INFORMATION THEORY AND APPLICATIONS
lecture notes/laboratories

© Prof. Giorgio Taricco ©

Politecnico di Torino

2019/20

Copyright Notice

Outline

Section Outline

Structure of the Course

The course addresses the following topics (50 h):

- Characterization of an information source
- Source coding
- Discrete communication channels
- Information theory for secure communications

References

- T.M. Cover and J.A. Thomas, *Elements of Information Theory*. Wiley, 2006.
- G. Grimmett and D. Stirzaker, *Probability and Random Processes 3rd ed*. Oxford, 2001.
- G. Grimmett and D. Stirzaker, *One Thousand Exercises in Probability*. Oxford, 2001.
- M. Lefebvre, *Basic Probability Theory with Applications*. Springer Undergraduate Texts in Mathematics and Technology, 2009.
- A. El Gamal and Y.-H. Kim, *Network Information Theory*. Cambridge University Press, 2011.

Outline

Section Outline

2. Entropy of discrete random variables
   - Probability and Random Variables
   - Information sources
   - Information content and measure
   - Entropy and Mutual Information
   - Entropy rate of a source
   - Laboratory: Calculation of entropies
   - Laboratory: Test of entropy inequalities
   - Laboratory: Computation of entropy rates
   - Laboratory: Entropy rate of a language

Probability

- Probability is based on the concept of probability space, consisting in the following three components:
  1. A set containing all possible outcomes: Ω.
  2. A set of events, $\mathcal{F}$, whose elements are subsets of outcomes: $\mathcal{F} \subseteq \Omega$.
  3. A probability function $P : \mathcal{F} \mapsto [0, 1]$, assigning a probability to every event.
- The probability function is a normalized measure
  1. In the discrete case (when Ω is a finite or countably infinite set), the probability of the event $E \in \mathcal{F}$ is the sum of the probabilities of the outcomes in E:

$$P(E) = \sum_{\omega \in E} P(\omega).$$

     The outcomes are events themselves with positive probabilities $P(\omega)$.

Probability (cont.)

2. In the continuous case, it is an integral of the probability density function (pdf) over the event $E \in \mathcal{F}$:

$$P(E) = \int_{\omega \in E} d\mu(\omega).$$

   In this case, the outcomes are uncountable and are not events.
3. The set Ω is also an event: $\Omega \in \mathcal{F}$. The following normalization holds:

$$P(\Omega) = 1,$$

   that is, the probability of the set of all possible outcomes is equal to 1.

Probability (cont.)

- Given two events A, B, we can build the union $A \cup B$ and the intersection $A \cap B$:
  - $A \cup B =$ set of outcomes in A or in B.
  - $A \cap B =$ set of outcomes in A and in B.
- Some additional technical assumptions on $\mathcal{F}$ are made in order that probabilities can be calculated properly (σ-algebra assumptions):
  - If $E \in \mathcal{F}$ also its complement $\Omega \backslash E \in \mathcal{F}$.
  - If $E_n \in \mathcal{F}$ for $n = 1, 2, 3, \ldots$ (possibly a countable infinite sequence of event), then $\cup_n E_n \in \mathcal{F}$, as well.

Probability (cont.)

- The probability of the intersection is called joint probability:

$$P(A, B) \equiv P(A \cap B)$$

- The conditional probability is

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

- An important result is the total probability law.
- If the events $B_i, i = 1, 2, \ldots$ form a partition of Ω (i.e., $\uplus_i B_i = \Omega$ and $B_i \cap B_j = \emptyset$ for $i \neq j$), then:

$$P(A) = \sum_i P(A, B_i) = \sum_i P(A \mid B_i) P(B_i)$$

Example 1

- Consider the probability space corresponding to the outcome of a flipped coin, i.e., heads or tails.
- The outcomes are

$$\omega_1 = H, \omega_2 = T$$

- The set of events $\mathcal{F}$ contains $2^2 = 4$ different events:

$$E_0 = \emptyset, E_1 = \{\omega_1\}, E_2 = \{\omega_2\}, E_3 = \{\omega_1, \omega_2\}$$

- If the coin is fair, then

$$P(\omega_1) = P(\omega_2) = \frac{1}{2}$$

Example 1 (cont.)

- The probabilities of the events in $\mathcal{F}$ are

$$P(E_0) = 0, P(E_1) = \frac{1}{2}, P(E_2) = \frac{1}{2}, P(E_3) = 1$$

- In general, for finite probability spaces,

$$|\mathcal{F}| = 2^{|\Omega|}$$

because every event may contain, or not, each outcome from Ω.

Example 2

- Consider the probability space corresponding to the outcome of a die (a cube with 6 faces numbered from 1 to 6).
- The outcomes are

$$\omega_1 = 1, \omega_2 = 2, \omega_3 = 3, \omega_4 = 4, \omega_5 = 5, \omega_6 = 6$$

- An event can be the outcome 2: $E = \{\omega_2\}$.
- Another event can be an outcome ≤ 3: $E = \{\omega_1, \omega_2, \omega_3\}$.
- Another event can be an outcome which is an odd number: $E = \{\omega_1, \omega_3, \omega_5\}$.

Example 2 (cont.)

- The set of events $\mathcal{F}$ contains $2^6 = 64$ different events:

$$\emptyset, \{\omega_1\}, \{\omega_2\}, \{\omega_3\}, \{\omega_4\}, \{\omega_5\}, \{\omega_6\},$$
$$\{\omega_1, \omega_2\}, \{\omega_1, \omega_3\}, \ldots \{\omega_5, \omega_6\},$$
$$\{\omega_1, \omega_2, \omega_3\}, \{\omega_1, \omega_2, \omega_4\}, , \ldots \{\omega_4, \omega_5, \omega_6\}, \ldots$$
$$\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4, \omega_5, \omega_6\}$$

- Assume that the die is fair and then $P(\omega_i) = 1/6$ for $i = 1, \ldots, 6$.

Example 2 (cont.)

- Then, we have

$$\begin{aligned} P(\{\omega_2\}) &= \frac{1}{6} \\ P(\{\omega_1,\omega_2,\omega_3\}) &= \frac{3}{6} \\ P(\{\omega_1,\omega_3,\omega_5\}) &= \frac{3}{6} \\ P(\Omega=\{\omega_1,\omega_2,\omega_3,\omega_4,\omega_5,\omega_6\}) &= \frac{6}{6} = 1 \end{aligned}$$

Random variables

- A discrete random variable X is characterized by its probability distribution

$$p_X(x_n) = P(X = x_n)$$

for $n = 1, 2, \ldots, N$ (where N may become infinity).

- The expectation operator $\mathbb{E}[\cdot]$ is defined by

$$\mathbb{E}[\varphi(X)] = \sum_n \varphi(x_n) p_X(x_n)$$

for an arbitrary function $\varphi(\cdot)$.

Random variables (cont.)

- Since the expected value of every constant is the constant itself, we obtain by definition:

$$\mathbb{E}[1] = \sum_n p_X(x_n) = 1.$$

  This property holds for all probability distributions.
- The mean of X is $\mu_X = \mathbb{E}[X] = \sum_n x_n p_X(x_n)$.
- The second moment of X is $\mu_X^{(2)} = \mathbb{E}[X^2] = \sum_n x_n^2 p_X(x_n)$.
- The variance of X is $\sigma_X^2 = \mathbb{E}[(X - \mu_X)^2] = \mu_X^{(2)} - \mu_X^2$.
- The square root of the variance is called standard deviation.

Examples

- Calculate the mean and variance of a random variable X with probability distribution

x	-2	-1	0	1	2
$P(X=x)$	0.1	0.2	0.4	0.2	0.1

- Calculate the mean and variance of the sum X of independent random variables X_i whose means and variances are μ_i and σ_i^2, respectively, for $i = 1, \ldots, N$.

Examples (cont.)

- Given the joint probability distribution

$P(X=x,Y=y)$	$y=0$	1	2
$x=0$	0.3	0.1	0.1
	0.1	0.1	0.3

calculate the probabilities

$$P(X=0), \qquad P(X=0|Y=1) \qquad P(Y=2|X=1)$$

Information sources

- An information source is a device that outputs (at a certain rate) a bit sequence representing any kind of information.
- An information source is characterized as a sequence of discrete random variables (X_n) and the set of all possible probabilities

$$P(X_{\mathcal{S}} = x_{\mathcal{S}})$$

where $\mathcal{S}$ are all possible index sets, $X_{\mathcal{S}} \triangleq \{X_n\}_{n\in\mathcal{S}}$, $x_{\mathcal{S}} \triangleq \{x_n\}_{n\in\mathcal{S}}$, and $x_n \in \mathcal{X} \triangleq \{\xi_1, \ldots, \xi_M\}$, the source alphabet.

Information sources (cont.)

