Overview

Note Title 5/7/2007

Read Chap 1. of Cover & Thomas

Information theory studies the nature of "information" without caring the exact content of the information.

Two major questions to answer:

- Data compression limit: how do we quantify information?
  How much information do we have in a Cambridge dictionary? More precisely, how many bits do we need to store this information?

- Communication limit of transmission medium: how many bits we can pass through an twisted pair cable per second? How about an optical cable?

Amazingly, both questions are almost completely answered by Shannon's original paper - "A Mathematical Theory of Communication" (See course website for link).

Quantifying information in the Kolmogorov sense

How much information in $\pi = 3.14159265358979313 85\ldots$ comparing with $0.11111111111\ldots$?

Both of them will need infinit number of digits to represent exactly. Does that mean that both numbers encapsulate infinit amount of information in them? Intuitively, I think most people will answer no. At least not for $0.11111\ldots$, because it can be constructed simply by padding infinite # of 1 after the decimal point. For example, it is generated by the pseudo-code as follows:

```
print "0."
while (true)
{
   print "1"
}
```

The idea above is the Kolmogorov view of information. Let c be a piece of code & $\mathcal{U}(c)$ be the output of executing c over a universal computer. And $l(c)$ is the length of the code.
Then, Kolmogorov complexity (information content in Kolmogorov's sense),

$$K_U(x) = \min_{\mathcal{U}(c)=x} l(c)$$

We can do the same thing for π. It is known that

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} \cdots$$

So $\exists$ is possible to construct π with some code c with $l(c) < \infty$.

It should be convincing to see that the amount of information in π is more than that of $0.1111111\ldots$ in the Kolmogorov sense

Quantifying information in the probabilistic sense

Consider a random binary source with $p(X=1) = p$.
What's the average amount of information generated from the source?

Let's stick 100 consecutive outputs from the source, we have let say
1011 10 111 11 0 ....
We can compute the amount of information in the Kolmogorov sense as described previously.
We can repeat for another 100 outputs and another 100 outputs & so on.
Then we can estimate the information generated by the source as

$$\frac{E[K]}{100} \leftarrow \text{Average Kolmogorov complexity of length-100 sequence from source}$$

The Kolmogorov complexity computation requires us to find the optimal code for each input. Let's just consider a simple fixed coding scheme here - run-length code. The idea is simply to count and write out the number of consecutive 1 or 0.
For example,

0 11111 0 11 0 1 $\rightarrow$ 1-0, 5-1, 1-0, 2-1, 1-0, 1-1

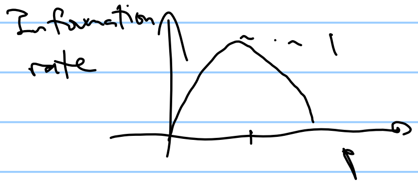
&

0000 1 0 11111 $\rightarrow$ 4-0, 1-1, 1-0, 5-1

Actually, one can see that as the distribution become more & more skew (i.e. $p \rightarrow 0$ or $p \rightarrow 1$), the average code length will become shorter & shorter.

Moreover the average code length for source with $p(X=1)=p$ is the same as that of $p(X=1)=1-p$.

Actually, the average amount of information per sample (info rate)
$= -p \log p - (1-p)\log(1-p)$



We are going to show this in the next few weeks.

Capacity of a medium

A very surprising result by Shannon was that we can always pass information across a noisy channel as long as the transmission rate is sufficiently small.

Previously, people believed that if a channel is subject to noise, the output is noisy and therefore there had to be error for the received signal.

Shannon not only showed that error free transmission is possible. He showed that increasing the transmission rate will not increase the amount of transmission error as long as increased rate is still lower than a "channel capacity".

Information theory & other areas

IT is related to many other disciplines. It borrows many ideas from statistical physics & is used in economics (portfolio theory) & genomics (bioinformatics). We won't cover those topics in class but you are extremely encouraged to dig deeper along these directions through your course project.

Probability review

Reading assignment / self-study = review calculus, probability.

Some notations:

$\forall$ = for all $\quad\quad$ $\vee$: or

$\exists$: there exists $\quad\quad$ $\wedge$: and

Interval : $(0,1)$, $[0,1)$, $(0,1]$ . . . .

x lies in $(0,1)$ $\Leftrightarrow$ $0 < x < 1$

x lies in $[0,1)$ $\Leftrightarrow$ $0 \le x < 1$

A set is a collection of elements. Usually denote by script capital letter.

ϕ denotes the empty set that contains nothing

$\mho$ denotes the universal set that contains everything of interest

$\mathbb{Z}$: set of all integers

$\mathbb{R}$: set of all real numbers

$\mathbb{C}$: set of all complex numbers

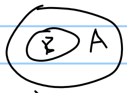
A is composed of a, b & c $\equiv$ $A = \{a,b,c\}$

A is set of all even integer $\equiv$ $A = \{a \mid b = 2a,\ a \in \mathbb{Z}\}$

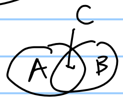
A is the set of all +ve even integers $\equiv$ $A = \{a \mid b = 2a,\ a > 0,\ a \in \mathbb{Z}\}$

a is an element of a set A $\equiv$ $a \in A$

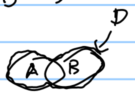
B is a subset of A $\equiv$ $B \subset A$
$(\forall\ b \in B \Rightarrow b \in A)$



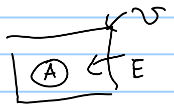
C is the intersection of A & B $\equiv$ $C = A \cap B$
$(\forall\ c \in C \Rightarrow c \in A$ & $c \in B)$



D is the union of A & B $\equiv$ $D = A \cup B$



E is the complement of A $\equiv$ $E = A^c$
$(A \cap E = \phi$ & $A \cup E = \mho)$



Probability describes how likely something happens.

By convention, probability of an event is bounded within $[0,1]$

$Pr(E) = 1$ $\leftarrow$ event always happens

$Pr(E) = 0$ $\leftarrow$ event can never happen

Eg. $Pr($sun rises from the east$) = 1$

$Pr($sun rises from the west$) = 0$

$Pr($I win lottery tomorrow$) = 0.00 \cdots 0 \text{—} 0\ 1$

Random Variable

X is a r.v. (random variable) if X is not deterministic (exact value not known)

x is a realization (outcome) of X (a constant)

Convention: big letter $\equiv$ r.v, small letter $\equiv$ const

Eg. $\Pr(X > 0)$ $\leftarrow$ make sense
$\Pr(x > 0)$ $\leftarrow$ doesn't make sense

If X can only take discrete value, it is a discrete r.v.
o.w, X is a continuous r.v.

Even tho exact value of X is not known, we usually know some of its "statistics".

Eg. mean & variance of X

In the best scenario, we know the "distribution" of X, then basically we know its complete statistics.

For discrete r.v., its distribution is specified by its pmf (probability mass function), $P_X(x)$

Eg. $$P_X(x) = \begin{cases} 0.1, & x = 1 \\ 0.2, & x = 2 \\ 0.7, & x = 3 \end{cases}$$

It means that with prob of 0.1, $X = 1$.

$$\therefore P_X(x) = \Pr(X = x)$$

Note that $P_X(x)$ should sum to 1 ($\sum_x P_X(x) = 1$) by the definition of probability.

For continuous r.v., its distribution can be specified by its pdf (probability density function)

Note that as X is continuous, $\Pr(X = x)$ can be equal to 0 for all x in general. So we can't define pdf the same as pmf for discrete r.v. Instead,

$$\text{pdf} \rightarrow P_X(x) \triangleq \lim_{\Delta x \to 0} \frac{\Pr(x - \frac{\Delta x}{2} \le X \le x + \frac{\Delta x}{2})}{\Delta x}$$

N.B. For both discrete (pmf) + continuous (pdf) cases, $P_X(x) \ge 0$
For discrete case $P_X(x) \le 1$, but for continuous case, $P_X(x) < \infty$ but is not bounded by any finite number.

When there is no ambiguity, we may use the shorthand notation $p(x)$ instead of $P_X(x)$

"Expectation" of a function of X is what you expect to get.

Notation: $E[g(X)]$

Eg. A hunter randomly shoots in a field, there is 0.6 prob. he misses, 0.1 prob. he shoots some birds, 0.3 prob. he shoots some mammals. What is the expected # of legs he get?

