ owns the same $\bar{l}$
- any uniquely decodable code is equivalent to one prefix code

Q: 如何找？

- $\bar{l} = H$ only when $P(a_j)=2^{-k}$.

Kraft-McMillan Inequality (1)

- Theorem (part I): Let C be a code with N codewords with length l_1, l_2, … l_N. If C is uniquely decodable, then

$$K(C) = \sum_{i=1}^{N} 2^{-l_i} \le 1 \quad \text{: Kraft-McMillan Inequality}$$

$\rightarrow$

$\not\leftarrow$

Pf: $$\left[\sum_{i=1}^{N} 2^{-l_i}\right]^n = \sum_{i_1=1}^{N} 2^{-l_{i_1}} \sum_{i_2=1}^{N} 2^{-l_{i_2}} \ldots \sum_{i_n=1}^{N} 2^{-l_{i_n}} = \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} \ldots \sum_{i_n=1}^{N} 2^{-(l_{i_1}+l_{i_2}+\ldots+l_{i_n})}$$

$$\therefore\ n \le l_{i_1} + l_{i_2} + \ldots + l_{i_n} \le nl \text{, where } l = \max\{l_1, l_2, \ldots l_N\}$$

$$\therefore K(C)^n = \sum_{k=n}^{nl} A_k 2^{-k}, \text{ where } A_k \text{ is the number of combinations of n codewords}$$

that have a combined length of k. Claim that : $A_k \le 2^k$.

$$\therefore K(C)^n \le \sum_{k=n}^{nl} 2^k 2^{-k} = nl - n + 1 = n(l-1) + 1 \ \therefore \text{if } k(C) > 1 \ (\rightarrow\leftarrow)$$

If $k(C)$ >1 , it will grow exponentially with n, while n(l-1)+1 can only grow linearly.

Kraft-McMillan Inequality (2)

The number of possible distinct binary sequence of length k is 2^k. If C is uniquely decodable, then each sequence can represent one and only one sequence of codwords.

1 2 2 ← length of codeword

n codewords

x x x x x x … x

k bits

non-ambiguous $\rightleftarrows$ uniquely decodable

(non-singular)

- Q: if $K(C) \le 1$, Is it possible that C is not a prefix code & not uniquely decodable.
- Q: If we have a uniquely decodable code that is not a prefix code , we can always find a prefix code with the same codeword lengths.

Kraft-McMillan Inequality (3)

- Theorem (part II): Given a set of integers $l_1, l_2, \ldots l_N$ that satisfy the inequality $\sum_{i=1}^{N} 2^{-l_i} \le 1$. We can always find a prefix code with codeword length $l_1, l_2, \ldots l_N$.

Pf: Let n_i be the number of codewords with length i,

$1 \le i \le l_{max}$,where $l_{\max} = \max_{1 \le i \le N} l_i$

$$\Rightarrow n_1 2^{-1} + n_2 2^{-2} + \ldots\ldots + n_{l_{\max}} 2^{-l_{\max}} \le 1$$

$$\Rightarrow n_1 2^{-1} \le 1$$

$$n_1 2^{-1} + n_2 2^{-2} \le 1$$

$$\vdots$$

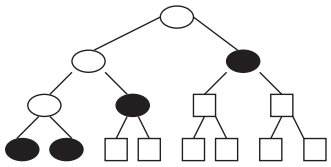
$$n_1 2^{-1} + n_2 2^{-2} + \ldots\ldots + n_{l_{\max}} 2^{-l_{\max}} \le 1$$

$$\Rightarrow n_1 \le 2$$

$$n_2 \le 2^2 - n_1 2^1$$

$$\vdots$$

$$n_{l_{\max}} \le 2^{l_{\max}} - n_1 2^{l_{\max}-1} - \ldots\ldots - n_{l_{\max}-1} 2^1$$



$N=4$ $l_{\max}=3$

$l_1=1, l_2=2, l_3=3, l_4=3$

$n_1=1, n_2=1, n_3=2$

$n_1 = 1 \le 2$

$n_2 \le 2^2 - 1 \cdot 2^1 = 2$

$n_3 \le 2^3 - 1 \cdot 2^2 - 1 \cdot 2^1 = 2$

Prefix code v.s. Kraft inequality

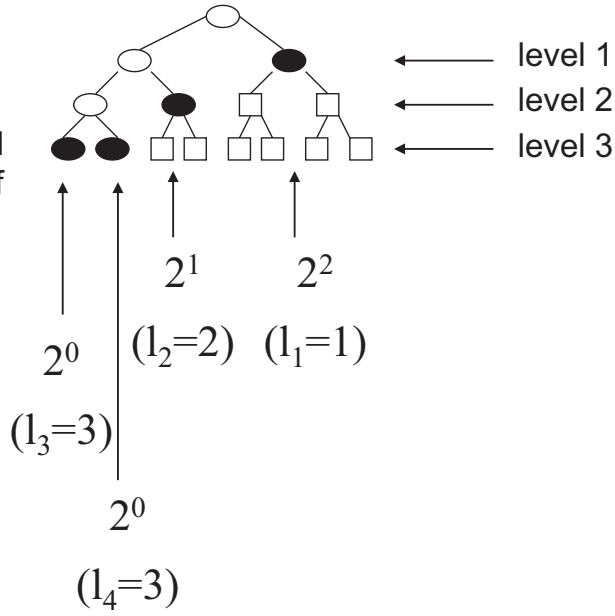
- Theorem : There exists a binary prefix code with N codewords of length l_i, i= 1,…,N. iff the kraft inequality holds.

Pf: (→)

Let $l_{\max} = \max_{1 \le i \le N} l_i$

The tree has $2^{l_{\max}}$ nodes on level l_{max} of the tree. Each codeword of length l_i obstructs $2^{l_{\max} - l_i}$ nodes on level l_{max}. Hence,

$$2^{l_{\max}} \sum_{i=1}^{N} 2^{-l_i} = \sum_{i=1}^{N} 2^{l_{\max} - l_i} \le 2^{l_{\max}},$$

$$\Rightarrow \sum_{i=1}^{N} 2^{-l_i} \le 1.$$

(←) see Theorem (part II)