- Example
- We take as alphabet $\mathcal{X} = \{A, B, C, \ldots, Z\}$, the 26-letter English alphabet
- The previous notation allows us to take $\mathcal{S} = \{0, 2, 4\}$, and $x_0 = A, x_2 = B, x_4 = C$, or $x_{\mathcal{S}} = (A, B, C)$.
- Then,

$$P(X_{\mathcal{S}} = x_{\mathcal{S}}) = P(X_0 = A, X_2 = B, X_4 = C)$$

Information sources (cont.)

- The general definition of an information source is often specialized by one of the following additional properties:
  - Stationarity:

    $$P(X_{\mathcal{S}} = x_{\mathcal{S}}) = P(X_{\mathcal{S}+\Delta} = x_{\mathcal{S}})$$

    for every index set $\mathcal{S} = \{n_1, \ldots, n_{|\mathcal{S}|}\}$ and integer offset Δ, where

    $$\mathcal{S} + \Delta \triangleq \{n_1 + \Delta, \ldots, n_{|\mathcal{S}|} + \Delta\}.$$

    If we take $\mathcal{S} = \{n\}$, a single-index set, we can see that for a stationary source we have

    $$P(X_n = x) = P(X_{n+\Delta} = x) = p_X(x)$$

    independently of the index n.

Information sources (cont.)

- Markovianity: A Markovian source $(X_n)_{n=0}^{\infty}$ with memory L satisfies the property

$$\begin{aligned} P(X_n = x_n | X_{\{0:n-1\}} = x_{\{0:n-1\}}) \\ = P(X_n = x_n | X_{\{n-L:n-1\}} = x_{\{n-L:n-1\}}) \end{aligned}$$

  In other words, X_n depends only on the previous L source symbols $X_{\{n-L:n-1\}}$.

- Notation: Given two integers a, b, the notation $a : b$ represents all the integers between a and b if $a \leq b$, i.e.:

$$a, a+1, \ldots, b$$

  Otherwise, if $a > b$, it represents the empty set $\emptyset$.

Information sources (cont.)

- The Markovian property is useful in many practical cases, such as the case when an information source represents the spoken language.
- For example, in English we cannot find a 'Z' followed by a 'Q' so that the probability of emitting a 'Q' given that the previous character was a 'Z' is zero.
- On the other hand, after 'TH' we find a vowel with high probability, but in this case we have to consider two preceding characters.

Information sources (cont.)

- How can we estimate the conditional probability distribution of a language?
- First of all we should add the space to our alphabet, which becomes of 27 symbols, 26 letters plus the space.
- We neglect upper and lower case, for simplicity, as well as punctuation symbols.
- Then, we fix a memory, let's say $L = 2$ (the greater the memory, the better the accuracy, but the higher the complexity).
- Then, we take a long English text, for example a book, or a magazine, and scan it sequentially by counting the number of occurrences of each possible triple, there are 27^3. For instance: 'AAA','AAB', etc.
- Finally, we obtain a probability distribution of the triples by dividing the number of times a certain sequence $\{x_1, x_2, x_3\}$ appears by the total number of triples scanned from the source.

Information sources (cont.)

- So far, we have obtained the joint probability

$$P(X_1 = x_1, X_2 = x_2, X_3 = x_3)$$

How do we obtain the conditional probability?

- We need $P(X_1 = x_1, X_2 = x_2)$ since

$$P(X_3 = x_3 | X_1 = x_1, X_2 = x_2) = \frac{P(X_1 = x_1, X_2 = x_2, X_3 = x_3)}{P(X_1 = x_1, X_2 = x_2)}$$

- To this end we apply the total probability law:

$$P(X_1 = x_1, X_2 = x_2) = \sum_{x_3 \in \{`A', `B', \ldots, `Z', ` '\}} P(X_1 = x_1, X_2 = x_2, X_3 = x_3)$$

Information sources (cont.)

- The first two symbols in each triple can be interpreted as the state of the source, so that we have 27^2 states.
- We can estimate the state transition probabilities from the joint probability as follows:

$$\begin{aligned} P(X_2 = x_2, X_3 = x_3 | X_1 = x_1, X_2 = x_2) \\ &= P(X_3 = x_3 | X_1 = x_1, X_2 = x_2) \\ &= \frac{P(X_1 = x_1, X_2 = x_2, X_3 = x_3)}{P(X_1 = x_1, X_2 = x_2)} \end{aligned}$$

Information sources (cont.)

- Example. We use the text sample "THE BOOK IS ON THE TABLE"
- The triples to consider are:
  'THE',' BO','OK ','IS ','ON ','THE',' TA','BLE'
- We count the occurrences of each triple and obtain their frequency dividing by the total number of them.
- To get good results we need a long text sample.

Information sources (cont.)

- Independence: An independent information source satisfies the property

$$P(X_{\mathcal{S}} = x_{\mathcal{S}}) = \prod_{n \in \mathcal{S}} P(X_n = x_n)$$

  for every index set $\mathcal{S}$ and all possible $x_{\mathcal{S}}$.

- A stationary independent information source satisfies the property

$$P(X_{\mathcal{S}} = x_{\mathcal{S}}) = \prod_{n \in \mathcal{S}} p_X(x_n)$$

  for every index set $\mathcal{S}$ and a given probability distribution $p_X(x)$.

Information content and measure

- Information theory was discovered by Claude E. Shannon in 1948 to study the quantitative meaning of information flow.
- The discipline finds applications in many fields, such as
  - source coding for data compression,
  - channel coding for error protection,
  - cryptography for secure communications.
- A basic idea from information theory is assigning an information measure to events, which is what we are going to investigate in the following.

Information content and measure (cont.)

- This measure is called information content and is given by

$$I(E) \triangleq \log_2 \frac{1}{P(E)} \text{ bits.}$$

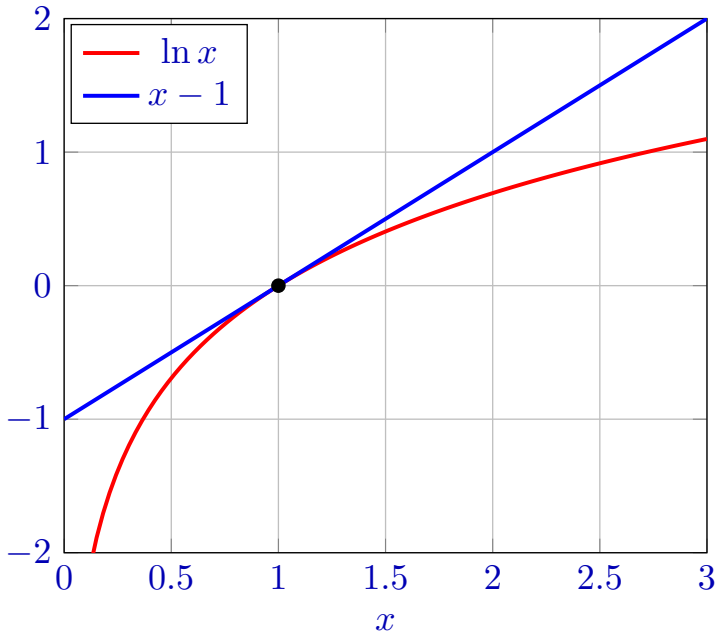
- Therefore:
  - unlikely events possess high information content;
  - common events possess low information content;
  - the information content increases slowly (because of the logarithm) with the inverse of the probability.
- The average information content of a random variable is called entropy of the random variable.

Entropy and Mutual Information

- **A basic tool: The logarithmic Inequality**
  The logarithmic inequality is given by

$$\ln x \leq x - 1$$

for every $x > 0$ with equality only if $x = 1$.



Entropy and Mutual Information (cont.)

- Given a random variable X with discrete alphabet $\mathcal{X}$ and probability distribution $p_X(x)$ defined for all $x \in \mathcal{X}$, we define its entropy as:

$$H(X) \triangleq -\sum_{x \in \mathcal{X}} p_X(x) \log_2 p_X(x) \text{ bits..} \tag{1}$$

- Sometimes entropy is measured in nats by taking $\ln(\cdot) \equiv \log_{\mathrm{e}}(\cdot)$ instead of $\log_2(\cdot)$.
- If $p_X(x) = 0$, we assume $0 \log_2 0 = 0$, since $\lim_{\varepsilon \downarrow 0} \varepsilon \log_2 \varepsilon = 0$.
- Since probabilities are not greater than 1, the entropy is always nonnegative: $H(X) \geq 0$.

Entropy and Mutual Information (cont.)

- We define the entropy function of a probability vector

$$\boldsymbol{p} = (p_1, \ldots, p_N),$$

such that $0 \leq p_i \leq 1$ and $\sum_{i=1}^{N} p_i = 1$, as follows:

$$\begin{aligned} H(\boldsymbol{p}) &= H(p_1, \ldots, p_N) \\ &\triangleq \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} \\ &= -\sum_{i=1}^{N} p_i \log_2 p_i \end{aligned}$$

Zero probabilities must be disregarded:

$$H(p_1, 0, p_2, p_3) = H(p_1, p_2, p_3).$$

Entropy and Mutual Information (cont.)

