
Signals & Systems (3F1)






– Examples Paper 3F1/3 Information Theory –

January 20, 2016

Standard notation

\mathbb{R}	Set of real numbers
$\log(a)$	base 2 logarithm of $a \in \mathbb{R}$
$P_X(\cdot)$	Probability distribution of rv X
$\mathbb{E}[\cdot]$	Expectation operator

Symbol legend

	Important fact
	Computations needed
	Use data book
	Pay attention
	Clarification

Question 3. Show that for statistically independent random variables,

$$H(X_1, X_2, \dots, X_n) = \sum_{i=1}^n H(X_i).$$

By definition of joint entropy, we have

$$H(X_1, X_2, \dots, X_n) = - \sum_{x_1, \dots, x_n} P_{X_1 \dots X_n}(x_1, \dots, x_n) \log P_{X_1 \dots X_n}(x_1, \dots, x_n), \quad (1)$$

where $P_{X_1 \dots X_n}(X_1 = x_1, \dots, X_n = x_n)$ is the joint probability distribution of X_1, \dots, X_n evaluated at $X_1 = x_1, \dots, X_n = x_n$. Since by assumption X_1, \dots, X_n are independent, we have

$$P_{X_1 \dots X_n}(x_1, \dots, x_n) = P_{X_1}(x_1)P_{X_2}(x_2) \cdots P_{X_n}(x_n), \quad \forall x_1, \dots, x_n.$$

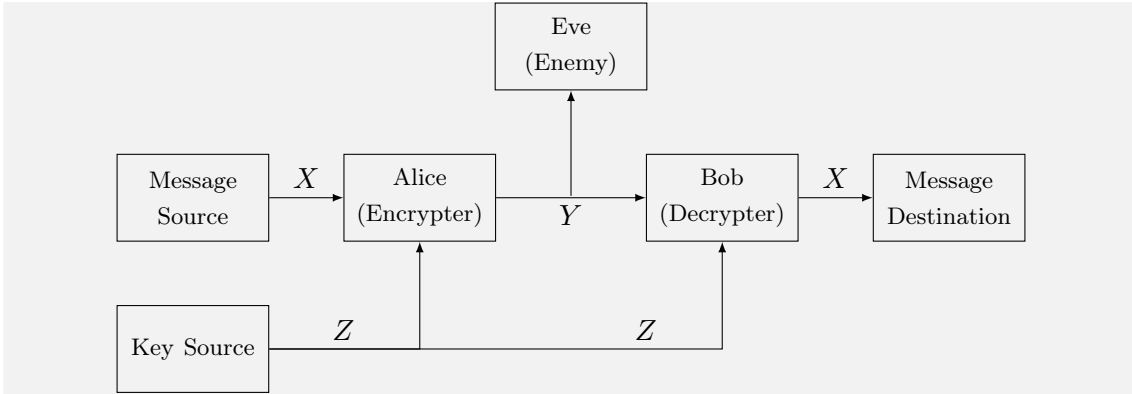
Consequently (1) can be rewritten as

$$\begin{aligned} H(X_1, X_2, \dots, X_n) &= - \sum_{x_1, \dots, x_n} P_{X_1}(x_1) \cdots P_{X_n}(x_n) \log P_{X_1}(x_1) \cdots P_{X_n}(x_n) \\ &= - \sum_{x_1, \dots, x_n} P_{X_1}(x_1)P_{X_2}(x_2) \cdots P_{X_n}(x_n) \log P_{X_1}(x_1) + \dots \\ &\quad - \sum_{x_1, \dots, x_n} P_{X_1}(x_1)P_{X_2}(x_2) \cdots P_{X_n}(x_n) \log P_{X_n}(x_n) \\ &= - \sum_{x_1} P_{X_1}(x_1) \log P_{X_1}(x_1) \left(\sum_{x_2, \dots, x_n} P_{X_2}(x_2) \cdots P_{X_n}(x_n) \right) + \dots \\ &\quad - \sum_{x_n} P_{X_n}(x_n) \log P_{X_n}(x_n) \left(\sum_{x_1, \dots, x_{n-1}} P_{X_1}(x_1) \cdots P_{X_{n-1}}(x_{n-1}) \right) \\ &= - \sum_{x_1} P_{X_1}(x_1) \log P_{X_1}(x_1) \left(\sum_{x_2} P_{X_2}(x_2) \cdots \sum_{x_n} P_{X_n}(x_n) \right) + \dots \\ &\quad - \sum_{x_n} P_{X_n}(x_n) \log P_{X_n}(x_n) \left(\sum_{x_1} P_{X_1}(x_1) \cdots \sum_{x_{n-1}} P_{X_{n-1}}(x_{n-1}) \right) \\ &= - \sum_{x_1} P_{X_1}(x_1) \log P_{X_1}(x_1) + \dots - \sum_{x_n} P_{X_n}(x_n) \log P_{X_n}(x_n) \\ &= H(X_1) + H(X_2) + \dots + H(X_n), \end{aligned}$$

