Non-singular Code: A code is nonsingular if every element of X maps into a different string of the alphabet set.

i.e. $x \neq x' \Rightarrow C(x) \neq C(x')$

Extension of a Code: The extension C^* of a code C is the mapping from finite strings of X to finite strings of the alphabet set $\mathcal{D}$ i.e.

$$C(\underbrace{x_1 x_2 \cdots x_n}_{\text{discrete memoryless source}}) = \underbrace{C(x_1)\,C(x_2)\cdots C(x_n)}_{\text{Concatenation of the codewords}}$$

Uniquely decodable Code: A code is uniquely decodable if its extension is non-singular. i.e. any encoded string in a uniquely decodable code has only one possible source string.

Note: ① prefix free code is uniquely decodable.

② Using the Shannon's idea the expected codeword length is $H(X) + 1$.

Q. Can we have/construct a uniquely decodable code with expected codeword length $H(X)$?

$$\begin{array}{cccc} a_1 & a_2 & \cdots & a_K \\ \downarrow & \downarrow & & \downarrow \\ p(a_1) & p(a_2) & \cdots & p(a_K) \end{array}$$

Q. Decide the codeword lengths $l_1, \cdots, l_K$ such that $\sum_{k=1}^{K} p_k l_k = H(X)$!!

Q. What should be the optimal average codeword length?

Huffman code (1952) (Optimal codeword length)

there is no other code whose average codeword length is smaller than the average codeword length of the Huffman code.

WLOG assume that $p(a_1) \geq p(a_2) \geq \cdots \geq p(a_K)$.

$\mathcal{D} = \{0, 1\}$ ie. binary code.

Notation: $\bar{a}_j = C(a_j)$

Goal: determine $l(\bar{a}_j)$ s.t. $\bar{l} = \sum_{k=1}^{K} p_k \, l(\bar{a}_k)$ is minimized.

Theorem For an optimal binary code the least likely codewords $\bar{a}_{K-1}$ and $\bar{a}_K$ have the same lengths ie. $l(\bar{a}_{K-1}) = l(\bar{a}_K)$ and they differ only at the last digit, e.g. if $\bar{a}_K$ ends with '1' and $\bar{a}_{K-1}$ ends with '0'.

Proof. Let $l_K = l(\bar{a}_K)$.

Observe: $l_K \geq l_j$ is true for optimal code, if not then there must exist a code in which $l_K < l_j$ for some j. Then we can interchange the codewords $\bar{a}_j$ with $\bar{a}_K$.

Then the effect of this procedure in the expected length is as follows:

$$\begin{aligned}
\Delta &= \left(p(a_j)\, l_K + p(a_K)\, l_j\right) - \left(p(a_j)\, l_j + p(a_K)\, l_K\right) \\
&= \left(p(a_j) - p(a_K)\right) l_K - \left(p(a_j) - p(a_K)\right) l_j \\
&= \left(p(a_j) - p(a_K)\right)\left(l_K - l_j\right) \\
&\leq 0 .
\end{aligned}$$

Therefore in an optimal code:

$$l_1 \leq l_2 \leq \cdots \leq l_{K-1} \leq l_K .$$

Now: we claim that there must be another codeword which differ from $\bar{a}_K$ only in the last digit. Otherwise for prefixfree code, ~~we can discard~~ the last bit of $\bar{a}_K$ can be dropped.

If $\bar{a}_j$ is the codeword which differs from $\bar{a}_K$ in only one position then we should have

$$l_j \geq l_{K-1} .$$

and hence again the codewords $\bar{a}_j$ and $\bar{a}_{K-1}$ can be interchanged without increasing the average codeword length.

Conclusion: The original problem of construction of the optimal code into a 'reduced' problem: we need codewords $\bar{a}_1, \cdots, \bar{a}_{K-2}$ and the first $l_K - 1$ digits of $\bar{a}_{K-1}$ or $\bar{a}_K$.

The set up now: $a'_1, a'_2, \cdots, a'_{K-1}$

where $$p(a'_k) = \begin{cases} p(a_k) & \text{if } 1 \leq k \leq K-2 \\ p(a_{K-1}) + p(a_K), & k = K-1. \end{cases}$$

Obs. If we can construct an optimal code for this set up then we can have the optimal code for the original problem.

Thm. If the prefix code corresponding to the reduced problem is optimal then the prefix free code corresponding to the original problem is optimal.