- The entropy of a random variable is a measure of the uncertainty of its value.
- If a quaternary random variable takes one value with high probability and the other three ones with small probabilities, than the entropy is very low. For example,

$$H(0.01, 0.01, 0.01, 0.97) = 0.2419.$$

- If instead the probabilities are equal:

$$H(0.25, 0.25, 0.25, 0.25) = 2.$$

- This is the maximum uncertainty for a quaternary random variable, as we are going to see.

Entropy and Mutual Information (cont.)

- We have the following

Entropy inequalities:

$$0 \leq H(\boldsymbol{p}) \leq \log_2 N$$

- The lower bound is trivial since $-p_i \log_2 p_i > 0$ for $p_i > 0$ and is equal to 0 for $p_i = 0$.

Entropy and Mutual Information (cont.)

- The upper bound derives from the logarithmic inequality:

$$
\begin{aligned}
H(\boldsymbol{p}) - \log_2 N &= \frac{1}{\ln 2}\sum_{i=1}^{N} p_i \ln \frac{1}{p_i} - \frac{\ln N}{\ln 2} \\
&= \frac{1}{\ln 2}\sum_{i=1}^{N} p_i \ln \frac{1}{p_i} - \frac{1}{\ln 2}\sum_{i=1}^{N} p_i \ln N \\
&= \frac{1}{\ln 2}\sum_{i=1}^{N} p_i \ln \frac{1}{Np_i} \leq \frac{1}{\ln 2}\sum_{i=1}^{N} p_i \left(\frac{1}{Np_i} - 1\right) \\
&= \frac{1}{\ln 2}\sum_{i=1}^{N} \left(\frac{1}{N} - p_i\right) = \frac{1-1}{\ln 2} = 0
\end{aligned}
$$

Entropy and Mutual Information (cont.)

- Another property of the entropy function is that averaging a subset of probabilities in-place can only increase the entropy itself.
- For example:

$$\begin{aligned} & H(0.05, 0.35, \boxed{0.2, 0.4}) = 1.7394 \\ < \quad & H(0.05, 0.35, \boxed{0.3, 0.3}) = 1.7884 \end{aligned}$$

where the last two probabilities have been averaged in-place.

Entropy and Mutual Information (cont.)

Proof

- Assume (without loss of generality since the entropy function is invariant to permutation of the arguments) that the probability vector is divided in two parts, the sub-vectors $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, and the probabilities in $\boldsymbol{p}_1$ are replaced by the average value of $\boldsymbol{p}_1$ itself.
- Let μ be the average value of $\boldsymbol{p}_1$ and $\bar{\boldsymbol{p}}_1 \triangleq (\mu, \ldots, \mu)$ be a vector with the same length as $\boldsymbol{p}_1$.
- Our goal is to show that

$$H(\boldsymbol{p}_1, \boldsymbol{p}_2) \leq H(\bar{\boldsymbol{p}}_1, \boldsymbol{p}_2) = H(\mu, \ldots, \mu, \boldsymbol{p}_2). \tag{2}$$

Entropy and Mutual Information (cont.)

- We calculate the difference

$$
\begin{aligned}
H(\boldsymbol{p}_1,\boldsymbol{p}_2)-H(\bar{\boldsymbol{p}}_1,\boldsymbol{p}_2) &= \sum_i (\boldsymbol{p}_1)_i \log_2 \frac{1}{(\boldsymbol{p}_1)_i} - \sum_i \mu \log_2 \frac{1}{\mu} \\
&\quad + \sum_j (\boldsymbol{p}_2)_j \log_2 \frac{1}{(\boldsymbol{p}_2)_j} - \sum_j (\boldsymbol{p}_2)_j \log_2 \frac{1}{(\boldsymbol{p}_2)_j} \\
&= \sum_i (\boldsymbol{p}_1)_i \log_2 \frac{1}{(\boldsymbol{p}_1)_i} - \sum_i (\boldsymbol{p}_1)_i \log_2 \frac{1}{\mu} \\
&= \sum_i (\boldsymbol{p}_1)_i \log_2 \frac{\mu}{(\boldsymbol{p}_1)_i} \\
&\overset{\leq}{=} \frac{1}{\ln 2} \sum_i (\boldsymbol{p}_1)_i \left(\frac{\mu}{(\boldsymbol{p}_1)_i} - 1 \right) = 0
\end{aligned}
$$

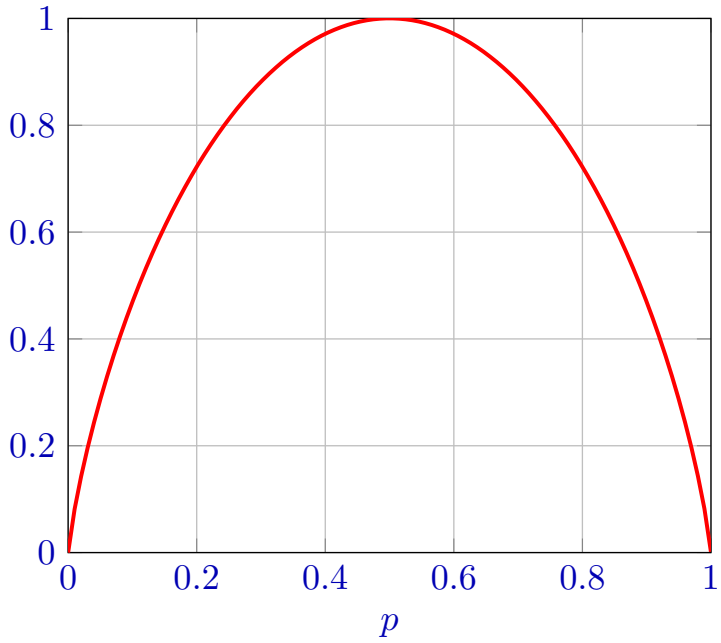
- Here we used the fact that $\sum_i (\boldsymbol{p}_1)_i = \sum_i \mu$ and applied the logarithmic inequality $\ln x \leq x - 1$.

Entropy and Mutual Information (cont.)

- The entropy of a binary random variable with probability vector $(p, 1-p)$ is

$$H_b(p) \triangleq H(p, 1-p) = -p \log_2 p - (1-p) \log_2 (1-p)$$

where $0 \leq p \leq 1$. Plainly, $H_b(0.5) = 1$ and the function plot is:



Entropy and Mutual Information (cont.)

- The joint entropy of two random variables X and Y is defined as

$$H(X,Y) \triangleq -\sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_{XY}(x,y)\log_2 p_{XY}(x,y). \tag{3}$$

  The concept can be extended directly to more than two random variables.

- Notice that the joint entropy is invariant to permutations of random variables. For example, $H(X,Y,Z) = H(Z,Y,X)$.

- The conditional entropy of two random variables X and Y is defined as

$$H(X|Y) \triangleq -\sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_{XY}(x,y)\log_2 p_{X|Y}(x|y). \tag{4}$$

Entropy and Mutual Information (cont.)

- Given a discrete random variable X and a function $\varphi(x)$ we have the following property:

$$H(\varphi(X)) \leq H(X).$$

- **Proof**

  Let $\boldsymbol{p}$ be the probability vector corresponding to the joint distribution of X, so that $H(X) = H(\boldsymbol{p})$.

  The probability vector of $\varphi(X)$ is obtained by adding subsets of probabilities from $\boldsymbol{p}$ since

$$P(\varphi(X) = z) = \sum_{\varphi(x)=z} P(X = x)$$

Entropy and Mutual Information (cont.)

The inequality follows by repeatedly applying the property

$$H(p_1 + p_2, \boldsymbol{p}_3) \leq H(p_1, p_2, \boldsymbol{p}_3),$$

which is equivalent to

$$(p_1 + p_2) \log_2 \frac{1}{p_1 + p_2} \leq p_1 \log_2 \frac{1}{p_1} + p_2 \log_2 \frac{1}{p_2},$$

and

$$p_1 \log_2 \frac{p_1}{p_1 + p_2} + p_2 \log_2 \frac{p_2}{p_1 + p_2} \leq 0,$$

which holds because both arguments of the logarithms are ≤ 1.

Entropy and Mutual Information (cont.)

- The mutual information between two random variables is defined as

Mutual information

$$I(X;Y) \triangleq \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_{XY}(x,y)\log_2 \frac{p_{XY}(x,y)}{p_X(x)p_Y(y)}. \quad (5)$$

- The mutual information represents the amount of information about one random variable (*e.g.*, X) obtained from the observation of the other random variable (*e.g.*, Y).
- If X and Y are independent, the observation of Y doesn't tell anything about X and the mutual information is zero.

Entropy and Mutual Information (cont.)

- The mutual information is always nonnegative. In fact,

$$
\begin{aligned}
-I(X;Y) &= \frac{1}{\ln 2} \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_{XY}(x,y) \ln \frac{p_X(x)p_Y(y)}{p_{XY}(x,y)} \\
&\leq \frac{1}{\ln 2} \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_{XY}(x,y) \left(\frac{p_X(x)p_Y(y)}{p_{XY}(x,y)} - 1 \right) \\
&= \frac{1}{\ln 2} \left(\sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_X(x)p_Y(y) - \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_{XY}(x,y) \right) \\
&= \frac{1}{\ln 2} \left(\sum_{x\in\mathcal{X}} p_X(x) \sum_{y\in\mathcal{Y}} p_Y(y) - \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_{XY}(x,y) \right) \\
&= 0.
\end{aligned}
$$

Entropy and Mutual Information (cont.)