and we are done. ◇

Question 4. While we cover in 3F1 and 4F5 the application of Shannon's theory to data compression and transmission, Shannon also applied the concepts of entropy and mutual information to the study of secrecy systems. The figure below shows a cryptographic scenario where Alice wants to transmit a secret plaintext message X to Bob and they share a secret key Z , while the enemy Eve has access to the public message Y .

Another way to solve the problem is to use the chain rule of entropies and recall the condition for equality in the conditioning theorem.



- (a) Write out two conditions using conditional entropies involving X , Y and Z to enforce the deterministic encryptability and decryptability of the messages.
- (b) Shannon made the notion of an “unbreakable cryptosystem” precise by saying that a cryptosystem provides perfect secrecy if the enemy’s observation is statistically independent of the plaintext, i.e., $I(X; Y) = 0$. Show that this implies Shannon’s much cited bound on key size

$$H(Z) \geq H(X),$$

i.e., perfect secrecy can only be attained if the entropy of the key (and hence its compressed length) is at least as large as the entropy of the secret plaintext.

- (c) Vernam’s cipher assumes a binary secret plaintext message X with any probability distribution $P_X(0) = p = 1 - P_X(1)$ and a binary secret key Z that’s uniform $P_Z(0) = P_Z(1) = 1/2$ and independent of X . The encrypter simply adds the secret key to the plaintext modulo 2, and the decrypter by adding the same key to the ciphertext can recover the plaintext. Show that Vernam’s cipher achieves perfect secrecy, i.e., $I(X; Y) = 0$.

- (a) Since the entropy of a function $f(\cdot)$ given its argument is zero, e.g., for any random variable X , $H(f(X)|X) = 0$, one condition is given by

$$H(Y|X, Z) = 0$$

because the ciphertext Y is a deterministic function of the secret plaintext message X and the secret key Z . The other condition is given by

$$H(X|Y, Z) = 0$$

because the secret plaintext message can be inferred from the ciphertext Y and the key Z .

- (b) Since the mutual information of X and Y satisfies

$$I(X; Y) = H(X) - H(X|Y) = 0,$$

we have

$$\begin{aligned}
H(X) &= H(X|Y) = H(X, Z|Y) - H(Z|X, Y) \\
&\leq H(X, Z|Y) \\
&= H(Z|Y) + H(X|Z, Y) \\
&= H(Z|Y) \\
&\leq H(Z).
\end{aligned}$$

The first line follows from the chain rule of entropies, specifically

$$\begin{aligned}
H(X, Y, Z) &= H(Y) + H(X|Y) + H(Z|X, Y) \\
\Rightarrow H(X|Y) &= H(X, Y, Z) - H(Y) - H(Z|X, Y) = H(X, Z|Y) - H(Z|X, Y).
\end{aligned}$$

The third line from the fact that in order to guarantee deterministic decryptability, $H(X|Y, Z) = 0$ (see point (a)). The last line is a consequence of the conditioning theorem.

☞ Recall that the conditioning theorem states that “conditioning on a random variable only ever reduces entropy”.

(c) We have to prove that

$$I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = 0.$$

To this end, we compute $H(Y)$ and $H(Y|X)$. Let us start with

$$H(Y) = -P_Y(0) \log P_Y(0) - (1 - P_Y(0)) \log(1 - P_Y(0)). \quad (2)$$

We have

$$\begin{aligned}
P_Y(0) &= \sum_{x,z} P_{XYZ}(x, 0, z) \\
&= \sum_{x,z} P_{Y|XZ}(0|x, z) P_X(x) P_Z(z) \\
&= p \frac{1}{2} + (1-p) \frac{1}{2} = \frac{1}{2},
\end{aligned}$$

where the last step follows from the fact that $P_{Y|XZ}(0|x, z) = 1$ if $x + z \bmod 2 = 0$, i.e. either $x + z = 2$ or $x + z = 0$, and $P_{Y|XZ}(0|x, z) = 0$ otherwise. By virtue of (2), the latter equation in turn implies

$$H(Y) = 1.$$

Now, in order to compute

$$H(Y|X) = P_X(0)H(Y|X=0) + P_X(1)H(Y|X=1),$$

we first calculate $H(Y|X=0)$ and $H(Y|X=1)$ using

$$H(Y|X=x) = -P_{Y|X}(0|x) \log P_{Y|X}(0|x) - P_{Y|X}(1|x) \log P_{Y|X}(1|x), \quad x = 0, 1. \quad (3)$$

