3F1 Information Theory, Lecture 4

Jossy Sayir

UNIVERSITY OF CAMBRIDGE
Department of Engineering

Michaelmas 2011, 30 November 2011

Summary of last lecture

- Block Coding of Memoryless Sources
- Arithmetic Coding
- Sources with Memory

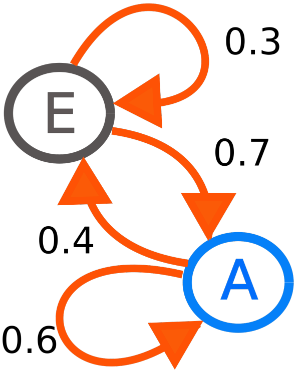
Shannon's converse Source Coding Theorem for a Discrete Stationary Source

$$\frac{E[W]}{N} \geq \frac{H_\infty(X)}{\log D}$$

where $H_\infty(X) = \lim_{N\to\infty} H(X_N|X_1 \ldots X_{N-1}) = \lim_{N\to\infty} \frac{1}{N} H(X_1 \ldots X_N)$

- Shannon's twin experiment

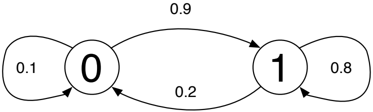
Markov Chain



Andrey Andreyevivich Markov

- Stationary state random process $S_1, S_2, \ldots$
- $P(s_N | s_1 \ldots s_{N-1}) = P(s_N | s_{N-1})$
- Markov information source: states S_i are mapped into source symbols X_i
- Unifilar information source: from any state, all neighbouring states map to distinct symbols

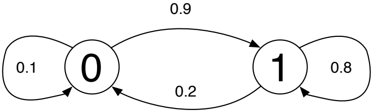
Unifilar Markov Source



- $P_{X_2|X_1}(1|0) = 1 - P_{X_2|X_1}(0|0) = 0.9$
- $P_{X_2|X_1}(1|1) = 1 - P_{X_2|X_1}(0|1) = 0.8$
- Can we compute $P_{X_1}(1) = 1 - P_{X_1}(0)$?
- Stationarity implies $P_{X_1}(1) = P_{X_2}(1)$ and thus

$$\begin{aligned} P_{X_1}(1) = P_{X_2}(1) &= P_{X_1X_2}(01) + P_{X_1X_2}(11) \\ &= P_{X_2|X_1}(1|0)P_{X_1}(0) + P_{X_2|X_1}(1|1)P_{X_1}(1) \end{aligned}$$

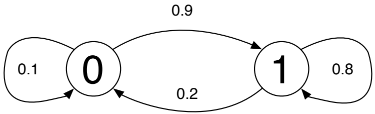
Unifilar Markov Source



- $P_{X_2|X_1}(1|0) = 1 - P_{X_2|X_1}(0|0) = 0.9$
- $P_{X_2|X_1}(1|1) = 1 - P_{X_2|X_1}(0|1) = 0.8$
- Can we compute $P_{X_1}(1) = 1 - P_{X_1}(0)$?
- Stationarity implies $P_{X_1}(1) = P_{X_2}(1)$ and thus

$$\begin{aligned} P_{X_1}(1) = P_{X_2}(1) &= P_{X_1X_2}(01) + P_{X_1X_2}(11) \\ &= P_{X_2|X_1}(1|0)P_{X_1}(0) + P_{X_2|X_1}(1|1)P_{X_1}(1) \end{aligned}$$

Unifilar Markov Source



- Define the matrix

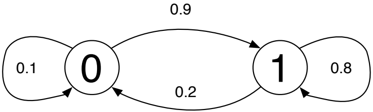
$$T = \begin{bmatrix} P_{X_2|X_1}(0|0) & P_{X_2|X_1}(0|1) \\ P_{X_2|X_1}(1|0) & P_{X_2|X_1}(1|1) \end{bmatrix}$$

and the vector $P = [P_{X_1}(0), P_{X_1}(1)]^T$, then we are looking for the solution P to the equation

$$P = TP,$$

i.e., the eigenvector of T for the eigenvalue 1. Note that since T is a stochastic matrix (its columns sum to 1), it will always have 1 as an eigenvalue.

Unifilar Markov Source



- $P = \begin{bmatrix} 0.1 & 0.2 \\ 0.9 & 0.8 \end{bmatrix} P$ implies

$$\begin{bmatrix} -0.9 & 0.2 \\ 0.9 & -0.2 \end{bmatrix} P = 0$$

which, together with the constraint $[11]P = 1$ (probabilities sum to 1) yields

$$\begin{bmatrix} -0.9 & 0.2 \\ 1 & 1 \end{bmatrix} P = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

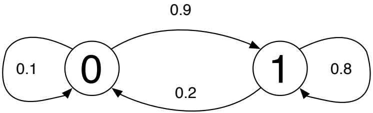
and finally

$$P = \begin{bmatrix} P_{X_1}(0) \\ P_{X_1}(1) \end{bmatrix} = \begin{bmatrix} 0.1818 \\ 0.8182 \end{bmatrix}$$

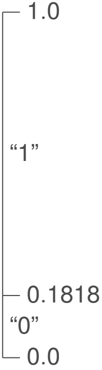
- Entropy rate of the source:

$$\begin{aligned} H_\infty(X) &= \lim_{N\to\infty} H(X_N|X_1 \dots X_{N-1}) = H(X_N|X_{N-1}) = H(X_2|X_1) \\ &= H(X_2|X_1 = 0)P_{X_1}(0) + H(X_2|X_1 = 1)P_{X_1}(1) \\ &= 0.1818h(0.1) + 0.8182h(0.2) = 0.6759 \text{ bits} \end{aligned}$$