Proof. $a_1, a_2, \ldots a_K$ with $p_1 \geq p_2 \geq \ldots \geq p_K$

$\rightarrow a_1', a_2', \ldots, a_{K-1}'$ with $p_1' \geq p_2' \geq \ldots \geq p_{K-1}'$

First let us find out l_k:

$$l_k = \begin{cases} l_k' & \text{if } k \leq K-2 \\ l_k' + 1 & \text{if } k = K-1 \end{cases}$$

Then $$\bar{l} = \sum_{k=1}^{K} p(a_k) l_k$$

$$= \sum_{k=1}^{K-2} p(a_k) l_k' + \left(p(a_{K-1}) + p(a_K)\right)(l_{K-1}' + 1)$$

$$= \sum_{k=1}^{K-2} p(a_k') l_k' + p(a_{K-1}')\,(l_{K-1}' + 1)$$

$$= \bar{l}' + p(a_{K-1}').$$

Now apply this procedure iteratively.

So finally the problem reduces to the case $K = 2$.

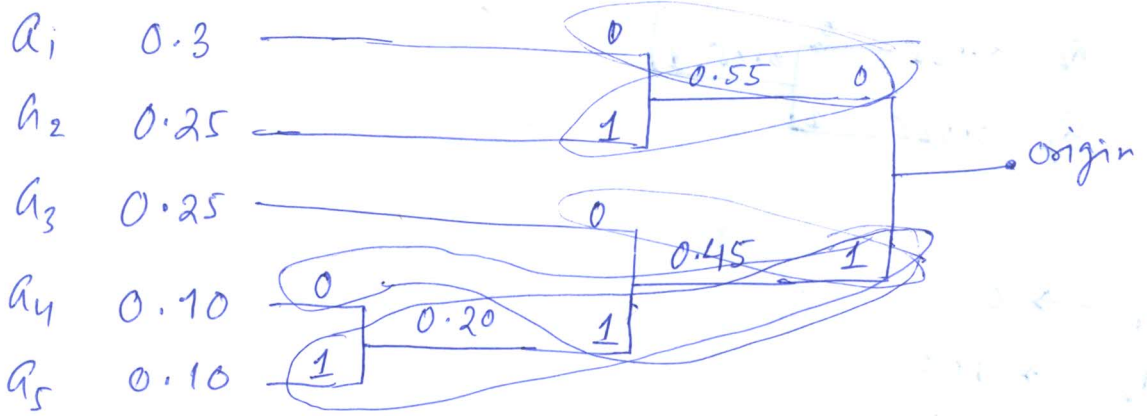
i.e. for two symbols.

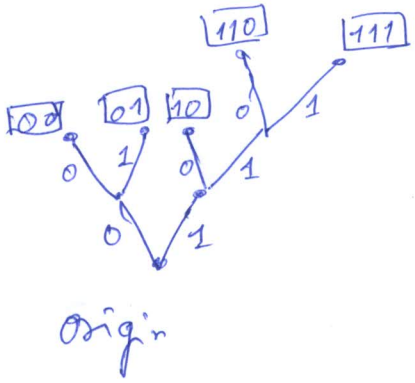
Exp 1.

$p(a_1) = 0.3$, $p(a_2) = 0.25$, $p(a_3) = 0.25$

$p(a_4) = 0.10$, $p(a_5) = 0.10$.

Note: $p(a_1) \geq p(a_2) \geq \cdots \geq p(a_5)$.





→ prefix code.

Algorithm for Huffman code.

(1) order source letters $a_1, \cdots, a_K$ such that $p(a_1) \geq \cdots \geq p(a_K)$

(2) assign code letter 0 to the letter a_{K-1} and 1 to a_K

(3) Construct a reduced source alphabet ~~$\{a_1, a_2, \cdots, a_{K-2},$~~
$\{a_1, a_2, \cdots, a_{K-2}, (a_{K-1}, a_K)\}$
with probabilities $p(a_1), p(a_2), \cdots, p(a_{K-2}), p(a_{K-1}) + p(a_K)$.