We have

$$\begin{aligned}
P_{Y|X}(0|0) &= \frac{P_{XY}(0, 0)}{P_X(0)} \\
&= \frac{P_{XYZ}(0, 0, 0) + P_{XYZ}(0, 0, 1)}{P_X(0)} \\
&= \frac{P_X(0)P_Z(0)P_{Y|XZ}(0|0, 0) + P_X(0)P_Z(1)P_{Y|XZ}(0|0, 1)}{P_X(0)} \\
&= \frac{\frac{1}{2}p + 0}{p} = \frac{1}{2},
\end{aligned}$$

and similarly

$$\begin{aligned}
 P_{Y|X}(0|1) &= \frac{P_{XY}(0,1)}{P_X(1)} \\
 &= \frac{P_{XYZ}(1,0,0) + P_{XYZ}(1,0,1)}{P_X(1)} \\
 &= \frac{P_X(1)P_Z(0)P_{Y|XZ}(1|0,0) + P_X(1)P_Z(1)P_{Y|XZ}(1|0,1)}{P_X(1)} \\
 &= \frac{0 + \frac{1}{2}(1-p)}{1-p} = \frac{1}{2}.
 \end{aligned}$$

Therefore, by (3), we obtain

$$H(Y|X = 0) = H(Y|X = 1) = 1,$$

which in turn implies

$$H(Y|X) = P_X(0)H(Y|X = 0) + P_X(1)H(Y|X = 1) = \frac{1}{2}p + \frac{1}{2}(1-p) = 1.$$

Thus, we get

$$I(X;Y) = H(Y) - H(Y|X) = 1 - 1 = 0.$$

◇

Question 7. A symmetric binary communications channel operates with signalling levels of $\pm A$ volts at the detector in the receiver, and the rms noise level at the detector is B volts.

- If the output of this channel is quantised by a two-level quantiser with threshold 0, determine the probability of error on the resulting channel and hence, based on mutual information, calculate the theoretical capacity of this channel for error-free communication in bits per channel use. Compute a numerical value for $A = 2$ and $B = 0.5$.
- If the binary signalling were replaced by symbols drawn from a continuous process with a Gaussian (normal) pdf with zero mean and the same mean power at the detector, determine the theoretical capacity of this new channel. Again compute a numerical result for the same signal and noise power as in the previous question.
- (Computer Exercise) In MATLAB/Octave, plot the two capacities above in function of the Signal to Noise Ratio (SNR) on a scale from -5dB to 15dB, where $\text{SNR [dB]} = 10 \log_{10}(V_0^2/\sigma^2)$. A third channel of interest that is closely related to the two channels studied is the channel with binary signal levels (as in (a)) but continuous output (as in (b)). The capacity of this channel can only be computed numerically. You may use the following approximation:

```

s2 = 4*10^(SNR/10);
eta = linspace(-20,20,1e5);
x=exp(-(eta-s2/2).^2./(2*s2))/sqrt(2*pi*s2).*log(1+exp(-eta))/log(2);
C = 1-trapz(x)*(eta(2)-eta(1));

```

Compare the capacity of the three channels and discuss the practical implications of your findings.

- (a) From the equation (2.15) of the lecture notes of the 3F1 Random Processes part, we know that the probability of error in the binary detector is given by

$$\varepsilon = Q\left(\frac{A}{B}\right)$$

where $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-u^2/2} du$. Now, the channel with a two-level symmetric quantiser is equivalent to the Binary Symmetric Channel for which we know that capacity is achieved for uniform input symbols, giving rise to uniform output symbols, and it takes the form (see the lecture handouts)

$$C_{\text{BSC}} = 1 - h(\varepsilon),$$

where $h(x) := -x \log x - (1 - x) \log(1 - x)$ is the binary entropy function. For the values $A = 2$ and $B = 0.5$, we get