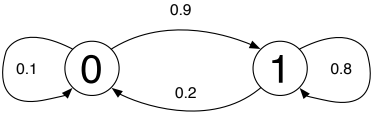
Encoding a unifilar Markov Source



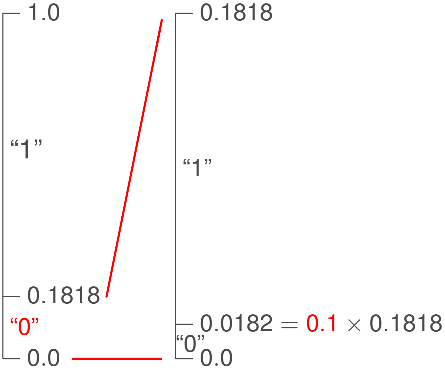
- Encode source output sequence: 0,1,1,1,1,1,1,1



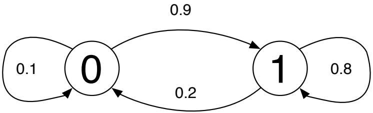
Encoding a unifilar Markov Source



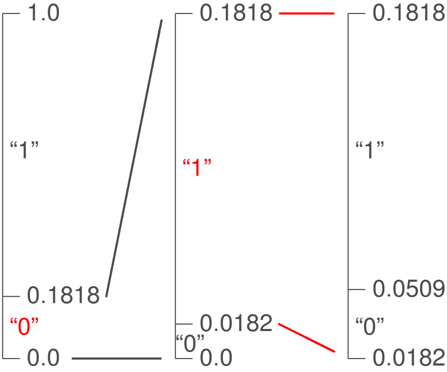
- Encode source output sequence: 0,1,1,1,1,1,1,1



Encoding a unifilar Markov Source

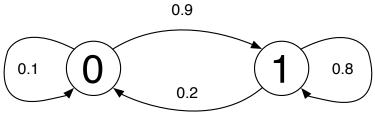


- Encode source output sequence: 0,1,1,1,1,1,1,1

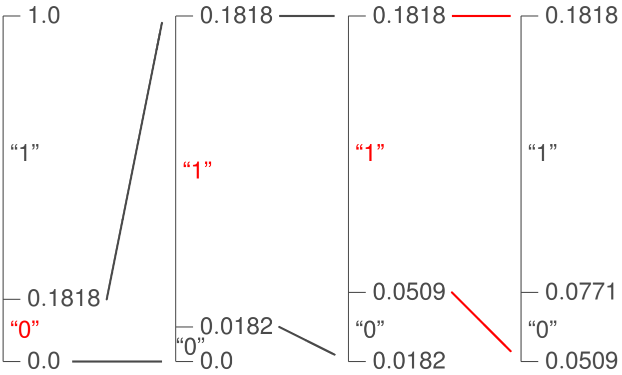


$0.0509 = 0.2 \times (0.1818 - 0.0182) + 0.0182$

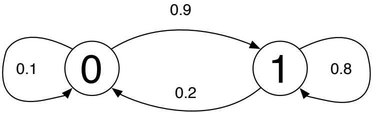
Encoding a unifilar Markov Source



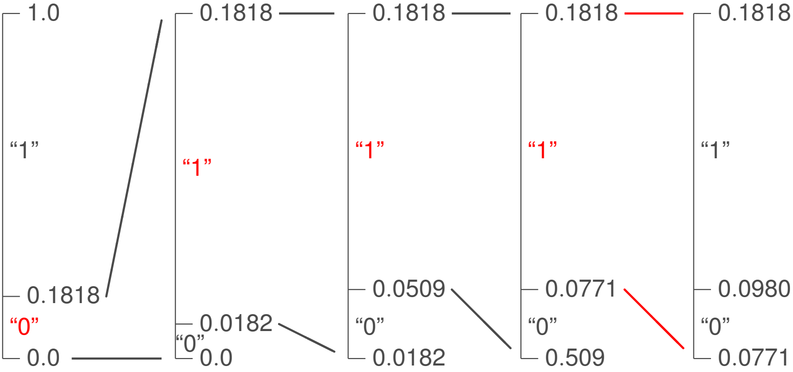
- Encode source output sequence: 0,1,1,1,1,1,1,1



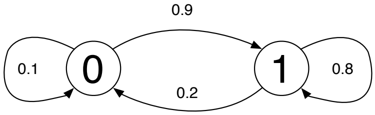
Encoding a unifilar Markov Source



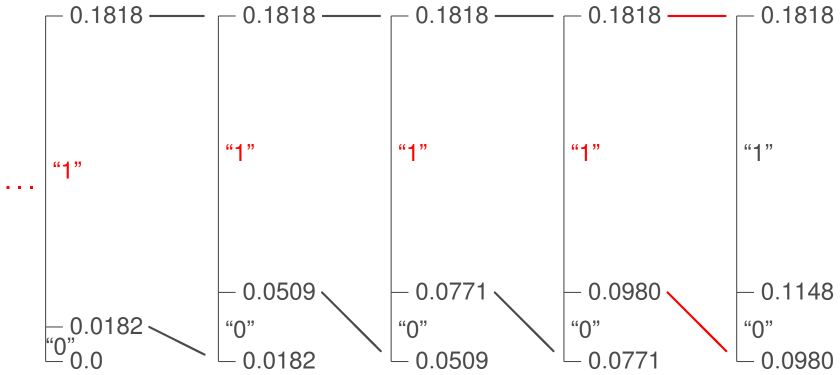
- Encode source output sequence: 0,1,1,1,1,1,1,1