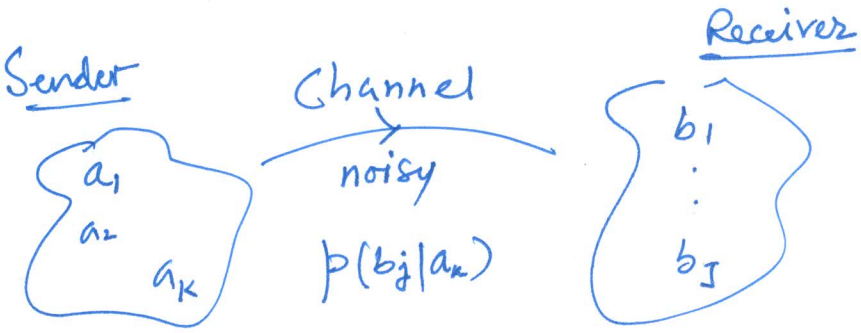
(4) repeat the steps. (1) & (2)

Drawback of Huffman Code

① Just by knowing the probabilities does not give us any idea for the codeword lengths.

~~This~~ i.e. there is no 'nice' functional representation for guessing the codeword length probabilities

— this could be a problem when K is large.



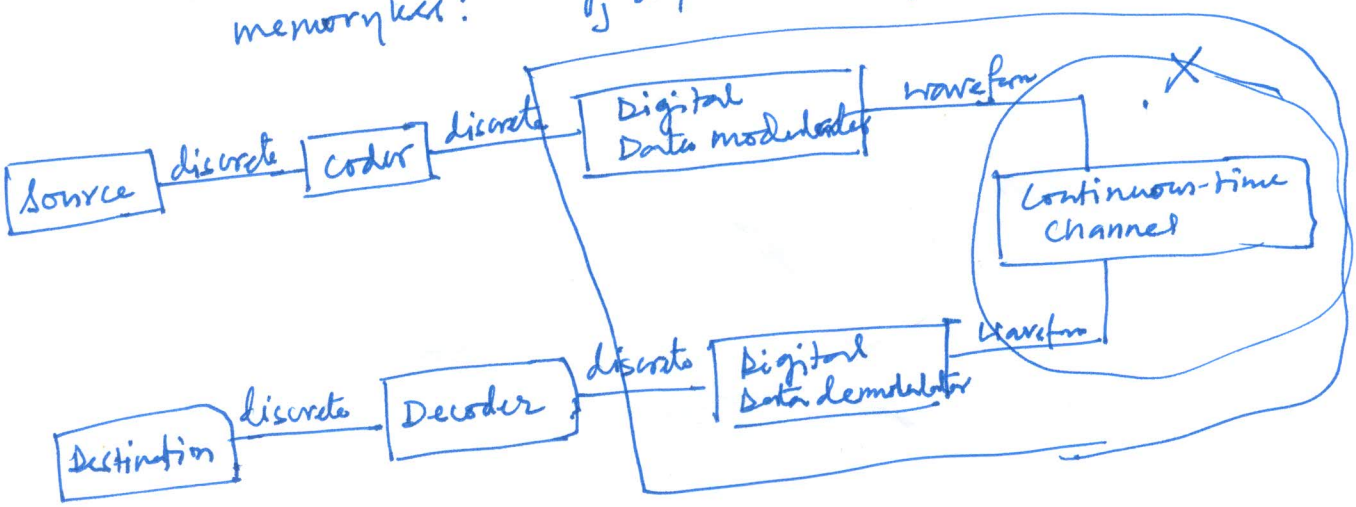
reliable communication through unreliable channel

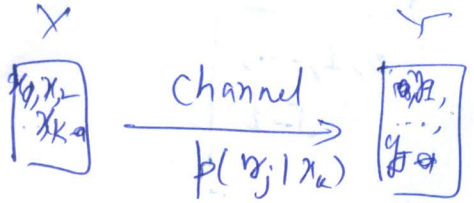
Discrete memoryless channel

$(x_1, x_2, \ldots x_n) \longrightarrow (y_1, y_2, \ldots, y_n)$

input — output

memoryless: y_j depends only on x_j





$$\begin{bmatrix} p(x_1) \\ p(x_2) \\ \vdots \\ p(x_K) \end{bmatrix} \longmapsto \begin{bmatrix} p(y_1) \\ p(y_2) \\ \vdots \\ p(y_{J-1}) \end{bmatrix}$$

input → output

where

$$p(y_j) = \sum_{k=1}^{K} p(y_j|x_k)\, p(x_k)$$

$$\therefore \begin{bmatrix} p(y_1) \\ p(y_2) \\ \vdots \\ p(y_J) \end{bmatrix} = \underbrace{\begin{bmatrix} p(y_1|x_1) & p(y_1|x_2) & \cdots & p(y_1|x_K) \\ \vdots & \vdots & & \vdots \\ p(y_J|x_1) & p(y_J|x_2) & \cdots & p(y_J|x_K) \end{bmatrix}}_{J \times K} \begin{bmatrix} p(x_1) \\ p(x_2) \\ \vdots \\ p(x_K) \end{bmatrix}$$