$$C_{\text{BSC}} = 0.99946 \text{ [bit/sample].}$$

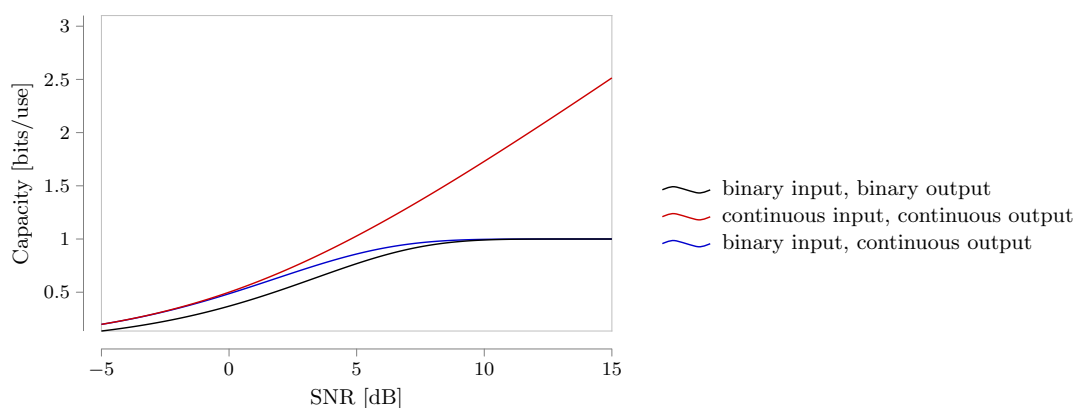
- (b) In this case the channel is an Additive White Gaussian Noise (AWGN) channel for which the capacity is given by (see the lecture handouts)

$$C_{\text{BSC}} = \frac{1}{2} \log\left(1 + \frac{A}{B}\right).$$

For the values $A = 2$ and $B = 0.5$, we get

$$C_{\text{BSC}} = \frac{1}{2} \log(1 + 16) = 2.0437 \text{ [bit/sample].}$$

- (c) The plot is shown below.



We observe that there is no loss at low SNR for using binary signalling as long as the output remains continuous. As the SNR increases, all binary signalling methods hit a 1 bit/use ceiling whereas the capacity for continuous signalling continues to grow unbounded. \diamond

Question 8. A discrete memoryless source has an alphabet of eight letters, x_i , $i = 1, 2, \dots, 8$ with probabilities 0.25, 0.20, 0.15, 0.12, 0.10, 0.08, 0.05 and 0.05.

- Determine the entropy of the source.
- Construct the Shannon-Fano code for this source. Try both Fano's and Shannon's constructions. Determine the average codeword length L .
- Use the Huffman algorithm to determine an optimal binary code for the source output. Determine the average codeword length L .

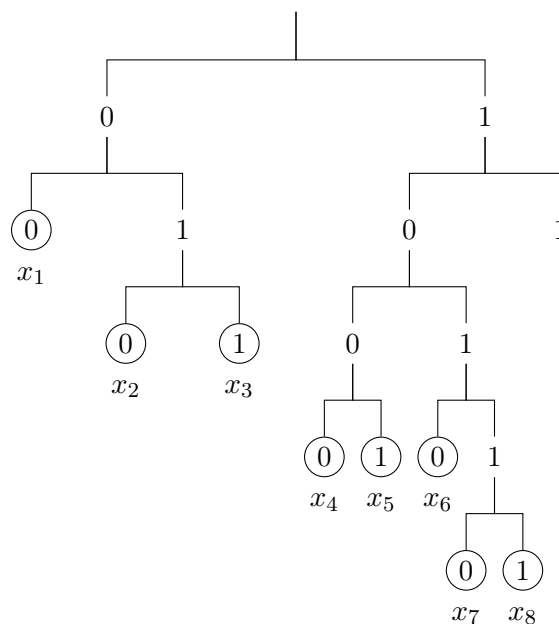
(a) The entropy of the source is given by

$$H(S) = -(0.25 \log 0.25 + 0.20 \log 0.20 + 0.15 \log 0.15 + 0.12 \log 0.12 + 0.10 \log 0.10 + 0.08 \log 0.08 + 0.05 \log 0.05) = 2.798 \text{ [bits]}.$$

(b) The table below shows, for each source symbol, the probabilities (p_i), cumulative probabilities (f_i), cumulative probabilities in binary notation ($f_{i,\text{bin}}$), codeword lengths ($\lceil -\log p_i \rceil$), Fano's (F) and Shannon's (S) codewords

symbol	p_i	f_i	$f_{i,\text{bin}}$	$\lceil -\log p_i \rceil$	F	S
x_1	0.25	0	0.0000000...	2	00	00
x_2	0.20	0.25	0.0100000...	3	010	010
x_3	0.15	0.45	0.0110011...	3	011	011
x_4	0.12	0.60	0.1001100...	4	1000	1001
x_5	0.10	0.72	0.1011100...	4	1001	1011
x_6	0.08	0.82	0.1101000...	4	1010	1101
x_7	0.05	0.90	0.1110011...	5	10110	11100
x_8	0.05	0.95	0.1111001...	5	10111	11110

Fano's codewords are obtained by growing a prefix-free tree to match the lengths $\lceil \log p_i \rceil$, as shown in the diagram below.