$X \in \{\phi, \text{bird}, \text{mammal}\}$ $\qquad g: \{\phi, \text{bird}, \text{mammal}\} \to \{0, 2, 4\}$

$p_X(\phi) = 0.6$ $\qquad g(\phi) = 0$

$p_X(\text{bird}) = 0.1$ $\qquad g(\text{bird}) = 2$

$p_X(\text{mammal}) = 0.3$ $\qquad g(\text{mammal}) = 4$

$$\text{Expected \# of legs} = E[g(X)] = 0.6 \cdot 0 + 0.1 \cdot 2 + 0.3 \cdot 4$$
$$= 1.4 \text{ legs.}$$

In general,
$$E[g(X)] = \begin{cases} \sum_{x \in \{x' \mid p_X(x') > 0\}} p_X(x)\, g(x) & \text{for discrete r.v.} \\ \int_{\mathbb{R}} p_X(x)\, g(x)\, dx & \text{for continuous real r.v.} \\ \int_{\mathbb{C}} p_X(x)\, g(x)\, dx & \text{for continuous complex r.v.} \end{cases}$$

When X takes numerical value, we can consider its mean (average) & variance.

mean of $X = E[X]$

& variance of $X = E[(X - E(X))^2]$

The mean of X is also commonly denoted by $\overline{X}$

Properties of expectation:

Linearity: $E[aX + bY] = aE[X] + bE[Y]$

Pf: $$E[aX + bY] = \sum_{\substack{z \in \\ \{z' \mid p_X(z') \neq 0 \text{ or } \\ p_Y(z') \neq 0\}}} a\, p_X(z) + b\, p_Y(z)$$

$$= a \sum_{z \in ..} p_X(z) + b \sum_{z \in ..} p_Y(z)$$

$$= a \sum_{z \in \{z' \mid p_X(z') \neq 0\}} p_X(z) + b \sum_{z \in \{z' \mid p_Y(z') \neq 0\}} p_Y(z) \quad \text{(why?)}$$

$$= aE(X) + bE[Y] \quad \blacksquare$$

Multiple Random Variables

The statistics of r.v.s X, Y is determined by the joint pmf (discrete) & pdf (continuous), $p_{XY}(x,y)$

When there is no ambiguity, we can write $p_{XY}(x,y)$ as $p(x,y)$

Sometimes, we write $p(y,x)$ to mean $p_{Y,X}(y,x)$, apparently

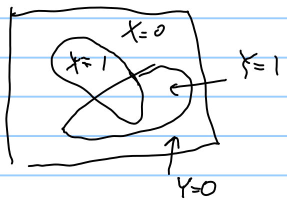
$p_{XY}(x,y) = p_{YX}(y,x)$ or $p(x,y) = p(y,x)$

$\therefore$ the order of the variables are not important

Conditional distribution:

$$\underset{\substack{\uparrow \\ p_{Y|X}(1|1)}}{Pr(Y=1 \mid X=1)} = \underset{\substack{\uparrow \\ p_{YX}(1,1)}}{Pr(Y=1 \cap X=1)} / \underset{\substack{\uparrow \\ p_X(1)}}{Pr(X=1)}$$ $\leftarrow$ pmf (discrete case)

In general, $p(y|x) = \frac{p(x,y)}{p(x)}$



For continuous case

$$\underset{\substack{\uparrow \\ p_{Y|X}(1|1)\cdot\Delta y}}{Pr(Y=1 \mid X=1)} = \underset{\substack{\uparrow \\ p_{XY}(1,1)\,\Delta x\cdot\Delta y}}{Pr(Y=1 \cap X=1)} / \underset{\substack{\uparrow \\ p_X(1)\,\Delta x}}{Pr(X=1)}$$ $\leftarrow$ pdf

$\Rightarrow$ still, $p(y|x) = \frac{p(x,y)}{p(x)} \Rightarrow p(x,y) = p(y|x)\,p(x)$

In general for r.v.s $X_1, X_2 \cdots X_n$, $p(x_1, x_2 \cdots x_n) = p(x_1)\,p(x_2|x_1)\,p(x_3|x_2 x_1) \cdots p(x_n|x_{n-1} \cdots)$ $\leftarrow$ chain rule

Moreover, since $p(\underbrace{z x_1}_{\substack{\uparrow \\ \text{treat as a single r.v.}}} x_2 x_3) = p(z x_1)\,p(x_2|z x_1)\,p(x_3|z x_1 x_2)$

Divide both LHS & RHS by z,

$$p(x_1 x_2 x_3 | z) = p(x_1|z)\,p(x_2|x_1, z)\,p(x_3|x_1 x_2 z)$$

Independence:

Two r.v.s X, Y are indep iff $p(x|y) = p(x)$ $\forall x, y$

Note that since $p(x,y) = p(y)\,p(x|y) \underset{\substack{\uparrow \\ \text{if independent } X \text{ \& } Y}}{=} p(y)\,p(x)$ $\forall x, y$

i.e. $p(x,y) = p(x)\,p(y)$ $\leftarrow$ a more common def of independence

Apparently, if X is indep. of Y, then Y is indep. of X.

E.g. prob(Head) = prob(Tail) = $\frac{1}{2}$

prob(a die's outcome = 1) = prob(die's outcome = 2) $\cdots = \frac{1}{6}$

prob(Head $\cap$ die's outcome = 1) = Pr(head) Pr(die's outcome = 1)

$= \frac{1}{2} \cdot \frac{1}{6} = \frac{1}{12}$

Conditional Independence & Markov chain:

R.v.s X, Y are cond. Indep given Z if $p(x|yz) = p(x|z), \forall x, y, z$ — (1)

Lemma 1. $p(x|yz) = p(x|z) \iff p(xy|z) = p(x|z)\,p(y|z)$ — (2)

Pf. $p(x,y|z) \overset{(2)}{=} p(x|z)\,p(y|z)$

chain rule $\Vert$

$p(x|yz)\,p(y|z)$ // (1)

From Lemma 1, $p(x|yz) = p(x|z) \iff p(y|xz) = p(y|z)$

i.e. X, Y cond. Ind. given Z $\equiv$ Y, X cond. Ind. given Z

We write $X \to Z \to Y$ $\equiv$ $Y \to Z \to X$ & say X, Z, Y form a Markov chain.

Uncorrelated variables

r.v.s X, Y are uncorrelated if $E[XY] = E[X]E[Y]$

* Conditional Expectation * (advanced topic)

For discrete r.v.s X, Y,

$$E[g(X)|y] \triangleq \sum_x g(x)\,p(x|y) \triangleq u(y)$$

$$E[g(X)|Y] \triangleq u(Y)$$

Conditional Expectation is a rather "advanced" topic & we will not use it in our class. For those who are interested, they can refer to classic probability text for engineers such as Papoulis.

Discrete Random Process

A discrete time random process is simply a sequence of r.v.s.

Random process $X_1, X_2 \cdots X_n \cdots$ is stationary if the statistics is unchange over time (i.e. progress of index)

Random process $X_1, X_2 \cdots X_n \cdots$ is wide-sense stationary if
$E[X_1] = E[X_2] \cdots = E[X_n]$
& $E[X_1 X_k] = E[X_{n+1} X_{k+n}]$ for any n, k

Random process $X_1, X_2 \cdots X_n \cdots$ is Markov (denoted by $X_1 \to X_2 \to X_3 \cdots X_n \to \cdots$) if

$$p(x_n | x_{n-1}\, x_{n-2} \cdots x_1) = p(x_n | x_{n-1}) \text{ for any } n$$

Discrete i.i.d. Source

i.i.d.: independent & identically distributed
An iid source generates an output as a random process & at each time instance, the output has the same statistics as that from any other time instances. That is, $p_{X_i}(x) = p_{X_j}(x)$ for any $i \neq j$
And outputs from different time instances are independent,
i.e. $p_{X_i X_j}(x_i x_j) = p_{X_i}(x_i)\, p_{X_j}(x_j)$ $i \neq j$
and $p_{X_i X_j X_k}(x_i x_j x_k) = p_{X_i}(x_i)\, p_{X_j}(x_j)\, p_{X_k}(x_k)$ for $i \neq j \neq k \neq i$
& so on ....

Ergodicity

When we say $E(X)$ is the expectation of X, what does we really "mean"?
Let X be the outcome on throwing a dice.
$E(X)$ refers to the "ensemble average" of X.
That is, to approximate $E(X)$, we consider an ensemble containing fairly large number of X, and then take the average.
In this case, for instance, we take 1,000 dices & throw them at the same time, add all the outcome & an estimate of $E(X)$ will be the sum over 1000.
We can increase the accuracy of our estimate by increasing the size of the ensemble. And theoretically we get $E[X]$ if the ensemble size goes to infinity.
Apparently, ensemble average is only useful for theoretical interest.
In practice, instead, we will only have one dice & throw it repeatedly, let say 1000 times and then approximate $E[X]$ as the sum of outcomes over the # of throw (1,000). This latter case gives us a time average.
In general,

time average $\neq$ ensemble average.

If it is not immediately apparent to you, consider a "crooked" dice with outcome always 6 is mixed into ensemble. Since we expect the ensemble is pretty large, the crooked dice has little effect to the ensemble average, hence the average is still the same as if all dices are fair (i.e. 3.5). However, if the crooked dice is ever chosen to compute the time average. The time average will become 6 even with infinite number of throws

For a random process, if for any function

$$\underbrace{E[f(X_1, X_2, X_3, \cdots X_k)]}_{\text{Ensemble Average}} = \underbrace{\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} f(X_i, X_{i+1} \dots X_{i+k-1})}_{\text{time average}}$$

We call a random process ergodic.

Lossless source coding

Read Ch 5.1 – 5.6

N.B. the log(·) we use here is always based 2 unless specified

How much info in a Webster dictionary?

A natural way to quantify info is to see how "far" we can compress it?

Eg. Assume that we have a very simple language with only 4 chars a, b, c, d

& the prob. of occurance for a, b, c, d are 0.1, 0.2, 0.3, 0.4

For a message of 1000 chars/samples long, how much info in there (in terms of bits)?

Compression, a.k.a. source coding, is generally done by mapping the source to coded sequences.

For example, we can choose

$$\left.\begin{array}{l} a \to 000 \\ b \to 001 \\ c \to 01 \\ d \to 1 \end{array}\right\} C1$$

then on average, a 1000 sample long message will be compressed into

$$1000 \cdot [0.1 \cdot 3 + 0.2 \cdot 3 + 0.3 \cdot 2 + 0.4 \cdot 1]$$
$$= 1900 \text{ bits}$$

Or 1.9 bits per sample (source rate / average code length)

Note that in binary, we need 2 bits to represent 4 chars

so it will be 2 bits without compression.