- In order that $I(X;Y) = 0$, all the arguments in the logarithms must be equal to 1.
- Then, we have the condition

$$\frac{p_X(x)p_Y(y)}{p_{XY}(x,y)} = 1 \quad \forall\,(x,y) \in \mathcal{X} \times \mathcal{Y}$$

- This condition is equivalent to

$$p_{XY}(x,y) = p_X(x)p_Y(y) \quad \forall\,(x,y) \in \mathcal{X} \times \mathcal{Y},$$

which is satisfied if and only if X and Y are statistically independent.

Entropy and Mutual Information (cont.)

- The mutual information can be expressed by using different types of entropies. We can see that

$$\begin{aligned} I(X;Y) &= H(X) + H(Y) - H(X,Y) \\ &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X). \end{aligned} \tag{6}$$

- The property $I(X;Y) \geq 0$ implies the inequality

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

  i.e., conditioning reduces the entropy.

- Notice that the mutual information is invariant to the exchange of the random variables. For example, $I(X;Y) = I(Y;X)$.

Entropy and Mutual Information (cont.)

- In order to measure the distance between probability distributions we use the relative entropy or Kullback-Leibner distance.
- Given two probability vectors $\boldsymbol{p} = (p_1, \ldots, p_N)$ and $\boldsymbol{q} = (q_1, \ldots, q_N)$, the relative entropy or Kullback-Leibner distance is defined as

$$D(\boldsymbol{p}||\boldsymbol{q}) \triangleq \sum_{i=1}^{N} p_i \log_2 \frac{p_i}{q_i}. \tag{7}$$

- For example, let $\boldsymbol{p} = (0.25, 0.25, 0.25, 0.25)$ and $\boldsymbol{q} = (0.7, 0.1, 0.1, 0.1)$. Then, it is easy to calculate

$$D(\boldsymbol{p}||\boldsymbol{q}) = 0.6201, \qquad D(\boldsymbol{q}||\boldsymbol{p}) = 0.6432$$

- We can see that $D(\boldsymbol{p}||\boldsymbol{q}) \neq D(\boldsymbol{q}||\boldsymbol{p})$, in general.

Entropy and Mutual Information (cont.)

- Also, $D(\boldsymbol{p}||\boldsymbol{q}) \geq 0$ by the logarithmic inequality applied to $-D(\boldsymbol{p}||\boldsymbol{q})$:

$$
\begin{aligned}
-D(\boldsymbol{p}||\boldsymbol{q}) &= \sum_{i=1}^{N} p_i \log_2 \frac{q_i}{p_i} \\
&= \frac{1}{\ln 2} \sum_{i=1}^{N} p_i \ln \frac{q_i}{p_i} \\
&\leq \frac{1}{\ln 2} \sum_{i=1}^{N} p_i \left(\frac{q_i}{p_i} - 1 \right) \\
&= \frac{1}{\ln 2} \sum_{i=1}^{N} (q_i - p_i) \\
&= \frac{1}{\ln 2} \left\{ \sum_{i=1}^{N} q_i - \sum_{i=1}^{N} p_i \right\} = 0
\end{aligned}
$$

Entropy and Mutual Information (cont.)

- Additionally, $D(\boldsymbol{p}||\boldsymbol{q}) = 0$ only if $\boldsymbol{p} = \boldsymbol{q}$.
- The mutual information can be interpreted as the Kullback-Leibner distance between the joint probability and the product of the marginal probabilities.
- Since

$$I(X;Y) = \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} p_{XY}(x,y)\log_2\frac{p_{XY}(x,y)}{p_X(x)p_Y(y)}$$

$$D(\boldsymbol{p}||\boldsymbol{q}) = \sum_{i=1}^{N} p_i \log_2 \frac{p_i}{q_i}$$

we have

$$I(X;Y) = D(p_{XY}(x,y) \;||\; p_X(x)p_Y(y))$$

Entropy rate of a source

- An information source is an infinite sequence of random variables and, as such, its entropy tends to infinity.
- Therefore, the entropy rate of an information source is considered, according to the following definition:

$$\bar{H} = \lim_{n\to\infty} \frac{1}{n} H(X_{1:n}) \tag{8}$$

Entropy rate of a source (cont.)

- For a stationary independent source, we have

$$\begin{aligned} \bar{H} &= \lim_{n\to\infty} \frac{1}{n} H(X_{1:n}) \\ &= \lim_{n\to\infty} \frac{1}{n} [H(X_1) + \cdots + H(X_n)] \\ &= H(X) \end{aligned} \tag{9}$$

  where $H(X) = H(X_1) = \cdots = H(X_n)$.

- If the source is stationary, but not independent, we can show that

$$\bar{H} = \lim_{n\to\infty} \frac{1}{n} H(X_{1:n}) = \lim_{n\to\infty} H(X_n|X_{1:n-1}). \tag{10}$$

Entropy rate of a source (cont.)

- If the source is stationary Markovian with memory L, then the entropy rate is given by

$$\bar{H} = H(X_n|\Sigma_n)$$

  where Σ_n corresponds to the symbols $X_{n-L:n-1}$.
- Σ_n is interpreted as the source state at time n.
- The index n can be dropped because of the stationarity of the source, so that we write

$$\bar{H} = H(X|\Sigma) \tag{11}$$

Entropy rate of a source (cont.)

- The calculation of $H(X|\Sigma)$ requires the state probability distribution, $P(\Sigma = \sigma)$ and the conditional distribution $P(X = x|\Sigma = \sigma)$. Then,

$$H(X|\Sigma) = \sum_{\sigma} P(\Sigma = \sigma) H(X|\Sigma = \sigma)$$

$$= \sum_{\sigma} P(\Sigma = \sigma) \sum_{x} P(X = x|\Sigma = \sigma) \log_2 \frac{1}{P(X = x|\Sigma = \sigma)}$$

- The entropy rate is linked to the minimum number of bits per symbol required in source encoding, which is analyzed in the followng.
- The entropy $H(X|\Sigma = \sigma)$ is easy to calculate if we have the conditional distribution of the source output symbol given the state.

Entropy rate of a source (cont.)

- To calculate the steady-state state distribution we define the state transition matrix $\boldsymbol{P}$ by

$$(\boldsymbol{P})_{i,j} = P(\Sigma_n = \sigma_j | \Sigma_{n-1} = \sigma_i)$$

  and the steady-state probability vector $\boldsymbol{q}$ such that $(\boldsymbol{q})_i = P(\Sigma_n = \sigma_i)$.
- Next, we solve the linear equations

$$\boldsymbol{q}^\mathsf{T}\boldsymbol{P} = \boldsymbol{q}^\mathsf{T} \qquad \mathbf{1}^\mathsf{T}\boldsymbol{q} = 1$$

  where $\mathbf{1}$ is an all-1 vector.

Entropy rate of a source (cont.)

- Example 1: consider a stationary independent binary source characterized by $P(X_n = 0) = 0.2$ and calculate the entropy rate.
- Solution: since the source is stationary independent,

$$\bar{H} = H(X) = H_b(0.2) = 0.7219.$$

Entropy rate of a source (cont.)

- Example 2: calculate the entropy rate of a Markovian binary source characterized by

$$P(X_n = 0 | X_{n-3:n-1} = x_{n-3:n-1}) = \begin{cases} 0.2 & x_{n-3} + x_{n-2} + x_{n-1} \leq 1 \\ 0.5 & x_{n-3} + x_{n-2} + x_{n-1} \geq 2 \end{cases}$$

Obviously,

$$\begin{aligned} P(X_n = 1 | X_{n-3:n-1} = x_{n-3:n-1}) &= 1 - P(X_n = 0 | X_{n-3:n-1} = x_{n-3:n-1}) \\ &= \begin{cases} 0.8 & x_{n-3} + x_{n-2} + x_{n-1} \leq 1 \\ 0.5 & x_{n-3} + x_{n-2} + x_{n-1} \geq 2 \end{cases} \end{aligned}$$

Entropy rate of a source (cont.)

- Solution
  - We have 8 possible states characterized by $\sigma_n \equiv x_{n-3:n-1}$ and we need to find the state transition probabilities

$$P(\Sigma_n = \sigma_n | \Sigma_{n-1} = \sigma_{n-1}) = P(X_{n-3:n-1} = x_{n-3:n-1} | X_{n-4:n-2} = x_{n-4:n-2})$$

  - For example,

$$\begin{aligned} P(X_{n-2:n} = (0,0,0) | X_{n-3:n-1} = (0,0,0)) &= P(X_n = 0 | X_{n-3:n-1} = (0,0,0)) \\ &= 0.2 \end{aligned}$$

since $0 + 0 + 0 \leq 1$, i.e., it satisfies the first condition.