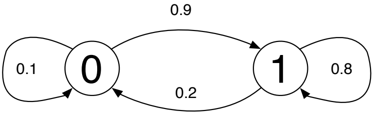
Encoding a unifilar Markov Source



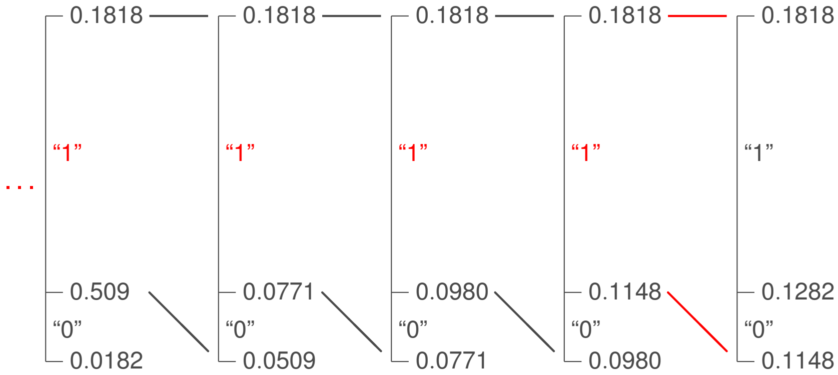
- Encode source output sequence: 0,1,1,1,1,1,1,1



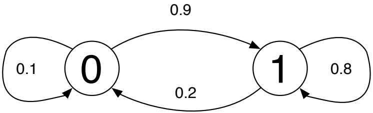
Encoding a unifilar Markov Source



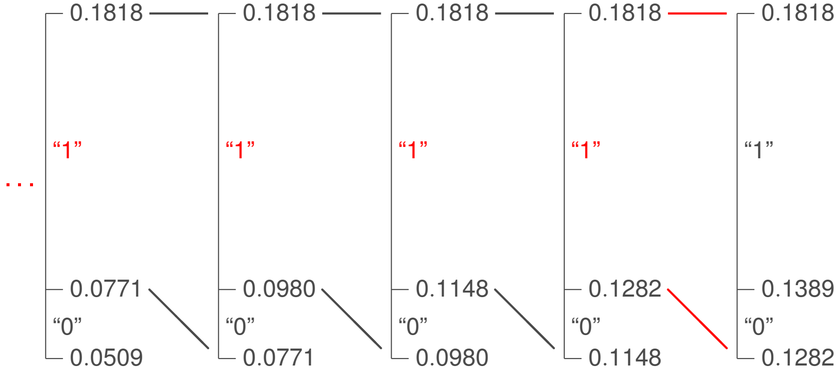
- Encode source output sequence: 0,1,1,1,1,1,1,1



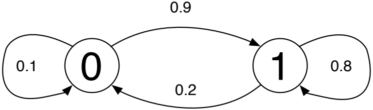
Encoding a unifilar Markov Source



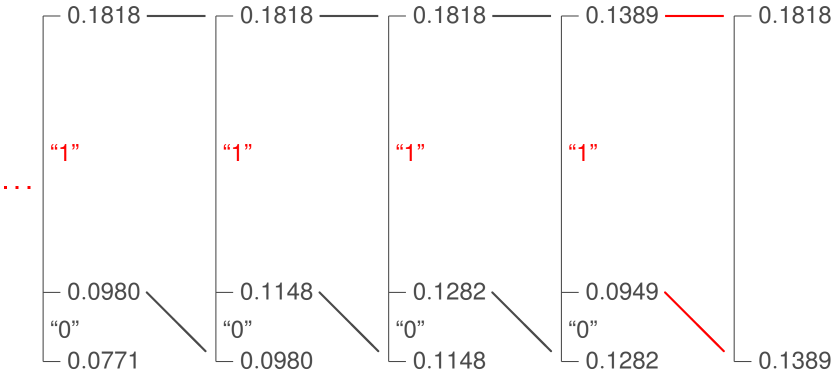
- Encode source output sequence: 0,1,1,1,1,1,1,1



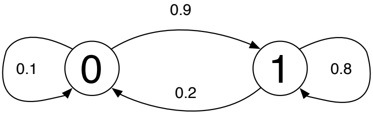
Encoding a unifilar Markov Source



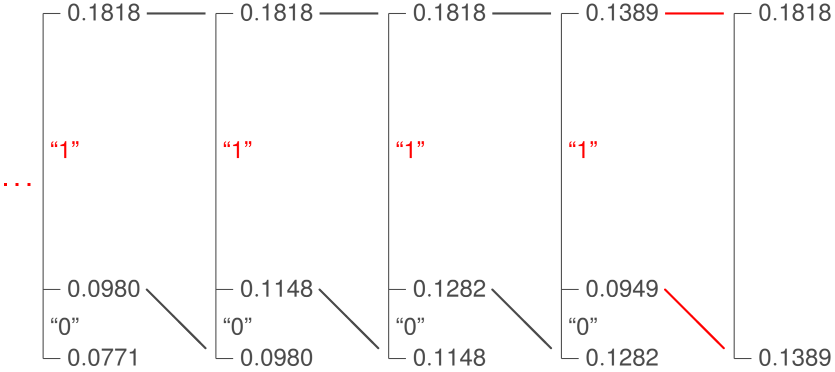
- Encode source output sequence: 0,1,1,1,1,1,1,1



Encoding a unifilar Markov Source



- Encode source output sequence: 0,1,1,1,1,1,1,1



Determining the codeword

- Source interval $[0.1389, 0.1818]$ in binary:

$$[0.00100011, 0.00101110]_b$$

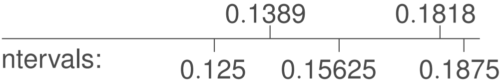
- The probability of the source sequence is

$$P_{X_1 \dots X_8}(0, 1, 1, 1, 1, 1, 1, 1) = 0.1818 - 0.1389 = 0.042896$$

- $-\log_2 P_{X_1 \dots X_8}(0, 1, 1, 1, 1, 1, 1, 1) = 4.543$, therefore we can either truncate after 5 or 6 digits, depending if the resulting code sequence is contained in the source interval
- No 5 digit code sequence corresponds to a code interval contained in our source interval:

Source interval:

Length 5 codeword intervals:

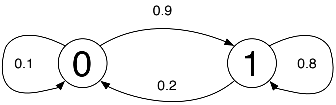


- The 6 digit code sequence 001010 corresponds to the code interval

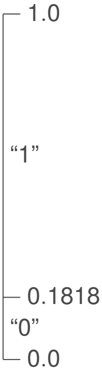
$$[0.001010, 0.001011]_b = [0.15625, 0.171875]$$

which is fully contained in the source interval and therefore satisfies the prefix condition

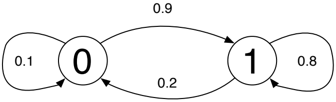
Decoding a unifilar Markov Source



- Decode code sequence: 0,0,1,0,1,0 corresponding to interval $[0.15625, 0.171875]$
- Decoding rule: always pick sub-interval that contains the codeword interval

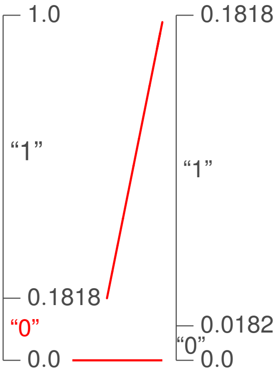


Decoding a unifilar Markov Source

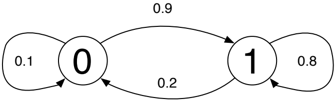


- Decode code sequence: 0,0,1,0,1,0 corresponding to interval $[0.15625, 0.171875]$

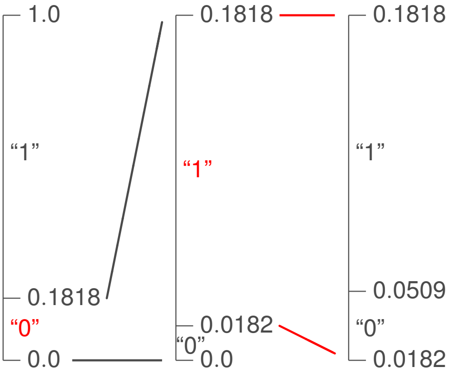
- Decoding rule: always pick sub-interval that contains the codeword interval, result: 0



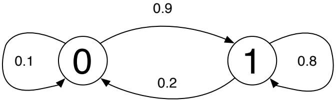
Decoding a unifilar Markov Source



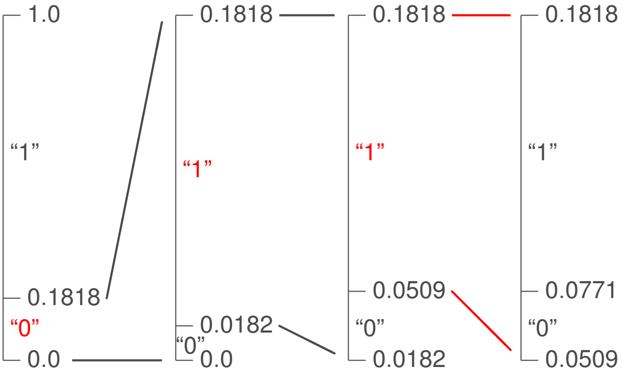
- Decode code sequence: 0,0,1,0,1,0 corresponding to interval $[0.15625, 0.171875]$
- Decoding rule: always pick sub-interval that contains the codeword interval, result: 0,1



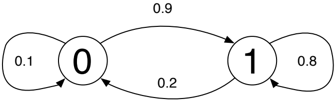
Decoding a unifilar Markov Source



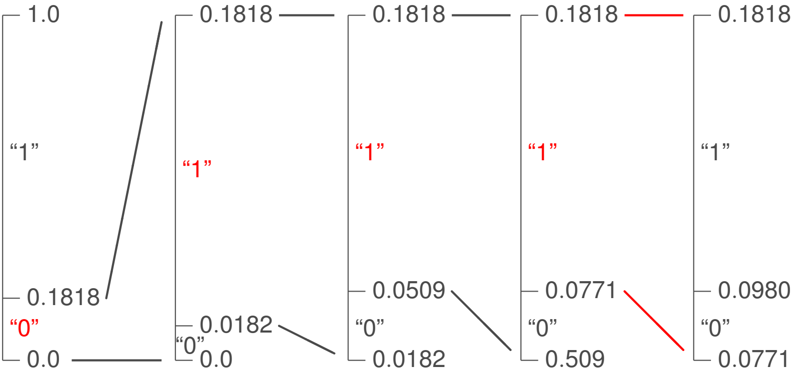
- Decode code sequence: 0,0,1,0,1,0 corresponding to interval $[0.15625, 0.171875]$
- Decoding rule: always pick sub-interval that contains the codeword interval, result: 0,1,1



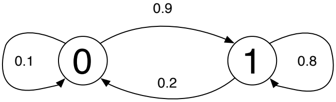
Decoding a unifilar Markov Source



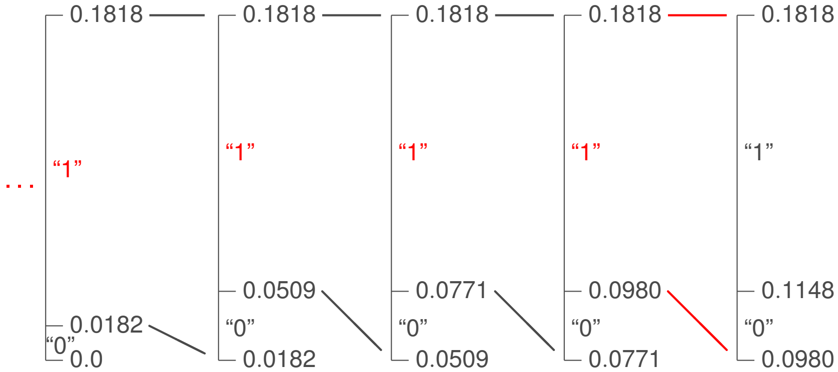
- Decode code sequence: 0,0,1,0,1,0 corresponding to interval $[0.15625, 0.171875]$
- Decoding rule: always pick sub-interval that contains the codeword interval, result: 0,1,1,1