How did we choose the code C1? We will explain it more later on. But first realize that we should always assign shorter codewords to more probable char & vice versa. (Is it intuitive?)

For example, if we map

$a \to 1$
$b \to 001$
$c \to 01$
$d \to 000$

the average code length $= [0.1 \cdot 1 + 0.2 \cdot 3 + 0.3 \cdot 2 + 0.4 \cdot 3]$
$= 2.5$ bits per char

Before going further, let's define $\mathcal{X}$ as the source alphabet & $c(x)$ is the codeword for char $x \in \mathcal{X}$

What is minimum requirement for the code?

Apparently, codeword should uniquely represent a char

ie. $c(x) \neq c(x')$ if $x \neq x'$ — (1)

Codes satisfy (1) are "non-singular".

Codewords may not be uniquely decoded even if the code is "non-singular".

Consider
$a \to 0$
$b \to 1$
$c \to 01$
$d \to 10$

The code is "non-singular" but "10" can be decoded as d, or ba

In general, a code is "uniquely decodable" only if its extension is "non-singular", where an extension of a code is defined by

$$c(x_1 x_2 x_3 \ldots x_n) = c(x_1)\, c(x_2)\, c(x_3) \cdots c(x_n).$$

[ie. $c(ab) = c(a)\,c(b) = 000001$ for code C1]

For practical application, even uniquely decodable codes may not be enough. For example,

$a \to 10$
$b \to 00$
$c \to 11$
$d \to 110$

Although it is uniquely decodable, for coded sequence 110010 (= c(cbb)), the decoder cannot decode the character c right away reading the first 2 bits, actually, the decoder needs to read till the end to decode c.

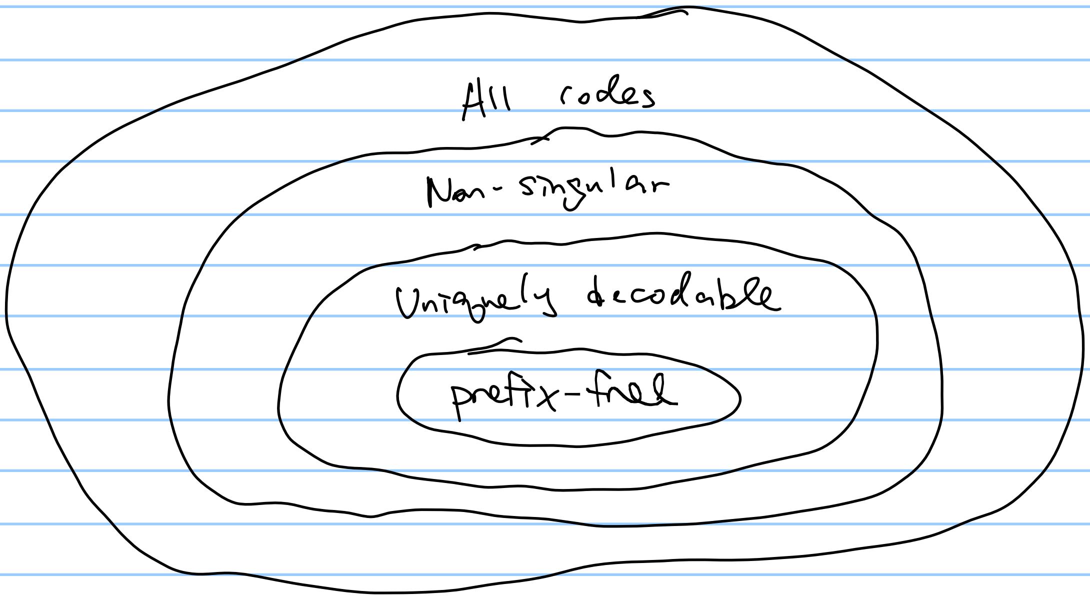
Instead, consider code C1 again, chars in any sequence can be decoded at earliest possible because no codeword is a prefix of any other codeword. The code is self-punctuating.

$$\text{(E.g. } \underset{\substack{\downarrow \\ d}}{1}\;\underset{\substack{\downarrow \\ c}}{01}\;\underset{\substack{\downarrow \\ a}}{000}\;\underset{\substack{\downarrow \\ b}}{001}\text{)}$$

We call codes that does not contain code ~~word~~ prefix of other codewords "prefix-free code" or "instantaneous code".

Apparently, prefix-free codes are uniquely decodable.

In summary, classes of codes are organized as follows



The questions immediately follow are :

① How can we construct optimum / good uniquely decodable code (or more preferably prefix code) for a source?
Ans: Huffman coding / Shannon-Fano-Elias coding / Arithmetic coding

② What is the average length of optimum uniquely decodable code?
Note that this length naturally quantifies the amount of info in the source.
Ans: Entropy of the Source.

Revisit our prefix code C_1

$a \to 000$	
$b \to 001$	
$c \to 01$	
$d \to 1$	

Apparently, we can increase the length of codewords by padding extra bits but still be instaneous code. However, it is not true if we try to shorten them. For example if we change $a \to 000$ to $a \to 00$. It is a prefix-free code anymore. It is always favorable to shorten the codewards (to compress the source more) but then a prefix code with the defined code length may not exist.

In general, what are the constraint on the codeword lengths for prefix-free code? Put it another way, given $\ell_1, \ell_2 \cdots \ell_n$, is it possible to construct prefix-free code with codeword length $\ell_1, \dots$ & ℓ_n.

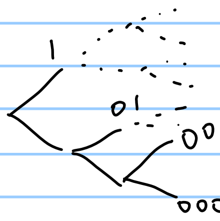
Kraft inequality gives a concise & precise answer.

Thm 1. (Kraft inequality, Thm 5.2.1 of Cover)

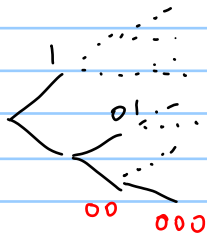
A prefix code exists iff the lengths of the corresponding codewords, $\ell_1, \ell_2, \cdots \ell_m$ s.t.

$$\sum_{i=1}^{m} 2^{-\ell_i} \leq 1 .$$

Pf: The proof is very intuitive, we can express any code as a tree with each codeword as a tree node. Then to be prefix-free, the ancestors of any code word tree node cannot be codewords themselves.



prefix-free



not prefix-free

(000 is decendant of both 000 & 00)

Let ℓ_{max} be the maximum code word length. For any length ℓ codeword, it has $2^{\ell_{max}-\ell}$ codewords of length ℓ_{max} with the codeward as prefix. If the code is prefix-free, those "decendants" cannot overlap. Therefore

$$\sum 2^{\ell_{max}-\ell} \leq \text{total \# of possible codewords with length } \ell_{max} = 2^{\ell_{max}}$$

$$\Leftrightarrow \quad \sum 2^{-\ell} \leq 1 .$$

Conversely, if Kraft inequality is satisfied, we can always construct the tree (thus the code) as in the figure intuitively ∎

Since all prefix code are also uniquely decodable, if codeword lengths satisfy Kraft inequality, a uniquely decodable code exists.

It turns out the converse is also true. For any uniquely decodable codes, Kraft inequality is satisfied. This is stated in the following theorem.

Thm 2 (McMillan, Thm 5.5.1 of cover)

A uniquely decodable exists iff the corresponding codeword lengths satisfy Kraft inequality.

i.e. $\sum 2^{-l} \leq 1$

Pf. The "if" (converse) part comes directly from Thm 1.

For the "only if" (forward) part,

Consider any message with k characters, note that

$$\left(\sum_{x\in\mathcal{X}} 2^{-l(x)}\right)^k = \left(\sum_{x_1\in\mathcal{X}} 2^{-l(x_1)}\right)\left(\sum_{x_2\in\mathcal{X}} 2^{-l(x_2)}\right)\left(\sum_{x_3\in\mathcal{X}} 2^{-l(x_3)}\right)\cdots$$

$$= \sum_{x_1,x_2\cdots x_k\in\mathcal{X}^k} 2^{-[l(x_1)+l(x_2)+\cdots l(x_k)]}$$

$$= \sum_{x^k\in\mathcal{X}^k} 2^{-l(x^k)} \quad \text{where } x^k = x_1x_2\cdots x_k$$

$$= \sum_{m=1}^{kl_{max}} a(m)2^{-m},$$

where $a(m)$ is # of codewords with length m.

However, for the code to be uniquely decodable, $a(m) \leq 2^m \leftarrow$ available codewords of length m

$$\therefore \left(\sum_{x\in\mathcal{X}} 2^{-l(x)}\right)^k \leq kl_{max}$$

$$\Leftrightarrow \sum_{x\in\mathcal{X}} 2^{-l(x)} \leq (kl_{max})^{\frac{1}{k}}$$

Note that the inequality has to be true for any k, so as we take $k\to\infty$, we have the RHS $= (kl_{max})^{\frac{1}{k}} = 1$, which gives us the Kraft inequality

$\blacksquare$

We are now ready to answer one of the fundamental question of IT: "How do we quantify the amount of info for a probabilistic source?"
$\equiv$ "How much we can compress losslessly a probabilistic source?"