Entropy rate of a source (cont.)

- The transition probabilities are summarized in the following table:

	000	001	010	011	100	101	110	111
000	0.2	0.8	0.0	0.0	0.0	0.0	0.0	0.0
001	0.0	0.0	0.2	0.8	0.0	0.0	0.0	0.0
010	0.0	0.0	0.0	0.0	0.2	0.8	0.0	0.0
011	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.5
100	0.2	0.8	0.0	0.0	0.0	0.0	0.0	0.0
101	0.0	0.0	0.5	0.5	0.0	0.0	0.0	0.0
110	0.0	0.0	0.0	0.0	0.5	0.5	0.0	0.0
111	0.0	0.0	0.0	0.0	0.0	0.0	0.5	0.5

- These probabilities can be collected into a matrix $\boldsymbol{P}$ whose entries represent the transition probabilities from the ith state to the jth state, where (i, j) are the row and column indexes.

Entropy rate of a source (cont.)

- The state probability vector $\boldsymbol{q}_n \triangleq \{P(\Sigma_n = \sigma_i)\}_{i=1}^{8}$ evolves according to the matrix equation

$$\boldsymbol{q}_{n+1}^{\mathsf{T}} = \boldsymbol{q}_n^{\mathsf{T}} \boldsymbol{P}$$

- Under certain conditions it converges to a limit. In our case, starting from $\boldsymbol{q}_0^{\mathsf{T}} = (1, 0, 0, 0, 0, 0, 0, 0)$, we have:

n	$\boldsymbol{q}_n^{\mathsf{T}}$
1	$(0.2000, 0.8000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000, 0.0000)$
2	$(0.0400, 0.1600, 0.1600, 0.6400, 0.0000, 0.0000, 0.0000, 0.0000)$
3	$(0.0080, 0.0320, 0.0320, 0.1280, 0.0320, 0.1280, 0.3200, 0.3200)$
$\vdots$	
16	$(0.0259, 0.1037, 0.1036, 0.1659, 0.1035, 0.1658, 0.1658, 0.1658)$
17	$(0.0259, 0.1036, 0.1036, 0.1658, 0.1036, 0.1658, 0.1659, 0.1659)$
18	$(0.0259, 0.1036, 0.1036, 0.1658, 0.1036, 0.1658, 0.1658, 0.1658)$
19	$(0.0259, 0.1036, 0.1036, 0.1658, 0.1036, 0.1658, 0.1658, 0.1658)$

Entropy rate of a source (cont.)

- To find the steady-state probability distribution of Σ we can solve the equations

$$\boldsymbol{q}^{\mathsf{T}}\boldsymbol{P} = \boldsymbol{q}^{\mathsf{T}}, \qquad \mathbf{1}^{\mathsf{T}}\boldsymbol{q} = 1$$

- We obtain

$$\boldsymbol{q}^{\mathsf{T}} = (0.0259, 0.1036, 0.1036, 0.1658, 0.1036, 0.1658, 0.1658, 0.1658)$$

- Next, we calculate the conditional entropies $H(X|\Sigma = \sigma)$:

$$H(X|\Sigma = \{000, 001, 010, 100\}) = H_b(0.2) = 0.7219$$
$$H(X|\Sigma = \{011, 101, 110, 111\}) = H_b(0.5) = 1$$

Entropy rate of a source (cont.)

- Finally, we can calculate the entropy rate:

$$\begin{aligned}\bar{H} &= \sum_{\sigma\in\{000,001,010,100\}} P(\Sigma=\sigma)H_b(0.2) \\ &+ \sum_{\sigma\in\{011,101,110,111\}} P(\Sigma=\sigma)H_b(0.5) \\ &= (0.0259+0.1036+0.1036+0.1036)\times 0.7219 \\ &+ (0.1658+0.1658+0.1658+0.1658)\times 1 \\ &= 0.9063\end{aligned}$$

Laboratory: Computation of entropies

- Consider two discrete random variables X, Y with alphabets $\mathcal{X} = \mathcal{Y} = (1:10)$ and joint probability distribution

$$P(X = x, Y = y) = \frac{2x + 5y}{K}$$

Determine

- the normalization constant K;
- the entropies $H(X), H(Y), H(X, Y)$;
- the entropies $H(X|Y), H(Y|X)$;
- the mutual information $I(X;Y)$;
- the entropy $H(X + Y)$.

Laboratory: Computation of entropies (cont.)

- Consider $N = 10$ independent binary random variables $X_1, \ldots, X_N$ with $P(X_i = 0) = 0.2$ for $i = 1, \ldots, N$. Determine
  - the entropy $H(X_1, \ldots, X_N)$;
  - the entropy $H(X_1 + \cdots + X_N)$;

Laboratory: Test of entropy inequalities

- Consider two discrete random variables X, Y with alphabets $\mathcal{X} = \mathcal{Y} = (1:10)$ and joint probability distribution

$$P(X = x, Y = y) = \frac{x^2 + y}{K}$$

Determine the normalization constant K and verify the following inequalities:

- $H(X|Y) \leq H(X) \leq H(X,Y)$;
- $H(Y|X) \leq H(X) \leq H(X,Y)$;
- $H(X \times Y) \leq H(X,Y)$;
- $H(X + Y) \leq H(X,Y)$.

Laboratory: Test of entropy inequalities (cont.)

- Consider two discrete random variables X, Y with alphabets $\mathcal{X} = \mathcal{Y} = (1:10)$ and joint probability distribution

$$P(X = x, Y = y) = \begin{cases} \frac{x+y}{K} & (x, y) = 1 \\ 0 & (x, y) > 1 \end{cases}$$

where (x, y) is the maximum common divisor of x and y. Determine the normalization constant K and verify the following inequalities:

- $H(X|Y) \leq H(X) \leq H(X, Y)$;
- $H(Y|X) \leq H(X) \leq H(X, Y)$;
- $H(XY) \leq H(X, Y)$;
- $H(X + Y) \leq H(X, Y)$.

Laboratory: Computation of entropy rates

- Calculate the entropy rate of a Markov source with alphabet $\mathcal{X} = (1:10)$ characterized by

$$P(X_n = a | X_{n-1} = b) = \frac{a+b}{K_b}$$

after determining the constants K_b.

Laboratory: Computation of entropy rates (cont.)

- Calculate the entropy rate of a binary Markov source with memory $L = 4$ characterized by

$$P(X_n = a | X_{n-4:n-1} = \boldsymbol{b}) = \frac{a + b_1 + \cdots + b_4}{K_{\boldsymbol{b}}}$$

where $a = 0, 1$ and $\boldsymbol{b}$ is a binary vector, after determining the constants $K_{\boldsymbol{b}}$.

Laboratory: Entropy rate of a language

In this experiment we estimate the entropy rate of the English language by using Matlab.

- Select a long text sample written in English.
- Read the text into memory, split it into L-character sequences.
- Estimate the frequency of the sequences found (the choice of L depends on the language memory).
- Interpret the frequencies as probabilities and calculate the entropy $H(X_{1:L})$.
- The entropy rate is approximately

$$\bar{H}_L \triangleq \frac{H(X_{1:L})}{L}$$

Laboratory: Entropy rate of a language (cont.)

- Determine the compression ratio as follows:

$$\rho_L = \frac{N_s H(X_{1:L}) + N_D L N_c}{N_s L N_c}$$

where

- N_s is the number of L-character sequences read;
- $H(X_{1:L})$ is approximately the number of bit/encoded sequence with a Huffman code;
- N_D is the number of different L-character sequences found (dictionary size);
- N_c is the number of bits per character used (typically, if characters are bytes, $N_c = 8$).

Outline

Section Outline

3 Source coding
- Fixed-length encoding
- Fixed-to-variable length encoding
- Source code classification
- Kraft inequality
- McMillan inequality
- Shannon theorem for source coding
- Huffman codes
- Laboratory: Huffman codes

Fixed-length encoding

- The goal of source coding is the efficient transmission of data symbols emitted by a source by using the minimum average number of bits per source symbol.
- To achieve this goal, the source symbols must not be independent and equiprobable over their own alphabet because in that case source coding would not be effective.
- A key parameter for source coding is the entropy rate of the source, which, as we are going to see, represents a lower bound to the minimum average number of bits per symbol required by any source code.

Fixed-length encoding (cont.)

- A simple way to implement source coding is based on the following method.
  - Let the source symbols belong to an alphabet $\mathcal{X}$ with cardinality M, i.e., $|\mathcal{X}| = M$.
  - The minimum number of bits required to represent all the symbols in $\mathcal{X}$, say: $\xi_1, \xi_2, \ldots, \xi_M$, is the minimum integer m greater than or equal to $\log_2 M$, i.e., $m = \lceil \log_2 M \rceil$.
  - Then, the source symbols are replaced by binary vectors of constant length m appended to each other
- For example, let us consider the alphabet

$$\mathcal{X} = \{\text{'D'}, \text{'E'}, \text{'F'}, \text{'I'}, \text{'N'}, \text{'R'}\}$$

- In this case, $M = 6$ and $m = \lceil \log_2 6 \rceil = 3$.