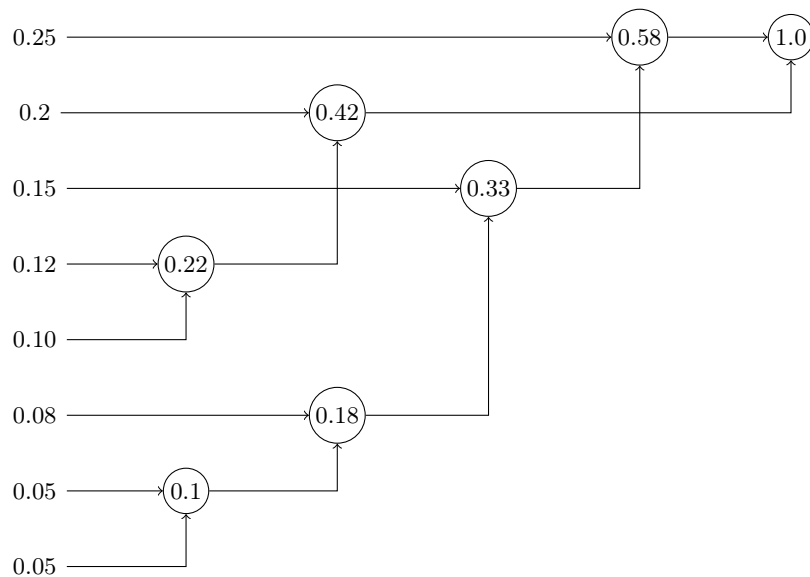


☞ Note that there is an unused leaf in the Fano's tree construction, namely 11. This shows clear potential to improve this code.

Shannon's construction consists in writing the cumulative probabilities in the third column of the table above in binary notation and truncating those at the specified lengths $\lceil \log p_i \rceil$ (in the clarification box at the end of this problem is described a procedure to perform decimal to binary conversion). The average codeword length L is the same Fano's and Shannon's codes, since they have the same codeword lengths, and is given by

$$\begin{aligned} L &= 0.25 \cdot 2 + 0.20 \cdot 3 + 0.15 \cdot 3 + 0.12 \cdot 4 + 0.10 \cdot 4 + 0.08 \cdot 4 + 0.05 \cdot 5 + 0.05 \cdot 5 \\ &= 3.25 \quad [\text{bits}]. \end{aligned}$$

- (c) The Huffman algorithm consists in merging the least probable symbols at every iteration. The procedure is described in the diagram below.



The Huffman's codewords (H), obtained from the previous diagram by labelling the upper branch 0 and the lower branch 1, are listed in the following table.

symbol	p_i	H
x_1	0.25	00
x_2	0.20	01
x_3	0.15	010
x_4	0.12	110
x_5	0.10	111
x_6	0.08	0110
x_7	0.05	01110
x_8	0.05	01111

The average codeword length is given by

$$\begin{aligned} L &= 0.25 \cdot 2 + 0.20 \cdot 2 + 0.15 \cdot 3 + 0.12 \cdot 3 + 0.10 \cdot 3 + 0.08 \cdot 4 + 0.05 \cdot 5 + 0.05 \cdot 5 \\ &= 2.83 \quad [\text{bits}]. \end{aligned}$$



There is a simple procedure to convert a decimal fraction to binary:

Step 1. Begin with the decimal fraction and multiply by 2. The whole number part of the result is the first binary digit to the right of the point. For instance, by considering the decimal fraction 0.625, we have $0.625 \times 2 = 1.25$, hence the first decimal digit of the right of the point is 1.

Step 2. We disregard the whole number part of the previous result and multiply by 2 once again. In our example, we have $0.25 \times 2 = 0.5$, hence the second decimal digit of the right of the point is 0.

Step 3. We iterate Step 2 until we get a zero as our decimal part or until we recognize an infinite repeating pattern. In our example, the second iteration of Step 2 yields $0.5 \times 2 = 1.00$. Since we get a zero as decimal part we conclude that the binary expansion of 0.625 is 0.101.