If a source output character x_i with prob. p_i,

Consider mapping x_i to $c(x_i)$ & the codeword length $|c(x_i)| = l_i$

Average codeword length $L = \sum_i p_i l_i$

However, l_i is not arbitrary as Kraft inequality should be satisfy to have the code to be uniquely decodable.
i.e. $\sum_i 2^{-l_i} \leq 1$ —— (1)

l_i should be integers in practice but lets neglect this constraint. Intuitively doing this should still give us a pretty good estimate of $E[L]$.
Note that then constraint (1) will be satisfied at the optimal point.

Otherwise, let $(l_1^*, l_2^* \ldots)$ be the optimum lengths & $\sum_i 2^{-l_i^*} < 1$ apparently, we can set l_i^* to $l_i^* - \Delta$ for a very small Δ & still get $\sum_i 2^{-l_i^*} < 1$ but the average length $E[L]$ decreases.

So we can consider (1) as a equality constraint, i.e. $\sum_i 2^{-l_i} = 1$

It is a constraint optimization problem, we can use the standard Langrange Multiplier techique,
i.e. we want to have $\frac{\partial J}{\partial l_i} = 0$

with $J = \sum p_i l_i + \lambda \left(\sum 2^{-l_i} - 1 \right)$ $\left[2^{-l_i} = e^{-(\ln 2) l_i} \right]$

This gives us $p_i - \lambda 2^{-l_i^*} \ln 2 = 0$

$$\Rightarrow 2^{-l_i^*} = \frac{p_i}{\lambda \ln 2}$$

Sub this to constraint, $\sum_i 2^{-l_i^*} = 1$

$$\Rightarrow \sum \frac{p_i}{\lambda \ln 2} = 1 \Rightarrow \lambda = \frac{1}{\ln 2}$$

$\therefore$ $2^{-l_i^*} = p_i$

$\Rightarrow l_i^* = -\log p_i$ $\swarrow$ this suggests the length of codeword to be $-\log p_i$

$\therefore$ Optimum length $L^* = \sum p_i l_i^* = -\sum_i p_i \log p_i$ bits $\triangleq H(X)$

$H(X) = -\sum_i p_i \log p_i$ is defined as the entropy of the source. It specifies how many bits per char the source can be compressed optimally.

Let us put back the integer constraint to the picture.

For each symbol, the codeword length is set to $\lceil -\log p(x) \rceil$ instead. Then we have

$$E[l^*(X)] \leq \sum_{x \in \mathcal{X}} p(x) \lceil -\log p(x) \rceil$$
$$\leq \sum_{x \in \mathcal{X}} p(x) [-\log p(x) + 1]$$
$$= H(X) + 1$$

i.e. we can compress within one bit from the entropy.

On the other hand, as long as a code s.t. Kraft equality

$$E[l(X)] - H(X) = \sum_{x \in \mathcal{X}} p(x) l(x) + \sum_{x \in \mathcal{X}} p(x) \log p(x)$$
$$= -\sum_{x \in \mathcal{X}} p(x) \log \frac{2^{-l(x)}}{p(x)}$$

$-\log x \geq -x + 1$

$$\geq \sum_{x \in \mathcal{X}} p(x) \left(-\frac{2^{-l(x)}}{p(x)} + 1 \right)$$
$$= -\sum_{x \in \mathcal{X}} 2^{-l(x)} + \sum_{x \in \mathcal{X}} p(x)$$

$\left(\sum_{x \in \mathcal{X}} 2^{-l(x)} \leq 1 \right)$ $\uparrow$ Kraft

$$\geq -1 + 1 = 0$$

In summary, $H(X) \leq E(l^*(X)) \leq H(X) + 1$

N.B. Although $H(X)$ is actually a function of the probabilistic distribution of X ($p(x)$), it is customary to write it as a function of the random variable X.

Huffman Coding

Given the distribution of the source, how do we construct the optimal code?

Ans: Huffman Coding

We will illustrate the code construction through an example

Eg. Prob. for $\left.\begin{matrix} a \\ b \\ c \\ d \end{matrix}\right\}$ are $\begin{matrix} 0.1 \\ 0.2 \\ 0.3 \\ 0.4 \end{matrix}$ (same as our original example)

1). Combine probs

Combine the smallest two probs ↓ repeat

$$\begin{tikzcd}
p(a)=0.1 \arrow[r, dash] & p(a\cup b)=0.3 \arrow[dr, dash] & & \\
p(b)=0.2 \arrow[ur, dash] & & p(a\cup b\cup c)=0.6 \arrow[r, dash] & p(a\cup b\cup c\cup d)=1 \\
p(c)=0.3 \arrow[urr, dash] & & & \\
p(d)=0.4 \arrow[uurrr, dash] & & &
\end{tikzcd}$$

2). Trace back to generate code words

generate codeword in this direction ←

$$\begin{tikzcd}
p(a)=0.1 \arrow[r, dash, "0"] & p(a\cup b)=0.3 \arrow[dr, dash, "0"] & & \\
p(b)=0.2 \arrow[ur, dash, "1"'] & & p(a\cup b\cup c)=0.6 \arrow[r, dash, "0"] & p(a\cup b\cup c\cup d)=1 \\
p(c)=0.3 \arrow[urr, dash, "1"'] & & & \\
p(d)=0.4 \arrow[uurrr, dash, "1"'] & & &
\end{tikzcd}$$

$\Rightarrow$

a	→	000
b	→	001
c	→	01
d	→	1

It is apparent that Huffman code is prefix-free from its construction. And Huffman coding is optimum in the sense that if C is the Huffman code & C' is an arbitrary uniquely decodable code.

$$L(C) \le L(C')$$

↑ average length

Note that codeword length is approximately $-\log(p(x))$ through the construction. (Convince yourself with a few examples with probs of the form 2^{-m}) This agrees with the optimum code length in the previous discussion.

We will try to show the optimality of Huffman codes through mathematical induction. But before that, let's consider the conditions for optimality.

If a code C is optimal then
1. The corresponding tree is full (Kraft inequality is satisfied with equality)
2. If $p(x) > p(y) \Rightarrow C(\ell(x)) \leq C(\ell(y))$

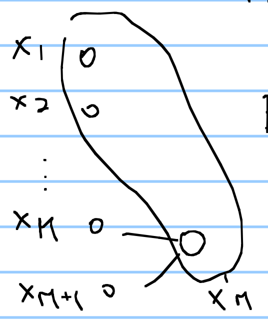
It is not hard to argue that they are the sufficient conditions as well.

Proof of optimality of Huffman code:
Consider source of alphabet size of 2,
The constructed code by Huffman will be $\{1, 0\}$
Apparently it is optimum.

Now, consider source with alphabet size of 3. The Huffman code will be $\{1, 01, 00\}$ or $\{0, 10, 11\}$. Apparently it is optimal as well.

Now, assume that the Huffman construction will result in an optimum code for alphabet size less than or equal to M

Now, consider source with alphabet size M+1, WLOG, assume x_M & x_{M+1} with the smallest probabilities.



By Huffman construction, x_M & x_{M+1} merge into x'_M & a Huffman code C' is constructed for $x_1, x_2, x_3 \cdots x'_M$ (the combined node of x_M & x_{M+1})

Since C' is optimum, the tree corresponding to C' is full, and thus the tree corresponding to C is full as well. Therefore, condition 1 is satisfied.

Moreover Condition 2 is satisfied by C', therefore for all $x \in \{x_1 \dots x_{M-1}\}$ the condition is also satisfied by C.
So we just need to make sure that $C(x_M)$ & $C(x_{M+1})$ are the longest codeword. Assuming that it is not true, then, there exists $x_a \notin \{x_M, x_{M+1}\}$ as the longest code word ($l(x_a) > l(x_M)$). But C' is optimum. So $\exists\, x_b \notin \{x_M x_{M+1}\}$ s.t. $C(x_a) = C'(x_a) = C'(x_b) = C(x_b)$ since the C' corresponds to a full tree.

Now $l(C(x_a)) > l(C(x_M)) \Rightarrow l(C(x_a)) > l(C'(x'_M)) + 1$
Since x_a, x_b are longest codewords in C', by optimality of C', they have smallest probabilities. So in the next Huffman step x_a & x_b are combined (to x''_c, let say) & Huffman code C'' is found for $x_1 \dots x''_c \dots x'_M$ with alphabet size $M-1$

By assumption, C'' is optimum as well. But we have

$$l(C(x_a)) > l(C'(x'_M)) + 1$$

note that $(C'(x'_M) = C''(x'_M))$

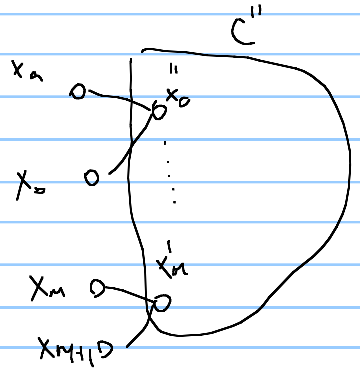
$$\Rightarrow l(C''(x''_c)) + 1 > l(C''(x'_M)) + 1$$

$$\Rightarrow l(C''(x''_c)) > l(C''(x'_M))$$

But $Pr(x''_c) = Pr(x_a) + Pr(x_b) > Pr(x_M) + Pr(x_{M+1}) = Pr(x'_M)$

This contradicts with the optimality assumption of C''
∴ x_M & x_{M+1} have to be the longest code words in C.
Therefore, condition 2 satisfied & the code C is optimal wrt the source.