Fixed-length encoding (cont.)

- Let us use the code represented in the following table:

Source symbol:	'D'	'E'	'F'	'I'	'N'	'R'
Code vector:	000	001	010	011	100	101

- If we want to encode the word 'FRIEND', the result is:

  010 101 011 001 100 000

- The number of bits per symbol is fixed and depends only on the cardinality of the source alphabet. No optimization is possible.
- This is called fixed-to-fixed encoding because a fixed number of source symbols is encoded by a fixed number of bits.
- In the case considered, part of the encoding potential is wasted because three bits would enable to encode a source alphabet of 8 symbols.

Fixed-length encoding (cont.)

- To improve efficiency we could encode symbol triplets.
- In fact, $\lceil \log_2(|\mathcal{X}^3|) \rceil = \lceil \log_2(6^3) \rceil = 8$, so that encoding three source symbols would require 8 bits per symbol triplet instead of 9 as in the original code.
- The resulting code would be described by a bigger table with $6^3 = 216$ entries associating a source symbol triplet and a bit vector of length 8.
- In other words, additional efficiency is traded off by more complexity, which is a general pattern.
- Besides grouping the source symbols, we can improve coding efficiency by exploiting the source statistics but this requires a different approach to encoding: variable-length encoding.

Fixed-to-variable length encoding

- A fixed-to-variable source code is characterized by
  - a source alphabet $\mathcal{X} = \{\xi_1, \ldots, \xi_M\}$;
  - an encoding function $\boldsymbol{c}$ mapping all symbols $x \in \mathcal{X}$ into bit vectors $\boldsymbol{c}(x)$ of length $\nu(x)$;
  - encoding a symbol sequence $x_{1:N}$ produces a bit sequence obtained by concatenating the bit vectors $\boldsymbol{c}(x_i)$ for $i = 1, \ldots, N$ and corresponds to an average number of bits per symbol required equal to

$$\bar{n}(x_{1:N}) = \frac{1}{N}\sum_{n=1}^{N}\nu(x_n) = \sum_{i=1}^{M}\frac{N_i}{N}\nu(\xi_i)$$

  where N_i counts how many times ξ_i is present in $x_{1:n}$.

Fixed-to-variable length encoding (cont.)

- We have seen that the average number of bits per symbol $\bar{n}(x_{1:N})$ depends on the source sequence through the frequencies $f_i \triangleq N_i/N$ of the alphabet symbols.
- We cannot minimize $\bar{n}(x_{1:N})$ over all possible source symbol sequences so that we minimize it over the ensemble of the symbol sequences according to their statistic distribution.
- Let us consider a stationary independent source.
- By the Law of Large Numbers, it turns out that the ensemble average of the number of bits per symbols is obtained by replacing the frequencis f_i by the probabilities of emission of the source symbols, $p_i \triangleq P(X = \xi_i)$ for $i = 1, \ldots, M$.

Fixed-to-variable length encoding (cont.)

- Then, we design the source code with the goal of minimizing the ensemble average number of bits per symbol:

$$\bar{n} \triangleq \sum_{i=1}^{M} p_i \nu(\xi_i). \tag{12}$$

- It is intuitive that source symbols with higher probabilities should be encoded by shorter bit vectors (or codewords).
- However, the code design cannot be reduced to an arbitrary choice of bit vectors minimizing $\bar{n}$ and some additional insight is required.
- Basically, in order to be useful, a source code must be invertible, i.e., the encoding must be followed by a decoding operation which reproduces exactly the source symbol sequence.

Source code classification

- Consider a source code $\mathcal{C}$ over an alphabet $\mathcal{X} = \{\xi_1, \ldots, \xi_M\}$ with an encoding function $\boldsymbol{c}$.
- **Definition.**
  A source code is nonsingular if $\boldsymbol{c}(\xi_i) \neq \boldsymbol{c}(\xi_j)$ for any $i \neq j$.
- The previous condition is not sufficient to uniquely decode an encoded sequence but only to decode individual source symbols.
- **Definition.**
  The code extension of length n of $\mathcal{C}$ is the set of all possible source sequences of length n and is denoted by $\mathcal{C}^n$.
- For example,

$$\begin{aligned} \mathcal{C}^1 &= \{\boldsymbol{c}(\xi_1), \ldots, \boldsymbol{c}(\xi_M)\} \\ \mathcal{C}^2 &= \{(\boldsymbol{c}(\xi_1), \boldsymbol{c}(\xi_1)), (\boldsymbol{c}(\xi_1), \boldsymbol{c}(\xi_2)), \ldots, (\boldsymbol{c}(\xi_M), \boldsymbol{c}(\xi_M))\} \end{aligned}$$

Source code classification (cont.)

- **Definition.**
  The complete extension of $\mathcal{C}$ is defined as

$$\mathcal{C}^* \triangleq \bigcup_{n=1}^{\infty} \mathcal{C}^n. \tag{13}$$

- In other words, $\mathcal{C}^*$ is the set of all possible encoded binary sequences of any length.

Source code classification (cont.)

- **Definition.**
  A source code $\mathcal{C}$ is uniquely decodable if its complete extension $\mathcal{C}^*$ is nonsingular.
- In fact, the condition is equivalent to excluding the occurrence of the following condition:

$$(\boldsymbol{c}(x_1), \ldots, \boldsymbol{c}(x_n)) = (\boldsymbol{c}(x'_1), \ldots, \boldsymbol{c}(x'_{n'}))$$

  for any two different source symbol sequences $x_{1:n}$ e $x'_{1:n'}$.
- The previous condition is required in order that a source code be useful and is not always verified, as illustrated by the following example.

Source code classification (cont.)

- Consider a source with ternary alphabet $\mathcal{X} = \{\xi_1, \xi_2, \xi_3\}$ and the code specified by

$$\boldsymbol{c}(\xi_1) = (0) \qquad \boldsymbol{c}(\xi_2) = (0,1) \qquad \boldsymbol{c}(\xi_3) = (1,0).$$

- We can see that

$$\boldsymbol{c}(\xi_1, \xi_3, \xi_2, \xi_1) = (0,1,0,0,1,0)$$
$$\boldsymbol{c}(\xi_2, \xi_1, \xi_1, \xi_3) = (0,1,0,0,1,0)$$

so that the code **is not** uniquely decodable.

Source code classification (cont.)

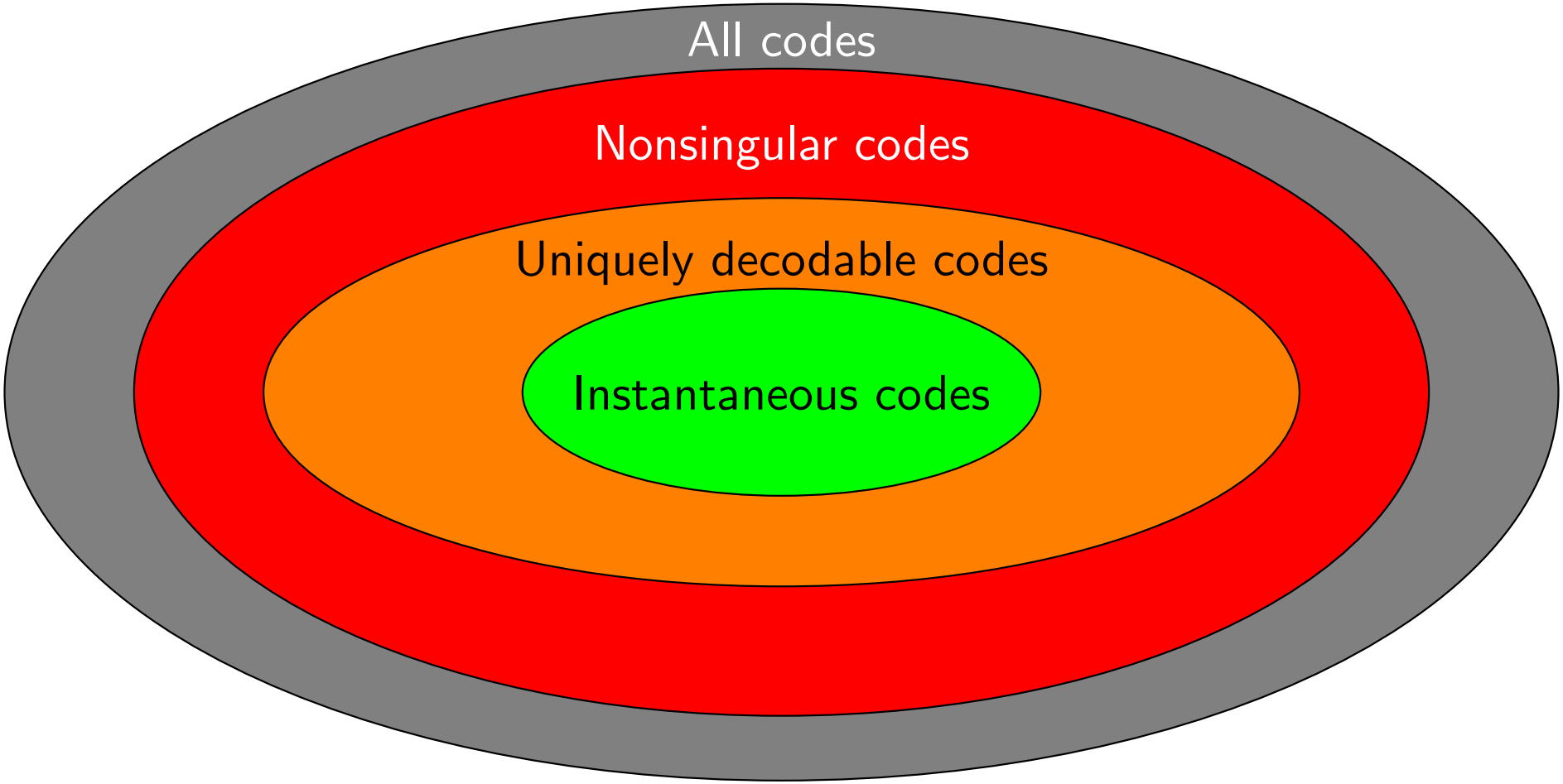
- An additional requirement for source codes, besides unique decodability, is that always be possible to decode a source symbol sequence as soon as the last encoded bit is available.
- **Definition.**
  A source code $\mathcal{C}$ is called instantaneous or prefix-free if no codeword is a prefix of another codeword.
- For example, $(0, 1, 0)$ is a prefix of $(0, 1, 0, 1, 1)$.
- The following table clarifies the different properties of source codes on a quaternary alphabet.