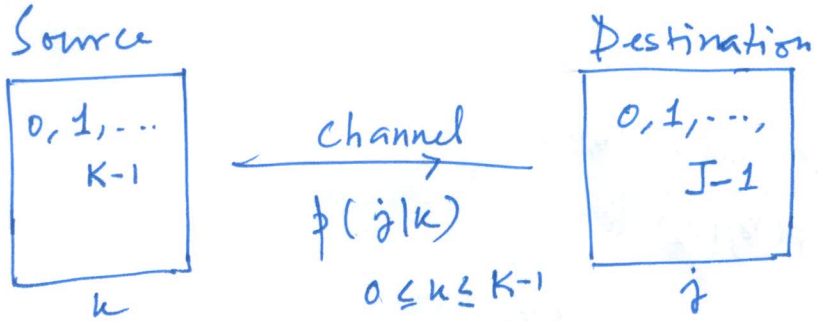
channel matrix

Q. What are the properties of the matrix

Obs. (1) non-negative entries.

(2) Sum of the entries of each column is 1

(column) stochastic matrix



$\bar{x} \equiv (x_1, x_2, \cdots, x_N)$, $x_n \in \{0, 1, \ldots, K-1\}$, $1 \le n \le N$

$\bar{y} \equiv (y_1, y_2, \cdots, y_N)$, $y_n \in \{0, 1, \ldots, J-1\}$, $1 \le n \le N$

$p(y_n | x_n) \longrightarrow$ transition probability

Defn. A channel is called memoryless if each output letter in the output sequence depends only on the corresponding input i.e.

$$p_N(\bar{y} | \bar{x}) = p_N\left((y_1, \ldots, y_N) | (x_1, \ldots, x_N)\right)$$

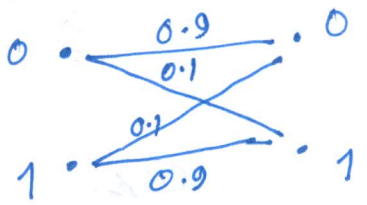
$$= \prod_{n=1}^{N} p(y_n | x_n)$$

probability of receiving $\bar{y}$ when $\bar{x}$ is sent

for all n, N, $\bar{x}$, $\bar{y}$.

Exp.

$p(0|0) = 0.9$
$p(1|0) = 0.1$
$p(0|1) = 0.1$
$p(1|1) = 0.9$



BSC

Q. What do we mean by saying memoryless BSC?

$$p(\underbrace{00}_{\bar{y}} | \underbrace{00}_{\bar{x}}) = p(0|0)\, p(0|0) = (0.9)^2$$

$$p(10|00) = p(1|0)\, p(0|0) = (0.1) \times (0.9).$$

Recall if

$$I(x;y) = \log \frac{p(x|y)}{p(x)} = \log \frac{p(y|x)}{p(y)} = \log \frac{p(x,y)}{p(x)p(y)} = I(y;x)$$

and

$$I(X;Y) = \sum_{x,y} p(x,y)\, I(x;y)$$

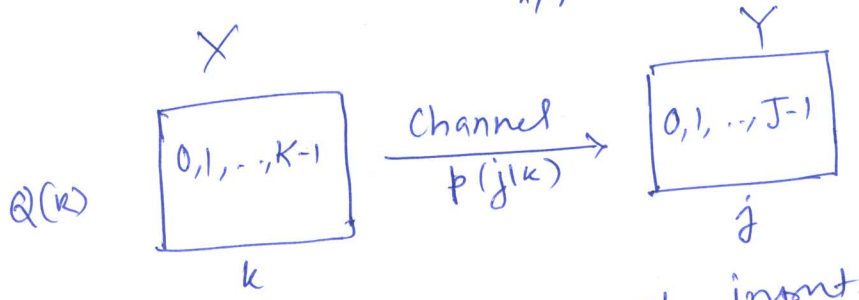
$$= \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)}$$