∎



$C''(x''_c)$ shouldn't be longer than $C''(x'_M)$

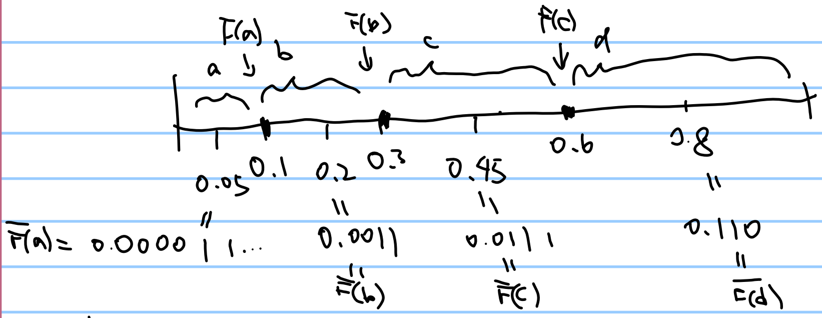
$$\begin{tikzcd}
 & & \text{optimal wrt source} & \\
 & \text{optimal / minimal} \arrow[ur] & & p(a)>p(b) \Rightarrow l(a) \le l(b) \arrow[ul] \\
\text{Prefix-free} \arrow[ur, "\text{full tree}"] \arrow[d, "\text{trivial}"'] & & \text{Huffman} \arrow[ul] \arrow[ur] & \\
\text{Uniquely decodable} \arrow[rr, "\text{McMillan}"] & & \text{Kraft inequality} \arrow[ull, "\text{Kraft}"'] &
\end{tikzcd}$$

Shannon - Fano - Elias Coding

We will introduce Shannon-Fano-Elias Coding here. It is not an optimum code. But it is more intuitive than Huffman coding and is the basis for arithmetic coding.

Consider the same prob distribution
$p(a) = 0.1$
$p(b) = 0.2$
$p(c) = 0.3$
$p(d) = 0.4$

Write them in a cumulative way, define $F(x_i) = \sum_{j=1}^{i} p(x_j)$ & $\bar{F}(x_i) = F(x_i) - p(x_i)/2$ as shown in the figure below.



We can express a, b, c, d by the fractional parts of the $\bar{F}(a)$, $\bar{F}(b)$, $\bar{F}(c)$ & $\bar{F}(d)$.

If we pick $l(x)$ bits out of the fractional part to represent the region from $0.b_1b_2 \cdots b_{l(x)}$ to $0.b_1b_2 \cdots b_{l(x)}1$, where b_i are bits in the fractional part,

We have the distance b/w lower boundary to the mid point $\bar{F}(x)$
$= 0.b_1b_2 \cdots b_{l(x)} b_{l(x)+1} \cdots - 0.b_1b_2 \cdots b_{l(x)} = 0.00 \cdots 0\, b_{l(x)+1} \cdots$
$\leq 2^{-l(x)}$

Similarly, we also have the distance b/w upper boundary to the midpoint
$= 0.b_1b_2 \cdots b_{l(x)}111 \cdots - 0.b_1b_2 \cdots b_{l(x)} b_{l(x)+1} \cdot b_{l(x)+2} \cdots$
$= 0.00 \cdots (1 - b_{l(x)+1})(1 - b_{l(x)+2}) \cdots \leq 2^{-l(x)}$

So if we pick $l(x) = \lceil -\log p(x) + 1 \rceil$, we have the specified region lies inside the the corresponding interval of x.

Specifically,
$l(a) = \lceil -\log 0.1 + 1 \rceil = 5$
$l(b) = \lceil -\log 0.2 + 1 \rceil = 4$
$l(c) = \lceil -\log 0.3 + 1 \rceil = 3$
$l(d) = \lceil -\log 0.4 + 1 \rceil = 3$

As $\bar{F}(a) = 0.00001$
$\bar{F}(b) = 0.0011$ therefore
$\bar{F}(c) = 0.011$
$\bar{F}(d) = 0.110$

$a \to 00001$
$b \to 0011$
$c \to 011$
$d \to 110$

And the codewords specify the intervals as follows

$a = [0.00001, 0.00001\underset{\cdot}{1}] = [0.00001, 0.0001) = [0.03125, 0.0625)$

$b = [0.0011, 0.0011\underset{\cdot}{1}] = [0.0011, 0.01) = [0.1875, 0.25)$

$c = [0.011, 0.01\underset{\cdot}{1}] = [0.011, 0.1) = [0.375, 0.5)$

$d = [0.110, 0.110\underset{\cdot}{1}] = [0.11, 0.111) = [0.75, 0.875)$

Note that the code has to be prefix code as those intervals cannot overlap from code construction.

For example if we have codeword that is a prefix of the other, let say 0.11 d 0.11101

note that 0.11 characterises the interval $[0.11, 0.11\underset{\cdot}{1}]$ which overlaps (includes) that of 0.11101.

Moreover we have

$$L(S\text{-}F\text{-}E) = \sum p(x)\left(\lceil -\log_2 p(x) \rceil + 1\right) \leq -\sum p(x)\log p(x) + 2 = H(X) + 2$$

i.e. the average code length is not more than 2 bits away from the optimum.

As for Huffmann codes,

note that if we let $l(x) = \lceil -\log_2 p(x) \rceil$, $l(x)$ will satisfy Kraft inequality as

$$\sum 2^{-l(x)} = \sum 2^{-\lceil -\log_2 p(x) \rceil} \leq \sum 2^{-(-\log_2 p(x))} = \sum p(x) = 1$$

$\therefore$ An prefix code with the specified code lengths exist.

And $L' = \sum p(x)\, l(x) = \sum p(x) \lceil -\log_2 p(x) \rceil$

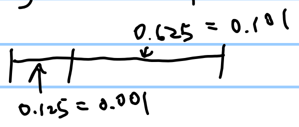
$\leq \sum p(x)\left(-\log_2 p(x) + 1\right) = H(X) + 1$

$\because$ Huffman coding is optimum, the average code length

$$L(\text{Huffman}) \leq L' \leq H(X) + 1$$

Indeed, S-F-E code can be a little bit silly sometimes,

let $p(a) = 0.25$, $p(b) = 0.75$



S-F-E: $a \to 001$
$b \to 10$

But of course the optimum code is just $a \to 0$, $b \to 1$

Arithmetic coding

Although Huffman coding is "optimal", as its code length needs to be integer, loss is introduced when the optimal code length $-\log p(x)$ is not an integer. The loss is exacerbate when the alphabet size is small. In the extreme case, when we have only 2 chars, No compression can be achieved by coding.

For example $p(a) = 0.25$
$p(b) = 0.75$, the best code is simply no coding at all, i.e. $a \to 1$, $b \to 0$

One way to alleviate this issue is to combine several code symbol before applying coding. That is, instead of designing code to

$p(a) = 0.25$
$p(b) = 0.75$, design code for
$p(aa) = \frac{1}{16}$
$p(ab) = 3/16$
$p(ba) = 3/16$
$p(bb) = 9/16$

Note that in this way $l(X_1, X_2) \le H(X_1, X_2) + 1 = 2H(X) + 1$

& average length per char $= \frac{l(X_1, X_2)}{2} \le H(X) + \frac{1}{2}$

[Note that we have $H(X_1, X_2) = H(X_1) + H(X_2)$ assuming that the source outputs are independent at different time instances
This is reasonable if we recall $H(\cdot)$ measure the amount of information in the argument. So for 2 indep events, X, Y, $H(X, Y) = H(X) + H(Y)$. We will give a rigorous proof for this in later lectures]

In general we can group more inputs together before coding, for n inputs, we have

average length per char $\le H(X) + \frac{1}{n}$

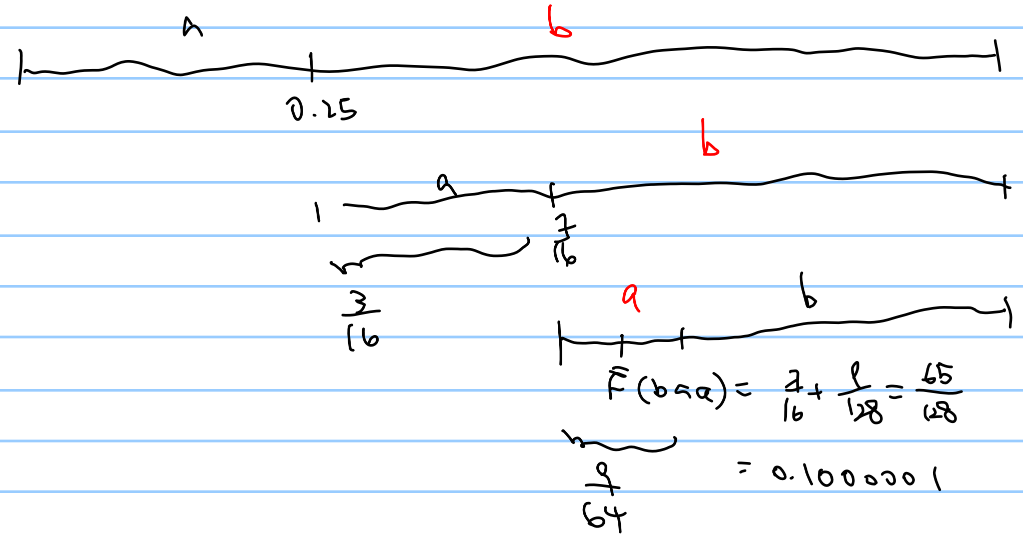
But how can we do it efficiently?