Source code classification (cont.)

$\mathcal{X}$	Singular	Nonsingular,but not uniquely decodable	Uniquely decodable but not instantaneous	Instantaneous
1	0	0	10	0
2	1	010	00	10
3	0	01	11	110
4	1	10	110	111
		$\boldsymbol{c}(1,4) = \boldsymbol{c}(2)$	11 prefix of 110	

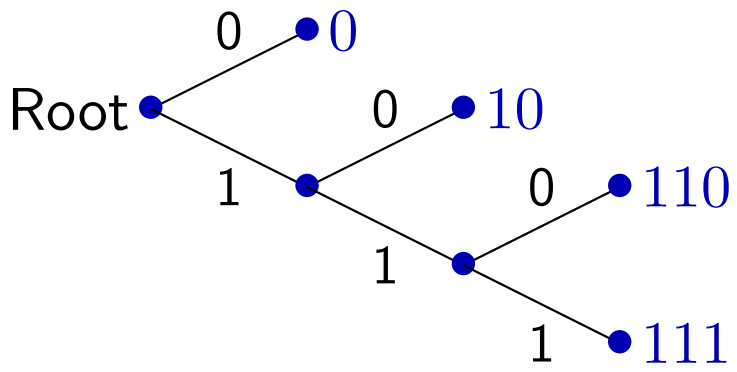
Source code classification (cont.)

Source code classification



Kraft inequality

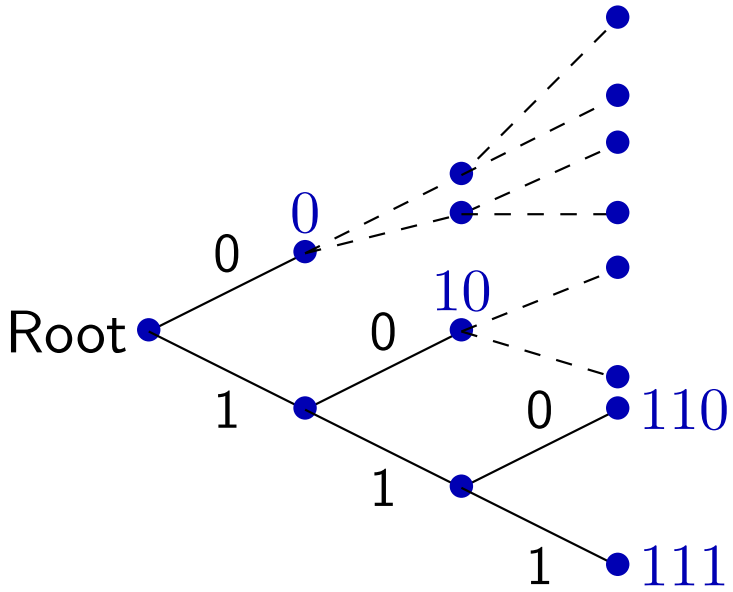
- From the previous arguments, our goal is to find a prefix-free source code with minimum $\bar{n}$.
- Prefix-free codes admit a binary tree representation:



- Unique decodability corresponds to decoding by traveling the tree from the root by choosing the branches according to the received bits and decoding a symbol whenever a leaf (terminal node) is reached.
- The number of branches from the root to any leaf is the corresponding codeword length.

Kraft inequality (cont.)

- The tree can be extended as follows:



- Now let $n_i \triangleq \nu(\xi_i), i = 1, \ldots, M$ and $n_{\max} \triangleq \max_{1 \le i \le n} n_i$.
- We can see that a codeword of length n_i (corresponding to a node at depth n_i generates $2^{n_{\max} - n_i}$ nodes at depth $n_{\max}$.

Kraft inequality (cont.)

- The prefix-free condition implies that the nodes at depth $n_{\max}$ generated by all codewords are all different.
- Therefore,

$$\sum_{i=1}^{M} 2^{n_{\max}-n_i} \leq 2^{n_{\max}}$$

  because the total number of nodes at depth $n_{\max}$ is $2^{n_{\max}}$.
- Dividing both sides of the above inequality by $2^{n_{\max}}$ we obtain Kraft inequality.
- **Kraft inequality.** A prefix-free code exists if and only if the following inequality is fulfilled:

$$\sum_{i=1}^{M} 2^{-n_i} \leq 1. \tag{14}$$

McMillan inequality

- Kraft inequality appears in a related result, holding for uniquely decodable codes.
- **McMillan inequality.** A uniquely decodable code satisfies (14):

$$\sum_{i=1}^{M} 2^{-n_i} \leq 1.$$

- **Proof.** The code extension $\mathcal{C}^k$ (encoding symbols from $\mathcal{X}^k$) is uniquely decodable since $\mathcal{C}$ is uniquely decodable.
- Its encoding function is

$$\boldsymbol{c}_k(x_{1:k}) = (\boldsymbol{c}(x_1), \ldots, \boldsymbol{c}(x_k))$$

McMillan inequality (cont.)

- Therefore, the codeword length is given by

$$\nu(x_{1:k}) = \nu(x_1) + \ldots + \nu(x_k)$$

- Then, consider the sum

$$\begin{aligned}
\sum_{x_{1:k} \in \mathcal{X}^k} 2^{-\nu(x_{1:k})} &= \sum_{x_1 \in \mathcal{X}, \ldots, x_k \in \mathcal{X}} 2^{-(\nu(x_1)+\ldots+\nu(x_k))} \\
&= \sum_{x_1 \in \mathcal{X}} 2^{-\nu(x_1)} \cdots \sum_{x_k \in \mathcal{X}} 2^{-\nu(x_k)} \\
&= \left[\sum_{x \in \mathcal{X}} 2^{-\nu(x)} \right]^k
\end{aligned}$$

McMillan inequality (cont.)

- Denoting by N_m the number of codewords from $\mathcal{C}^k$ of length m, we have

$$\sum_{x_{1:k}\in\mathcal{X}^k} 2^{-\nu(x_{1:k})} = \sum_{m=k}^{kn_{\max}} N_m 2^{-m}$$

  since the length of codewords from $\mathcal{C}$ is between 1 and $n_{\max}$ and then the length of codewords from $\mathcal{C}^k$ is between k and $kn_{\max}$.
- Since $\mathcal{C}^k$ is uniquely decodable, codewords of given length m must be distinct and hence their number cannot exceed the maximum number of binary vectors of length m, so that

$$N_m \leq 2^m.$$

McMillan inequality (cont.)

- As a result, we have

$$\left[\sum_{x\in\mathcal{X}} 2^{-\nu(x)}\right]^k \leq \sum_{m=k}^{kn_{\max}} 2^m 2^{-m} = k(n_{\max}-1)+1 \ ,$$

- It follows that

$$\sum_{x\in\mathcal{X}} 2^{-\nu(x)} \leq [k(n_{\max}-1)+1]^{1/k}$$

for every $k > 0$.

McMillan inequality (cont.)

- By letting $k \to \infty$ we obtain

$$\begin{aligned}\sum_{x\in\mathcal{X}} 2^{-\nu(x)} &\leq \lim_{k\to\infty} [k(n_{\max}-1)+1]^{1/k} \\ &= \lim_{k\to\infty} \exp[\ln(k(n_{\max}-1)+1)/k] \\ &= \exp[\lim_{k\to\infty} \ln(k(n_{\max}-1)+1)/k] \\ &= \exp[0] \\ &= 1\end{aligned}$$

  which is equivalent to (14).