= 'average' mutual information $\rightarrow I(X;Y)$

Notation:

$$I(X;Y) = \sum_{x,y} p_{XY}(x,y) \log \frac{p_{X|Y}(x|y)}{p_X(x)}$$

$$= \sum_{x,y} p_X(x)\, p_{Y|X}(y|x) \log \frac{p_{X|Y}(x|y)}{p_X(x)}$$



$$p(j) = \sum_k Q(k)\, p(j|k)$$

$Q(k) \rightarrow$ denotes the pmf of inputs

$p(j|k) \rightarrow$ transition probability

Then

$$I(X;Y) = \sum_{k=0}^{K-1} \sum_{j=0}^{J-1} Q(k)\, p(j|k) \log \frac{p(k|j)}{Q(k)}$$

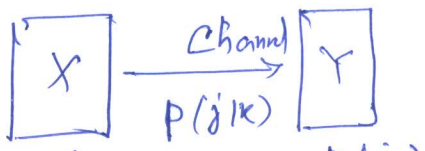
$$= \sum_k \sum_j Q(k)\, p(j|k) \log \frac{p(j|k)}{p(j)}$$

$$= \sum_k \sum_j Q(k)\, p(j|k) \log \frac{p(j|k)}{\sum_{i=0}^{K-1} Q(i)\, p(j|i)}$$

Obs. Property of a channel wrt different input pmfs!

Capacity of a DMC

The largest 'average' mutual information that can be obtained over the channel



$$C = \max_{Q(0), Q(1), \ldots, Q(K-1)} I(X;Y)$$

$$= \max_{Q} \sum_{k,j} Q(k) P(j|k) \log \frac{P(j|k)}{\sum_{i=0}^{K-1} P(i) P(j|i)}$$

Capacity.

i.e. $\max I(X;Y)$ wrt $\sum_{k=0}^{K-1} q_k = 1$, $Q(k) = q_k \geq 0$

Q. Does this exist?

Note: 'reliable communication' – rate of transmission should be less than the capacity of the channel.

Obs. $\bar{x} = (x_1, \ldots, x_N)$ → input, $x_i \in \{0, 1, \ldots, K-1\}$

$\bar{y} = (y_1, \ldots, y_N)$ → output, $y_j \in \{0, 1, \ldots, J-1\}$.

Thm. Let $Q_N(\bar{x})$ be the pmf on sequences of N input letters in a DMC.

Let $\bar{X}^N$, $\bar{Y}^N$ denote the random variables corresponding to the sequence of N-length input & output sequences respectively. $\bar{X}^N = (X_1, X_2, \ldots X_N)$, X_i are iid

$\bar{Y}^N = (Y_1, Y_2, \ldots Y_N)$, Y_i are iid

Then $I(\bar{X}^N; \bar{Y}^N) \leq \sum_{n=1}^{N} I(X_n; Y_n)$.

and $I(\bar{X}^N; \bar{Y}^N) \leq NC$.

Proof.

$$I(\bar{X}^N; \bar{Y}^N) = H(\bar{Y}^N) - H(\bar{Y}^N | \bar{X}^N)$$

$$= H(\bar{Y}^N) - \sum_{i=1}^{N} H(Y_i | Y_1, \ldots Y_{i-1}, \bar{X}^N)$$

$$= H(\bar{Y}^N) - \sum_{i=1}^{N} H(Y_i | X_i), \quad \text{since the channel is memoryless}$$

$$\leq \sum_{i=1}^{N} H(Y_i) - \sum_{i=1}^{N} H(Y_i | X_i)$$

$$= \sum_{i=1}^{N} \left(H(Y_i) - H(Y_i | X_i) \right)$$

$$= \sum_{i=1}^{N} I(X_i; Y_i).$$

Now recall that channel capacity is defined for individual rvs.

$$I(X_i; Y_i) \leq C. \quad \forall i.$$

$$\therefore I(\bar{X}^N; \bar{Y}^N) \leq NC.$$

Obs. $\bar{X}^N = (X_1, X_2, \ldots, X_N)$

$\downarrow$

$x_1 \quad x_2 \quad \ldots \quad x_N$

$\uparrow$

$\{0, \ldots, K-1\} \leftarrow p(0), p(1), \ldots p(K-1)$.