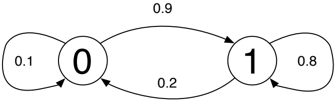
Decoding a unifilar Markov Source



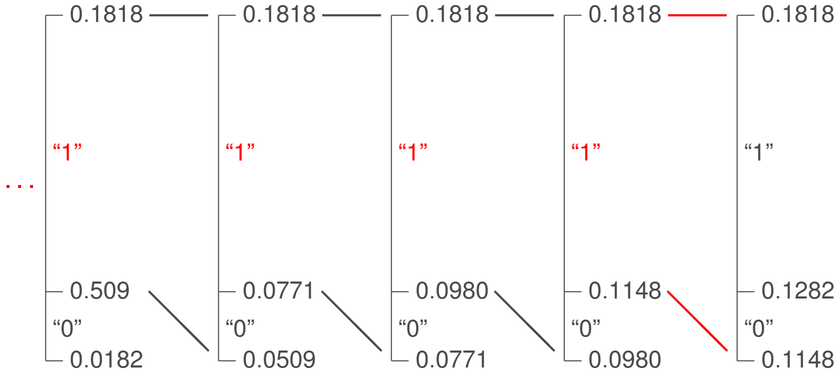
- Decode code sequence: 0,0,1,0,1,0 corresponding to interval $[0.15625, 0.171875]$
- Decoding rule: always pick sub-interval that contains the codeword interval, result: 0,1,1,1,1



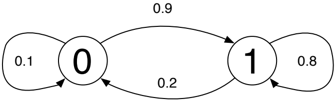
Decoding a unifilar Markov Source



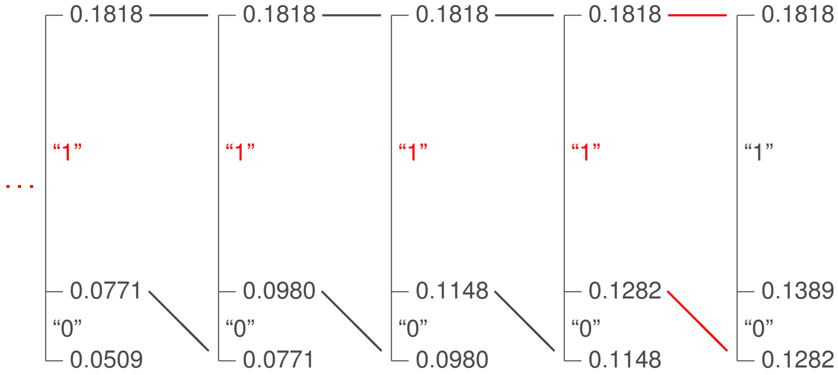
- Decode code sequence: 0,0,1,0,1,0 corresponding to interval $[0.15625, 0.171875]$
- Decoding rule: always pick sub-interval that contains the codeword interval, result: 0,1,1,1,1,1



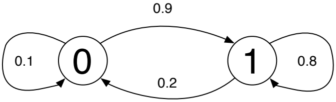
Decoding a unifilar Markov Source



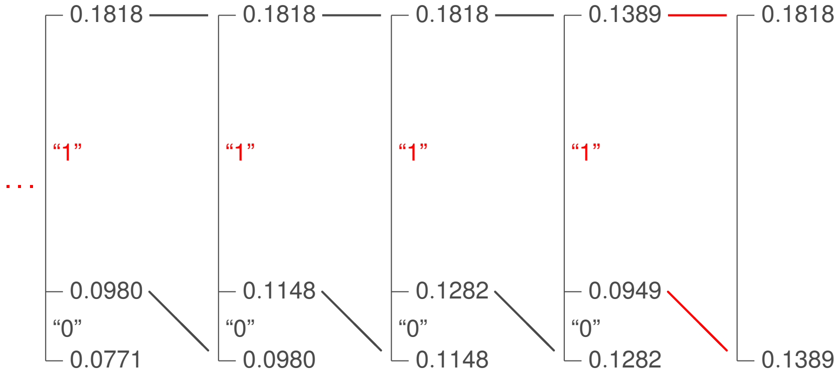
- Decode code sequence: 0,0,1,0,1,0 corresponding to interval $[0.15625, 0.171875]$
- Decoding rule: always pick sub-interval that contains the codeword interval, result: 0,1,1,1,1,1,1



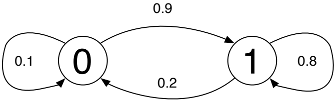
Decoding a unifilar Markov Source



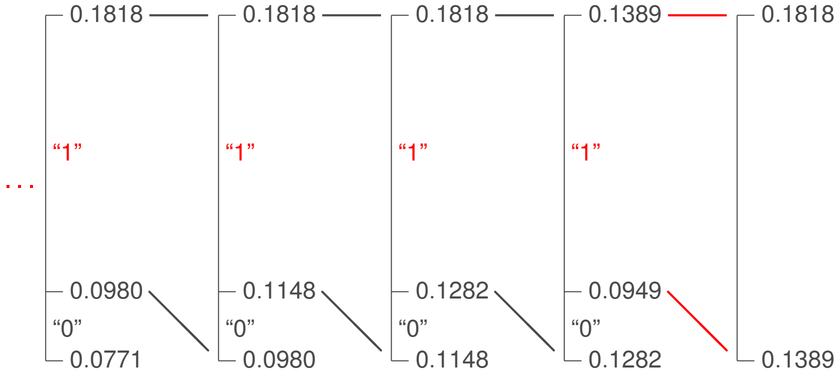
- Decode code sequence: 0,0,1,0,1,0 corresponding to interval $[0.15625, 0.171875]$
- Decoding rule: always pick sub-interval that contains the codeword interval, result: 0,1,1,1,1,1,1,1