Arithmetic coding is an ingenious way to group & code practically unlimited # of chars together. Therefore it can virtually achieve the entropy rate $H(x)$!!

The encoding & decoding is rather complicated. Here we will only give the intuitive idea. You are welcome to choose to implement arithmetic coding as your course project.

Arithmetic coding is based on S-F-E coding, consider distribution
$p(a) = 0.25$
$p(b) = 0.75$, we will demonstrate the basic idea by coding

bba.



$\bar{F}(baa) = \frac{7}{16} + \frac{1}{128} = \frac{65}{128}$

$= 0.1000001$

Use S-F-E on baa, as $l(baa) = \lceil \log \frac{1}{p(baa)} + 1 \rceil = 4$ $\therefore$ $c(baa) = 1000$

And the corresponding interval is $[0.1000, 0.1001) = [0.5, 0.5625)$

Asymptotic Equal Partition (AEP)

Read Ch. 3

In previous lectures, we quantify the amount of info from a probabilistic source as

$$H(X) = \sum_{x \in \mathcal{X}} -p(x) \log p(x)$$

by arguing that the source can be compressed at most into $H(X)$ bits per sample.

Let us look at an example.

What is the amount of info obtained from rolling a dice?

$\mathcal{X} = \{1,2,3,4,5,6\}$ & $p(x) = 1/6$ for all x

$\therefore \quad H(X) = \sum_{x \in \mathcal{X}} \frac{1}{6} \log_2 6 = \log_2 6$ bits

It makes quite a lot of sense intuitively, since we have 6 different outcome, we expect the amount of info $= \log_2$ # outcome $= \log_2 6$ bits.

The expression ~~for~~ $H(X)$ may seem obscure at first. I hope this should become intuitive now. We can assume that, for any $x \in \mathcal{X}$, the amount of info contained by $x = \log_2 \frac{1}{p(x)}$ & since x only occurs with a probability $p(x)$ $\therefore$ the average amount of info will be $\sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)}$ as we obtained previously.

We will give the third reason why $H(X)$ is as defined by studying the sequences of outcomes obtained from the source. It turns out that for most of the time, the resulting sequence share some identitical characteristics. We call these sequences typical sequences. While "untypical" sequences may also occur, the chance of getting them is negligibly small.

Consider an i.i.d. source X with distribution $p(x)$, we sample the source repeatedly to get a length-N sequence $x_1, x_2 \cdots x_N$

Consider any function $f(\cdot)$, if we evaluate the average of $f(\cdot)$ over the samples of any output sequence $x_1, x_2, x_3 \cdots x_n$

i.e. $\frac{1}{N}\sum_{i=1}^{N} f(x_i)$

this average will approach to the expectation $E(f(X))$ for large N

(by the Law of Large Number)

A quick proof of LLN:

By Chebyshev's inequality: $\Pr(|X-\bar{X}| \geq a) \leq \frac{Var(X)}{a^2}$

let $Y = \frac{1}{N}(\underset{\uparrow}{X_1} + \cdots \underset{\uparrow}{X_N})$ i.i.d $\quad E(Y) = E(X) = \bar{X}$

$Var(Y) = Var(X)/N$ (why?)

$\Rightarrow \quad \Pr(|Y-\bar{Y}| \geq a) \leq \frac{Var(X)}{N a^2} \to 0 \quad$ as $\quad N \to \infty$

Proof of Chebyshev: First let's prove the Markov inequality

$\Pr(X \geq b) \leq \frac{E(X)}{b}$ if X is a +ve r.v.

$X = I(X \geq b)\cdot X + I(X<b)\cdot X \geq I(X\geq b)\cdot b$

$\Rightarrow \quad E(X) \geq \Pr(X \geq b)\cdot b$

Now by Markov's inequality, take $X = (Y-\bar{Y})^2$ & $b = a^2$

$\Rightarrow \quad E((Y-\bar{Y})^2) \geq \Pr((Y-\bar{Y})^2 \geq a^2)\cdot a^2$

$\Rightarrow \quad \Pr(|Y-\bar{Y}| \geq a) \leq \frac{Var(Y)}{a^2}$

■

In particular, $\log_2 p(x)$ is a function of x & so for any output sequence $x_1, x_2 \cdots x_N$ from the iid source

$$\frac{1}{N}\sum_{i=1}^{N} \log_2 p(x_i) \to E[\log_2 p(X)] = -H(X)$$

$$\Rightarrow \quad 2^{\sum_{i=1}^{N} \log_2 p(x_i)} \to 2^{-NH(X)}$$

$$\Rightarrow p(x_1, x_2 \cdots x_N) \overset{(a)}{=} \prod_{i=1}^{N} p(x_i) \to 2^{-NH(X)}$$

where (a) is due to that the source is independent over different time instance.

We define all the sequences that approach the prob. $2^{-NH(X)}$ as typical sequence

More precisely, a typical set $A_\epsilon^N(X)$ of length-N sequences from iid source X for some small finite ϵ is

$$A_\epsilon^N(X) = \{x_1, x_2 \cdots x_N \mid 2^{-N(H(X)+\epsilon)} \le p(x_1, x_2 \cdots x_N) \le 2^{-N(H(X)-\epsilon)}\}$$

By the law of large number, for any finite ϵ, & as $N \to \infty$,

$$\Pr(X_1, X_2 \cdots X_N \in A_\epsilon^N(X)) = 1$$

Or in general for any finite δ, we can find a sufficiently large N s.t.

$$\Pr(X_1, X_2 \cdots X_N \in A_\epsilon^N(X)) \ge 1 - \delta$$

How many typical sequence do we have ?

Since almost all sequences are typical & each of them has equal probability $\sim 2^{-NH(X)}$

$$\therefore \quad \# \text{ typical sequence} \sim 2^{NH(X)}$$

More precisely, as $\Pr(x_1 \cdots x_N \in A_\epsilon^N(X)) \leq 1$ & if $x_1, x_2 \cdots x_n \in A_\epsilon^N(X)$

$$p(x_1 \cdots x_N) \geq 2^{-N(H(X)+\epsilon)}$$

$$\therefore 1 \geq \Pr(X_1 \cdots X_N \in A_\epsilon^N(X)) = \sum_{x^N \in A_\epsilon^N(X)} p(x^N)$$

$$\geq |A_\epsilon^N(X)| \, 2^{-N(H(X)+\epsilon)}$$

$$\therefore \quad |A_\epsilon^N(X)| \leq 2^{N(H(X)+\epsilon)}$$

Similarly, since $\Pr(X_1 \cdots X_N \in A_\epsilon^N(X)) \geq 1-\delta$

$$\therefore \quad 1-\delta \leq \Pr(X_1 \cdots X_N \in A_\epsilon^N(X)) = \sum_{x^N \in A_\epsilon^N(X)} p(x^N)$$

$$\leq |A_\epsilon^N(X)| \, 2^{-N(H(X)-\epsilon)}$$

$$\therefore \quad |A_\epsilon^N(X)| \geq (1-\delta) 2^{N(H(X)-\epsilon)}$$

In summary, $(1-\delta)2^{N(H(X)-\epsilon)} \leq |A_\epsilon^N(X)| \leq 2^{N(H(X)+\epsilon)}$

Why "AEP" ?

As we have seen, for any iid source, the possible output sequences are virtually all typical and have equal probability (hence equal partition) & this is only true when length of sequence N is sufficiently large (hence asymptotic).

Now, what is the information rate coming out of the iid source, we know there are $\sim 2^{NH(X)}$ sequences & hence $NH(X)$ bits information per N samples. Therefore, there are $H(X)$ bits per sample coming out of the source. This agree with the interpretation that the entropy $H(X)$ quantifies amount of info from a source.

Jointly typicality

Two sequences x^n, y^n are jointly typical if

$$x^n \in A_\epsilon^n(X) \quad \& \quad y^n \in A_\epsilon^n(Y)$$

$$\& \quad 2^{-n(H(X,Y)+\epsilon)} \leq p(x^n, y^n) \leq 2^{-n(H(X,Y)-\epsilon)}$$

& we denote $(x^n, y^n) \in A_\epsilon^n(X,Y)$

Note that for sequences (x^n, y^n) coming out of joint source $p(x,y)$, (x^n, y^n) is almost surely $\in A_\epsilon^n(X,Y)$

i.e. $\Pr((X^n, Y^n) \in A_\epsilon^n(X,Y)) = 1$ as $n \to \infty$ just as the case of typical sequences.

By the same argument, one can easily show that $|A_\epsilon^n(X,Y)| \approx 2^{nH(X,Y)}$

Information Measures

Read Ch 2.1 – 2.8, 4.2

We have introduced entropy $H(X)$ in the last lectures. For the coming few lectures, we will discuss some mathematical properties of $H(X)$. Moreover, we will introduce a few more information measures.

N.B. We only consider discrete random variables in this chapter. We will consider continuous random variable in later lectures.

Thm 1. $H(X) \geq 0$ (Lemma 2.1.1)

Pf: It follows directly from the definition that $H(x) = \sum_{x \in \mathcal{X}} -p(x) \log_2 p(x)$

Note that $0 \leq p(x) \leq 1 \quad \therefore 0 \leq -\log p(x) \leq \infty \quad \therefore 0 \leq -p(x) \log p(x) < \infty$

(Note that when $p(x) \to 0$, $p(x) \log p(x) = \frac{\log p(x)}{1/p(x)} = \frac{(\log p(x))'}{(1/p(x))'} = \frac{1/p(x)}{-1/p^2(x)} \to 0$)

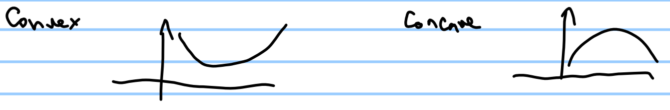
This result is intuitively true as it means the amount of info from a source cannot be smaller than 0.

Thm 2. $H(X) \leq \log |\mathcal{X}|$

Before we give the proof, let us introduce convexity, concavity & a famous Lemma.

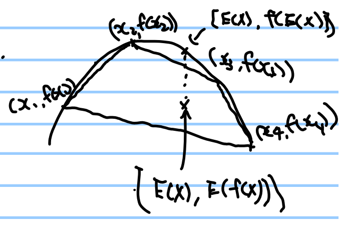
A function $f(\cdot)$ is convex if $\forall\ 0 \leq \lambda \leq 1$ & x_1, x_2,
$f(\lambda x_1 + (1-\lambda) x_2) \leq \lambda f(x_1) + (1-\lambda) f(x_2)$
If $f(\cdot)$ is differentiable, then $f''(x) \geq 0$.

A function $f(\cdot)$ is concave if $\forall\ 0 \leq \lambda \leq 1$ & x_1, x_2
$f(\lambda x_1 + (1-\lambda) x_2) \geq \lambda f(x_1) + (1-\lambda) f(x_2)$
If $f(\cdot)$ is differentiable, then $f''(x) \leq 0$



Lemma 1 (Jensen) (a) If $f(\cdot)$ is a convex function, then $f(E(X)) \leq E[f(X)]$
(Thm 2.6.2) And if $f(\cdot)$ is strictly convex, then equality hold only when $X = E(X)$ is a constant
(b) Similarly if $f(\cdot)$ is concave, then $f(E(X)) \geq E(f(X))$ & equality holds iff $X = E(X)$.

Pf. Proof for (a) + (b) are almost the same. We will show the case when $f(\cdot)$ is concave (b) here. As shown in the figure, let say X only has 4 different outcomes (the proof can be generalized intuitively to any # of outcomes)



Note that for any distribution of X,
$[E(X), E(f(X))] = [\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 + \alpha_4 x_4,$
$\alpha_1 f(x_1) + \alpha_2 f(x_2) + \alpha_3 f(x_3) + \alpha_4 f(x_4)]$
(for some $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \geq 0$ + $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$)
will be some point lied inside the polygon with $(x_i, f(x_i))$ as corners

Then apparently, if $f(\cdot)$ is concave, we have $f(E(X)) \geq E(f(X))$ from the figure

Moreover, if $f(\cdot)$ is strictly concave, we have $f(E(X)) = E(f(X))$ only if $E(X)$ lies at the corner point. That basically means that $X = E(X)$
(e.g. if $E(X) = x_2 \Rightarrow \alpha_1 = \alpha_3 = \alpha_4 = 0$ + $\alpha_2 = 1$, $X = x_2$ with prob. 1)
In other words, $X = x_2$, a constant

Pf of Thm 2: Back to Thm 2,

$$H(X) = E[-\log p(X)] = E\left[\log \frac{1}{p(X)}\right]$$

Since $\log(\cdot)$ is concave, by Jensen's Lemma,

$$E\left[\log \frac{1}{p(X)}\right] \leq \log E\left[\frac{1}{p(X)}\right] = \log \sum_{x \in \mathcal{X}} \frac{p(x)}{p(x)} = \log |\mathcal{X}|$$

We have equality only when $\frac{1}{p(x)}$ is constant. i.e. X is uniform.

Multiple variable & conditional Entropy

What is $H(X,Y)$? The amount of info for 2 variables?
Should $H(X,Y) = H(X) + H(Y)$?

Let's start from the basic definition.

$$H(X,Y) \triangleq \sum_{x,y \in \mathcal{X},\mathcal{Y}} -p(x,y) \log p(x,y)$$

but $p(x,y) = p(x)\, p(y|x)$

$$\therefore = \sum_{x,y \in \mathcal{X},\mathcal{Y}} -p(x,y) \log p(x) + \sum_{x,y \in \mathcal{X},\mathcal{Y}} -p(x,y) \log p(y|x)$$

$$= \sum_{x \in \mathcal{X}} -\log p(x) \underbrace{\sum_{y \in \mathcal{Y}} p(x,y)}_{p(x)} + \sum_{x,y \in \mathcal{X},\mathcal{Y}} -p(x,y) \log p(y|x)$$

$$= \underbrace{\sum_{x \in \mathcal{X}} - p(x) \log p(x)}_{H(X)} + \underbrace{\sum_{x,y \in \mathcal{X},\mathcal{Y}} -p(x,y) \log p(y|x)}_{H(Y|X)}$$

We define the second term $H(Y|X)$

i.e. $$H(Y|X) \triangleq \sum_{x,y \in \mathcal{X},\mathcal{Y}} -p(x,y) \log p(y|x) = \mathop{E}_{X,Y}[-\log p(Y|X)]$$

& we have $H(X,Y) = H(X) + H(Y|X)$

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

Note that $H(Y|X) = \sum_{x,y} -p(x,y) \log p(y|x) = \sum_x p(x) \sum_y -p(y|x) \log p(y|x)$
$= \sum_x p(x) \; H(Y|X=x)$

where $H(Y|X=x) = \sum_y -p(y|x) \log p(y|x)$

So we can interpret $H(Y|X=x)$ as the amount of info of Y given side information X with realization x
& $H(Y|X)$ as the amount of info of Y "on average" given side info X

When X & Y are independent, we expect $H(Y|X) = H(X)$
(we cannot gain any info from X)

Indeed, $H(Y|X) = \sum_{x,y} -p(x,y) \log p(y|x) = \sum_{x,y} -p(x,y) \log p(y)$
$= \sum_y -\log p(y) \sum_x p(x,y)$
$= \sum_y -p(y) \log p(y) = H(Y)$

(Note that if X, Y indep, $p(x,y) = p(x)p(y)$, but $p(x,y) = \frac{p(x)}{p(y|x)}$
$\therefore p(y|x) = p(y)$)

<u>Relative entropy</u>
Given two distributions, how can we quantify how much difference b/t them?

One way is via relative entropy (aka. Kullback-Leibler distance, information divergence)
Relative entropy of $q(x)$ from $p(x)$ is

$$D(p(x)\|q(x)) \triangleq \sum_x p(x) \log \frac{p(x)}{q(x)}$$

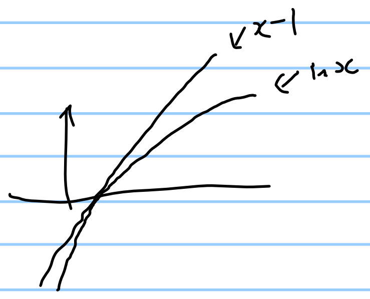
Note that relative entropy is not symmetric. That is,

$D(p(x)\|q(x)) \neq D(q(x)\|p(x))$ (hence not a true distance metric)

Thm 3: $D(P\|Q) \geq 0$ for any distribution P & Q & $D(P\|Q) = 0$ iff P & Q are identical.

To show Thm 3, we will show a very simple Lemma as follows.

Lemma 2 : $\ln x \le x-1$ & equality holds only if $x=1$, where $\ln(\cdot)$ is natural log



Pf: The proof is very simple. Note that log is a concave function
i.e. $(\ln x)'' = \left(\frac{1}{x}\right)' = \frac{-1}{x^2} < 0$

And at $x=1$, $(\ln x)' = (x-1)' = 1$ & $\ln x = (x-1) = 0$. So as shown in the figure, $\ln x$ will be strictly below $x-1$ everywhere except at $x=1$ ∎

~~Pf~~ of Thm 3 :

For any $p(x)$ & $q(x)$, $D(p(x)\|q(x)) = \sum_x p(x) \log \frac{p(x)}{q(x)}$

$= -\sum_x p(x) \log \frac{q(x)}{p(x)} = -\frac{1}{\ln 2} \sum_x p(x) \ln \frac{q(x)}{p(x)}$

$\overset{(a)}{\ge} -\frac{1}{\ln 2} \sum_x p(x) \left(\frac{q(x)}{p(x)} - 1 \right)$ (by Lemma 2) $\left(\begin{array}{l} \text{N.B. } a \ge b \\ \Rightarrow -b \ge -a \end{array} \right)$

$= -\frac{1}{\ln 2} \left(\overset{1}{\cancel{\sum_x q(x)}} - \overset{1}{\cancel{\sum_x p(x)}} \right) = 0$ ∎

Note that the equality in (a) only holds when $\frac{q(x)}{p(x)} = 1$ for all x. That is P & Q are identical.

Alt pf for Thm 2 :

We can show Thm 2 very easily with Thm 3.