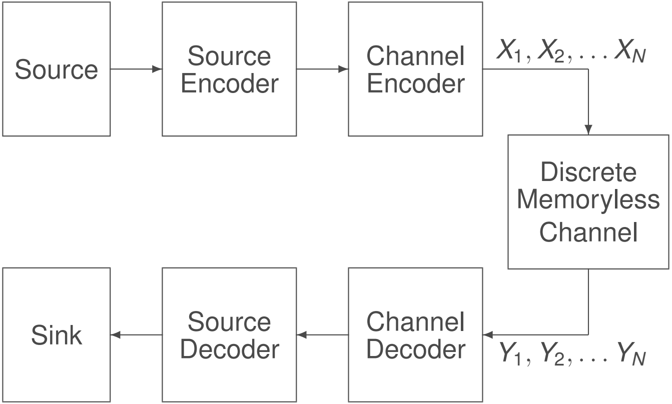
Decoding a unifilar Markov Source



- Decode code sequence: 0,0,1,0,1,0 corresponding to interval $[0.15625, 0.171875]$
- Decoding rule: always pick sub-interval that contains the codeword interval, result: 0,1,1,1,1,1,1,1



Channel Coding

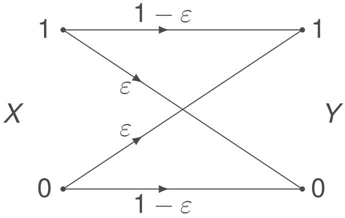


- Discrete Memoryless Channel (DMC):

$$P(y_1 \ldots y_N | x_1 \ldots x_N) = \prod_i P(y_i | x_i)$$

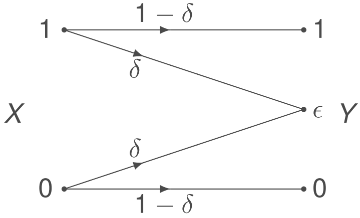
Two common DMCs

Binary Symmetric Channel (BSC)



$$P_{Y|X}(1|0) = 1 - P_{Y|X}(0|0)$$
$$= 1 - P_{Y|X}(1|1)$$
$$= P_{Y|X}(0|1) = \varepsilon$$

Binary Erasure Channel (BEC)



$$P_{Y|X}(1|1) = P_{Y|X}(0|0) = 1 - \delta$$
$$P_{Y|X}(1|0) = P_{Y|X}(1|0) = 0$$
$$P_{Y|X}(\epsilon|0) = P_{Y|X}(\epsilon|1) = \delta$$

Chain rule of entropies

Two random variables

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

Follows directly from our definition of $H(Y|X)$

Any number of random variables

$$H(X_1 X_2 \ldots X_N) = H(X_1) + H(X_2|X_1) + \ldots + H(X_N|X_1 \ldots X_N)$$

Follows from recursive application of the two variable chain rule

Mutual Information

Definition

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

Mutual information is mutual:

$$I(X;Y) = H(X) + H(Y) - H(XY) = H(Y) - H(Y|X)$$

Positivity of Mutual Information

Theorem

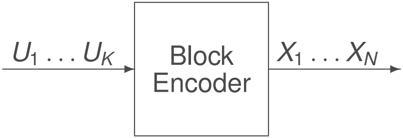
$$I(X;Y) \geq 0$$

with equality if and only if X and Y are independent

Equivalent to $H(X|Y) \leq H(X)$,i.e., conditioning on a random variable can only reduce uncertainty. This was stated without proof in the previous lecture, so we prove it here:

$$\begin{aligned}
-I(X;Y) &= H(XY) - H(X) - H(Y) \\
&= \sum_{x,y} P(x,y) \log \frac{P(x)P(y)}{P(x,y)} \\
&\leq \sum_{x,y} P(x,y) \left[\frac{P(x)P(y)}{P(x,y)} - 1 \right] \text{ (IT-inequality)} \\
&= \sum_{x,y} P(x)P(y) - \sum_{x,y} P(x,y) = 0
\end{aligned}$$

Block coding and coding rate



- Block coding rate: $R_B \stackrel{\text{def}}{=} K/N$
- Channel information rate (independently of the coding method used):

$$R \stackrel{\text{def}}{=} \frac{H(X_1 \dots X_N)}{N}$$

- If the block code is applied to a uniformly distributed source and all codewords are distinct, the two rates coincide

Channel Capacity

Definition

$$C = \max_{P_X} I(X;Y)$$

Weak Converse Coding Theorem

$$\begin{aligned} H(X_1 \dots X_N | Y_1 \dots Y_N) =& H(X_1 \dots X_N Y_1 \dots Y_N) - H(Y_1 \dots Y_N) \\ =& H(X_1 \dots X_N) + H(Y_1 \dots Y_N | X_1 \dots X_N) \\ & - H(Y_1 \dots Y_N) \\ =& NR + NH(Y_1|X_1) - NH(Y_1) + NH(Y_1) \\ & - H(Y_1 \dots Y_N) \\ \leq& NR - NI(X;Y) \text{ (since } H_N(Y) \text{ decreases with } N) \\ \leq& N(R-C) \text{ (since } I(X;Y) \leq C) \end{aligned}$$

Weak Converse

$$H(X_1 \dots X_N | Y_1 \dots Y_N) \geq N(R-C)$$

In other words, if $R > C$, there is necessarily a residual uncertainty about the input block after observing the output of the channel.