For any X & distribution $p(x)$, let $q(x) = \frac{1}{|\mathcal{X}|}$ for all $x \in \mathcal{X}$

Then by Thm 3, $D(p(x)\|q(x)) \ge 0$

$\therefore \sum_x p(x) \log \frac{p(x)}{q(x)} \ge 0 \Rightarrow \sum_x p(x) \log p(x) - \sum_x p(x) \log q(x) \ge 0$

$\Rightarrow -\sum_x p(x) \log q(x) \ge -\sum_x p(x) \log p(x)$

(N.B. $\sum_x p(x) \log q(x)$ & $\sum_x p(x) \log p(x) \le 0$)

$\Rightarrow -\sum_x p(x) \log \frac{1}{|\mathcal{X}|} \ge H(X)$

$\Rightarrow \log |\mathcal{X}| \ge H(X)$ ∎

Let's look at conditional entropy $H(X|Y)$ again. It quantifies the amount of info of X given Y is known. Intuitively we expect that $H(X) \geq H(X|Y)$. That is, the amount of info from X will be more if side info is not given.

Indeed we can easily show that with Thm 3.

Thm 4. $H(X) \geq H(X|Y)$

Pf. $$H(X) - H(X|Y) = \sum_x -p(x)\log p(x) + \sum_{xy} p(x,y) \log p(x|y)$$
$$= \sum_{xy} p(x,y) \log \frac{p(x|y)}{p(x)} = \sum_{xy} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)}$$
$$= D(p(x,y) \| p(x)p(y)) \geq 0 \text{ (by Thm 3)}$$ ∎

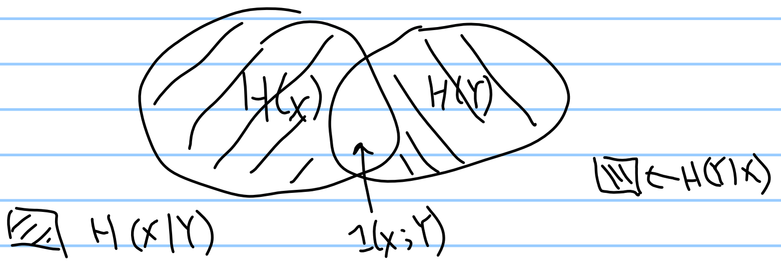
Mutual information

Define mutual information $I(X;Y) \triangleq H(X) - H(X|Y)$
$$= D(p(x,y) \| p(x)p(y)) \text{ (from Th 4)}$$

Note that $I(X;Y)$ is symmetric, i.e. $I(X;Y) = I(Y;X)$

$I(X;Y)$ can be interpreted as the amount of info share by X + Y.
Or amount of information gain (decrease in uncertainty) regarding X when Y is known.
Interesting $I(X;Y)$ is symmetric, so it is also equal to the amount of information gain regarding Y when X is known.
By Thm 4, $I(X;Y) \geq 0$. And when X + Y are indep, no info is shared %.
then & hence $I(X;Y) = D(p(x,y) \| p(x)p(y)) = D(p(x)p(y) \| p(x)p(y)) = 0$
as expected.
Note that the converse is also true: $I(X;Y) = 0 \Rightarrow X, Y$ indep.

The relationship among $H(X)$, $H(Y)$, $I(X;Y)$, $H(X|Y)$ & $H(Y|X)$ are summarized in the following Venn diagram.



Conditional Mutual Information + chain rules

Previously, we have shown that $H(XY) = H(X) + H(Y|X)$

What is $H(XY|Z)$ then? Start from scratch, we have

$$H(XY|Z) = -\sum_{xyz} p(xyz) \log p(xy|z) = -\sum_{xyz} p(xyz)\left[\log p(x|z) + \log p(y|xz)\right]$$

$$= H(X|Z) + H(Y|XZ)$$

Therefore, in general we have

$$H(X_1, X_2 \cdots X_N) = H(X_1) + H(X_2|X_1) + H(X_3|X_1, X_2) \cdots H(X_N|X_{N-1} \cdots X_1)$$

$$\left(\text{c.f.} \quad p(x_1 x_2 \cdots x_N) = p(x_1)\, p(x_2|x_1)\, p(x_3|x_2 x_1) \cdots p(x_N|x_{N-1} \cdots x_1) \right)$$

What is $I(X;Y|Z)$?

Define $I(X;Y|Z) \triangleq \sum_{xyz} p(xyz) \log \frac{p(xy|z)}{p(x|z)\,p(y|z)}$

$$= \sum_{xyz} p(xyz) \log \frac{p(x|yz)}{p(x|z)} = H(X|Z) - H(X|YZ)$$

Define a conditional version of relative entropy $D(p(x|z)\,||\,p(y|z))$ as

$$D(p(x|z)\,||\,p(y|z)) \triangleq \sum_{z} p(z) \sum_{yx} p(x|z) \log \frac{p(x|z)}{p(y|z)}$$

Basically, it is an average of "distance" between $p(x|z)$ & $p(y|z)$ over z

Then $I(X;Y|Z) = D(\,p(x,y|z)\,||\,p(x|z)\,p(y|z)\,)$

Note that the conditional version of relative entropy $D(p(x|z)\,||\,p(y|z))$ is still larger than 0. Since $\sum_{yz} p(x|z) \log \frac{p(x|z)}{p(y|z)}$ is larger than 0 for any z as shown previously

With $I(X;Y|Z)$ defined, we can obtain another chain rule for $I(X_1 \cdots X_N;Y)$

$$\begin{aligned} I(X_1 \cdots X_N ;Y) &= H(X_1 \cdots X_N) - H(X_1 \cdots X_N | Y) \\ &= H(X_1) + H(X_2|X_1) \cdots H(X_N|X_1 \cdots X_{N-1}) \\ &\quad - [H(X_1|Y) + H(X_2|X_1 Y) \cdots H(X_N|X_1 \cdots X_{N-1} Y)] \\ &= I(X_1;Y) + I(X_2;Y|X_1) + I(X_3;Y|X_1X_2) \cdots \\ &\quad I(X_N;Y|X_1X_2 \cdots X_{N-1}) \end{aligned}$$

Data Processing inequality

When $X \to Y \to Z$ forms a Markov chain, i.e. $p(x|y) = p(x|yz)$

Then $H(X|YZ) = \sum_{xyz} -p(xyz) \log p(x|yz) = \sum_{xyz} -p(x,y,z) \log p(x|y)$

$= H(X|Y)$

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

$+ \; I(X;Z) = H(X) - H(X|Z)$

$\therefore \; I(X;Y) - I(X;Z) = H(X|Z) - H(X|YZ) = I(X;Y|Z) \geq 0$

$\therefore$ we have $I(X;Y) \geq I(X;Z)$

In general if we preprocess a source (i.e. computing a function of Y, $T(Y)$), we always have Markov chain $X \to Y \to T(Y)$. Therefore $I(X;Y) \geq I(X;T(Y))$.
This is called data processing inequality; intuitively, information regarding the source (X) is lost due to processing.

When $I(X;Y) = I(X;T(Y))$, all information regarding X is still preserved in $T(Y)$. Then $T(Y)$ is a sufficient statistics in estimating X.
In this case, we also have $I(X;Y) \leq I(X;T(Y))$ & $X \to T(Y) \to Y$ forms a Markov Chain.

For example, let θ be the actual mean of a source, we sample the source 3 times to get $Y = (X_1 X_2 X_3)$

$T(Y) = X_1 + X_2 + X_3$ is the sufficient statistics for θ
i.e. $\theta \to T(Y) = X_1 + X_2 + X_3 \to Y = (X_1 X_2 X_3)$

However, let $T'(X) = X_1^2 + X_2^2 + X_3^2$. $T'(X)$ is NOT a sufficient statistics of θ.

Entropy rate

We will only consider iid source for most of this course. However, it is important to know how to quantify a random process in general.

Def : Entropy rate

For a random process, $X_1, X_2 \cdots X_n \cdots$, the entropy rate of the source is defined as

$$H(\mathcal{X}) \triangleq \lim_{n \to \infty} \frac{1}{n} H(X_1, X_2 \cdots X_n)$$

when the limit exists.

For iid source, $H(\mathcal{X})$ just equal to $H(X)$

We can also quantify a random source using conditional entropy

$$H'(\mathcal{X}) \triangleq \lim_{n \to \infty} H(X_n | X_{n-1} \cdots \cdots X_1)$$

For stationary case, $H'(\mathcal{X})$ exists since $H(X_n | X_{n-1} \cdots X_1)$ is a monotonic decreasing sequence bounded below by 0. Moreover, we have

Thm 4 : $H'(\mathcal{X}) = H(\mathcal{X})$

Before the proof, we will introduce a quick lemma without proof. You can refer to the book if you're interested in the proof.

Lemma 3 : Cesáro mean

If $a_n \to a$ & $b_n = \frac{1}{n} \sum_{i=1}^{n} a_i$, then $b_n \to a$

Pf: refer to Thm 4.23 of textbook.

Pf of Thm 4:

By chain rule, $\frac{H(X_1 \cdots X_n)}{n} = \frac{1}{n} \sum_{i=1}^{n} H(X_i | X_{i-1} \cdots X_1)$

Note that $H(X_i | X_1 \cdots X_{i-1}) \to H'(\mathcal{X})$ & hence by lemma 3,

$$\frac{H(X_1, \cdots X_n)}{n} \to H'(\mathcal{X}) \quad \text{but by def,} \quad \frac{H(X_1, \cdots X_n)}{n} \to H(\mathcal{X})$$

$$\therefore H(\mathcal{X}) = H'(\mathcal{X}) \quad \blacksquare$$