Note that we have implicitly assumed that $Y_1 \dots Y_N$ is stationary for the proof, which is not generally true, but a similar result can be shown for non stationary output blocks

Shannon's Coding Theorem

Converse

If information bits from a binary symmetric source are sent to their destination at rate R (in bits per use) via the DMC of capacity C (in bits per use) without feedback, then bit error probability P_b at the destination satisfies

$$P_b \geq h^{-1}(1 - C/R) \text{ , if } R > C.$$

Direct part

Consider transmitting information bits from a binary symmetric source to their destination at rate $R = K/N$ using block coding with blocklength N via a DMC of capacity C (in bits per use) used without feedback. Then, given any $\varepsilon > 0$, provided that $R < C$, one can always, by choosing N sufficiently large and designing appropriate encoders and decoders, achieve a block error probability

$$P_B < \varepsilon.$$

Capacity of two common channels

- Binary erasure channel:

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(X) - \delta H(X|Y = \epsilon) - (1-\delta)H(X|Y \neq \epsilon) \\ &= H(X) - \delta \end{aligned}$$

which is maximised when $P_X(0) = P_X(1) = 1/2$ for, so

$$C_{\text{BEC}} = h(1/2) - \delta = 1 - \delta \text{ bits per use}$$

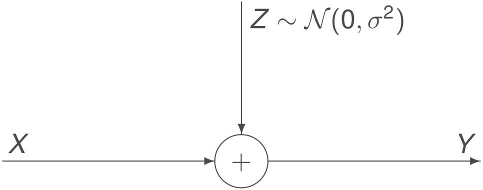
- Binary symmetric channel:

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H(Y) - H(Y|X=0)P_X(0) - H(Y|X=1)P_X(1) \\ &= H(Y) - h(\varepsilon(P_X(0) + P_X(1)) \end{aligned}$$

which again is maximises when $P_X(0) = P_X(1) = 1/2$ for which $P_Y(0) = P_Y(1) = 1/2$ and thus

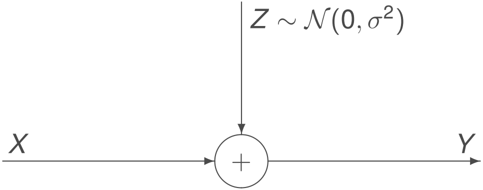
$$C_{\text{BSC}} = 1 - h(\varepsilon) \text{ bits per use}$$

An interesting continuous channel?



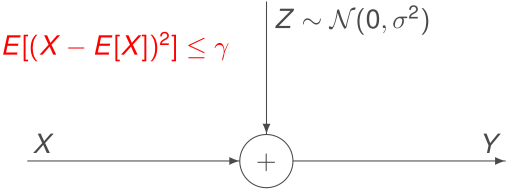
- X and Y continuous random variables
- Z is a continuous normal distributed random variable with mean 0 and variance σ^2
- Question: how much information can be transmitted over this channel?
- Answer: as much as desired! To transmit N bits, pick a density for X such that $E[X] = 0$ and $E[X^2] >> \sigma^2$ so that $Y \approx X$ to within N bits of accuracy with sufficiently high probability
- Conclusion: this is not an interesting communication problem

An interesting continuous channel?



- X and Y continuous random variables
- Z is a continuous normal distributed random variable with mean 0 and variance σ^2
- Question: how much information can be transmitted over this channel?
- Answer: as much as desired! To transmit N bits, pick a density for X such that $E[X] = 0$ and $E[X^2] >> \sigma^2$ so that $Y \approx X$ to within N bits of accuracy with sufficiently high probability
- Conclusion: this is not an interesting communication problem

Additive White Gaussian Noise (AWGN) channel



- Power constraint now makes it an interesting problem, unlike the problem on the previous page
- Power constraint often stated as $E[X^2] \leq \gamma, E[X] = 0$, which is essentially equivalent
- To understand this channel, we need an information theory of continuous variables

Information theory of continuous variables

- How much is our uncertainty/entropy about a continuous random variable?
- Infer from the discrete case: how many binary digits do we need on average to represent the outcome of a continuous random variable
- Example: the variable takes on the value $\pi = 3.141592\ldots$ How many binary (or decimal) digits do we need to represent π?
- Answer: infinitely many
- Conclusion: the (discrete) entropy of a continuous random variable in general is ∞

Information theory of continuous variables

- How much is our uncertainty/entropy about a continuous random variable?
- Infer from the discrete case: how many binary digits do we need on average to represent the outcome of a continuous random variable
- Example: the variable takes on the value $\pi = 3.141592\ldots$ How many binary (or decimal) digits do we need to represent π?
- Answer: infinitely many
- Conclusion: the (discrete) entropy of a continuous random variable in general is ∞

Differential (or relative) Entropy

Nonetheless, in analogy to discrete entropy, Shannon defined:

Definition

The differential entropy of a continuous random variable X with probability density function (pdf) $f_X(.)$ is

$$h(X) \stackrel{\text{def}}{=} -\int_{\text{supp } f_X} f_X(x) \log f_X(x) dx.$$

- retains most properties of discrete entropy (see next page)
- however: differential entropy can be negative and is not invariant under coordinate transformations. It is *relative* to a coordinate system (hence the appelation *relative entropy*.)

Properties of differential entropy and mutual information

- The differential entropy of joint distributions, conditional differential entropy or equivocation, and mutual information are defined in the same manner as for their discrete counterparts, and satisfy the same properties:

$$
\begin{aligned}
h(XY) &\leq h(X) + h(Y) \\
h(X|Y) &\leq h(X) \\
I(X;Y) &\stackrel{\text{def}}{=} h(X) - h(X|Y) \\
&= h(Y) - h(Y|X) \geq 0
\end{aligned}
$$

- For a given support of $f_X(.)$, $h(X)$ is maximised by the uniform density on supp f_X and equal to $\log V$, where V is the volume of supp f_X (or length of the support interval for scalar X).

Differential entropy and quantisation

Let us quantise supp f_X into regular bins of size Δ. By the mean value theorem, there exists a value x_i in each bin such that

$$f_X(x_i)\Delta = \int_{i\Delta}^{(i+1)\Delta} f_X(x)dx.$$

Let us define a discrete random variable Y that takes on the values x_i with probabilities $P_Y(x_i) = f_X(x_i)\Delta$. Then

$$\begin{aligned} H(Y) &= -\sum_i f_X(x_i)\Delta \log(f_X(x_i)\Delta) \\ &= -\sum_i \Delta f_X(x_i) \log f_X(x_i) - \log \Delta. \end{aligned}$$

By the definition of the Riemann integral,

$$\lim_{\Delta \to 0} \left[-\sum_i f_X(x_i) \log f_X(x_i)\Delta \right] = -\int f_X(x_i) \log f_X(x_i) dx = h(X).$$

Thus, for small Δ, $H(Y) \approx h(X) - \log \Delta$.

Differential entropy and quantisation

If Y is an n bit quantisation of X, then $\Delta = 2^{-n}$ and $H(Y) \approx h(X) + n$. Thus,

Source coding of continuous variables

$h(X) + n$ provides a lower bound for the average codeword length of a prefix-free code to reproduce X with n bit precision, which can be approached using Huffman or Shannon-Fano coding.

Examples:

- f_X uniform over $[0, 1]$, $h(X) = -\int_0^1 1 \log 1 = 0$. A block code of length n can reproduce X with n bit accuracy.
- f_X uniform over $[0, 1/2]$, $h(X) = -\int_0^{1/2} 2 \log 2 = -1$. A block code of length $n - 1$ can reproduce X with n bit accuracy, since the first digit of X is necessarily 0 and does not need to be encoded.

Normal Distribution

Differential entropy

For X Gaussian/Normal distributed, $f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}}$,

$$\begin{aligned} h(X) &= -\int f_X(x) \log f_X(x) dx \\ &= \int f_X(x) \log \sqrt{2\pi\sigma^2} + \frac{1}{2\sigma^2} \int f_X(x) x^2 dx \\ &= \frac{1}{2} \log(2\pi\sigma^2) + \frac{\sigma^2}{2\sigma^2} \\ &= \frac{1}{2} \log(2\pi e \sigma^2) \end{aligned}$$

where we used natural logarithms in the derivation, but the final result can revert to any desired base.

- If $\sigma = 1$, $h(X) = 2.0471$ bits, thus $2.0471 + n$ binary digits suffice on average to reproduce an $\mathcal{N}(0,1)$ r.v. with n bit accuracy.

Normal Distribution

Let X be normal distributed with mean 0 and variance σ^2 and Y have any distribution with the same mean and variance. Note that

$$-\int f_Y(z) \log f_X(z) dz = -\int f_X(z) \log f_X(z) dz \qquad (1)$$

as can be verified by repeating the derivation on the previous page replacing the f_X by f_Y and remembering that $\int y^2 f_Y(y) dy = \sigma^2$.

$$\begin{aligned} h(Y) - h(X) &= -\int f_Y(z) \log f_Y(z) dz + \int f_X(z) \log f_X(z) dz \\ &= \int f_Y(z) \log \frac{f_X(z)}{f_Y(z)} dz \qquad \text{(using (1))} \\ &\leq \int f_Y(z) \left(\frac{f_X(z)}{f_Y(z)} - 1 \right) dz = 0 \qquad \text{(IT-inequality)} \end{aligned}$$

Maximum Entropy

The normal distribution maximises the differential entropy among all distributions with a given variance σ^2.

Capacity of the AWGN Channel

Continuous Capacity

$$C \stackrel{\text{def}}{=} \max_{f_X \in \mathcal{P}} I(X;Y) = \max_{f_X \in \mathcal{P}} (h(Y) - h(Y|X))$$

where $\mathcal{P}$ is the set of permissible input distributions, e.g., for the AWGN channel the set of input distributions satisfying the power constraint $E[X^2] \leq \gamma$. A coding theorem can be proved for continuous channels analogous to the one we stated for discrete channels and the capacity remains the supremum of rates achievable with arbitrary reliability.

Capacity of the AWGN Channel

For the AWGN channel, $h(Y|X) = h(Z) = \frac{1}{2}\log(2\pi e\sigma^2)$ is independent of the choice of f_X. Therefore, maximising $I(X;Y)$ is equivalent to maximising $h(Y)$. Since X and Z are independent and zero mean, Y has zero mean and variance $E[Y^2] = E[X^2] + \sigma^2$. $h(Y)$ is maximised when Y has a normal distirbution, which is the case when X is normal. Let us denote $\sigma_X^2 \stackrel{\text{def}}{=} E[X^2]$, then

Capacity of the AWGN channel

$$C_{\text{AWGN}} = \frac{1}{2}\log(2\pi e(\sigma_x^2 + \sigma^2)) - \frac{1}{2}\log(2\pi e\sigma^2)$$
$$= \frac{1}{2}\log\left(1 + \frac{\sigma_X^2}{\sigma^2}\right) \quad \text{[bits/channel use]}$$

where σ_X^2/σ^2 is called the signal-to-noise ratio.

Communication engineers prefer to express capacity in bits/second, obtained by multiplying the above by the symbol rate.