- Therefore, from McMillan and Kraft inequalities, every uniquely decodable source code satisfies (14) and has an equivalent prefix-free code with the same codeword lengths and therefore the same $\bar{n}$.

McMillan inequality (cont.)

- Since prefix-free codes are instantaneous, there is no interest in searching uniquely decodable codes.

Shannon theorem for source coding

- Shannon's theorem provides an answer to the question if there exists a lower bound to the average number of bits per symbol required by a source code.
- **Shannon Theorem.**
  The average number of bits per symbol required by a uniquely decodable source code applied to a stationary independent source with entropy $H(X)$ satisfies the inequalities

$$H(X) \leq \bar{n} \leq H(X) + 1$$

Shannon theorem for source coding (cont.)

- **Proof.**
  The lower bound comes from the application of the logarithmic inequality and the Kraft/McMillan inequality (14) :

$$
\begin{aligned}
H(X) - \bar{n} &= \sum_{i=1}^{M} p_i \log_2 \frac{1}{p_i} - \sum_{i=1}^{M} p_i n_i \\
&= \sum_{i=1}^{M} p_i \log_2 \frac{2^{-n_i}}{p_i} \\
&\overset{\leq}{} \log_2 e \sum_{i=1}^{M} p_i \left(\frac{2^{-n_i}}{p_i} - 1 \right) \\
&= \log_2 e \left[\sum_{i=1}^{M} 2^{-n_i} - 1 \right] \\
&\overset{\leq}{} 0
\end{aligned}
$$

Shannon theorem for source coding (cont.)

- The upper bound can be obtained by setting

$$n_i = \lceil \log_2(1/p_i) \rceil < \log_2(1/p_i) + 1.$$

- Since $n_i = \lceil \log_2(1/p_i) \rceil \geq \log_2(1/p_i)$, $\sum_{i=1}^{M} 2^{-n_i} \leq \sum_{i=1}^{M} p_i = 1$ so that Kraft inequality is satisfied and the code is uniquely decodable.
- Moreover, from the upper bound we get

$$\bar{n} = \sum_{i=1}^{M} p_i n_i \leq \sum_{i=1}^{M} p_i \left(\log_2 \frac{1}{p_i} + 1 \right) = H(X) + 1$$

□

Shannon theorem for source coding (cont.)

- It is interesting to notice that the logarithmic inequality identity conditions imply that $\bar{n} = H(X)$ only if all the source symbol probabilities are negative integer powers of 2:

$$p_i = 2^{-n_i}, \quad i = 1, \ldots, M.$$

- Shannon theorem provides an operative meaning of entropy as the minimum number of bits per symbols required by a uniquely decodable source code.

Huffman codes

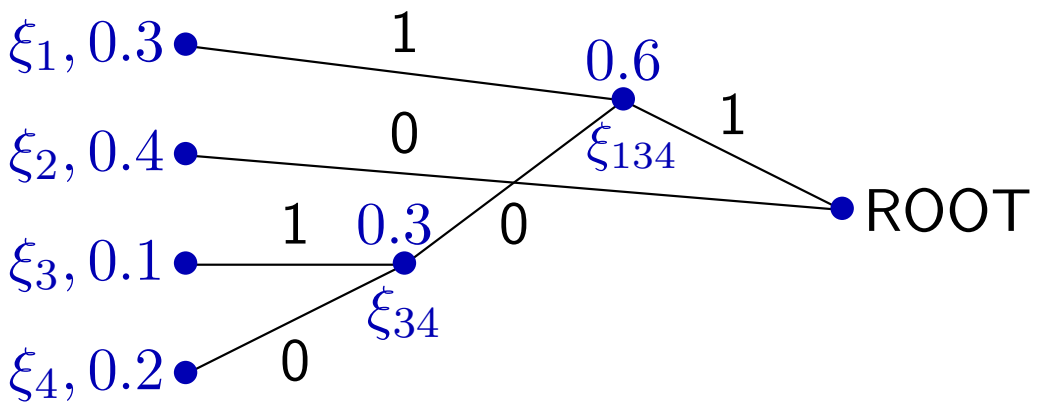
- Huffman codes are prefix-free source codes that minimize the average number of bits per symbol for a stationary independent source.
- For these reason they are commonly regarded as optimal source codes.
- Huffman codes are derived by construction of the code tree starting from the leaves according to the following algorithm.

Huffman codes (cont.)

- **Huffman algorithm.** Given a stationary independent source with alphabet $\mathcal{X} = \{\xi_1, \ldots, \xi_M\}$ and probabilities $p_i = P(X_n = \xi_i)$, execute the following iterative steps:
  1. Find the two lowest probabilities in the set $(p_1, \ldots, p_M)$, say p_{i_1}, p_{i_2}.
  2. Set the corresponding symbols ξ_{i_1}, ξ_{i_2} as leaves of the code tree.
  3. Connect the two leaves to an inner node and assign to this node a new symbol with probability $p_{i_1} + p_{i_2}$, meanwhile deleting the symbols ξ_{i_1}, ξ_{i_2}. As a result, the number of symbols to encode decreases from M to $M - 1$.
  4. If the updated source alphabet contains more than 1 symbol go to 1.
  5. Otherwise label the binary tree branches as 0 and 1 and find the codewords by reading the branch labels sequentially from the root (last created node) to the leaves.

Huffman codes (cont.)

Example. Construct a Huffman code for the source $\mathcal{X} = \{\xi_{1:4}\}$ with $p_1 = 0.3, p_2 = 0.4, p_3 = 0.1, p_4 = 0.2$.



Step 1: merge ξ_3, ξ_4 into ξ_{34} with probability $0.1 + 0.2 = 0.3$

Step 2: merge ξ_1, ξ_{34} into ξ_{134} with probability $0.3 + 0.3 = 0.6$

Step 2: merge ξ_2, ξ_{134} into ξ_{1234} with probability $0.4 + 0.6 = 1.0$

Resulting Huffman code:

$$\boldsymbol{c}(\xi_1) = (1,1), \boldsymbol{c}(\xi_2) = (0), \boldsymbol{c}(\xi_3) = (1,0,1), \boldsymbol{c}(\xi_4) = (1,0,0)$$

Huffman codes (cont.)

- Let us analyze the calculation of the average number of bits per symbol in this example in order to derive a general algorithm.
- We have

$$\begin{aligned}\bar{n} &= 1 \times 0.4 + 2 \times 0.3 + 3 \times (0.1 + 0.2) \\ &= \underbrace{1 \times 0.4 + 2 \times 0.3 + 2 \times (0.1 + 0.2)}_{(a)} + \underbrace{(0.1 + 0.2)}_{(b)}\end{aligned}$$

- The first term, (a), is the average number of bits per symbol of the source code corresponding to the symbols ξ_1, ξ_2, ξ_{34}.
- The second term, (b), is the probability of ξ_{34}, i.e., the sum of the probabilities of ξ_3 and ξ_4.

Huffman codes (cont.)

- It turn out that we can write a recurrence relationship giving the average number of bits per symbol of a Huffman code $\mathcal{C}$ corresponding to the alphabet $\mathcal{X}$ as the sum of
  - the average number of bits per symbol of a Huffman code $\mathcal{C}'$ corresponding to an alphabet $\mathcal{X}'$ obtained by merging the two symbols with least probabilities in $\mathcal{X}$ into one symbol with probability the sum of the probabilities of the merged symbols (in $\mathcal{X}$ and not in $\mathcal{X}'$)
  - plus the probability of this newly defined symbol.
- In our example, $\mathcal{X} = \{\xi_1, \xi_2, \xi_3, \xi_4\}$ and $\mathcal{X}' = \{\xi_1, \xi_2, \xi_{34}\}$.

Huffman codes (cont.)

- For the Huffman code considered we have

$$H(X) = H(0.1, 0.2, 0.3, 0.4) = 1.8464 \qquad \bar{n} = 1.9$$

- The code efficiency is defined as

$$\eta \triangleq \frac{H(X)}{\bar{n}} = \frac{1.8464}{1.9} = 97.2\%$$

Laboratory: Huffman codes

- Write a Matlab function to calculate the average number of bits per symbol of a Huffman code corresponding to a given probability distribution (vector `pv`).
- Write a Matlab function to build a Huffman code corresponding to a given probability distribution (vector `pv`).

Laboratory: Huffman codes (cont.)

- Evaluate the efficiency of the following data compression algorithm.
  - Fix a dictionary size, `Nentries`.
  - Fix a dictionary entry size, `Nbytes`.
  - Read an input file as a byte sequence forming entries of `Nbytes` bytes. Add the entries to the dictionary until it is full or the file is fully processed.
  - Evaluate the frequencies of the entries and construct a Huffman code based on these frequencies.
  - Evaluate the number of bits required to store the dictionary, the Huffman code, and the encoded content of the file.

Outline

Section Outline

4. Communication channels
   - Channel codes
   - Shannon Theorem
   - Channel capacity
   - Blahut-Arimoto algorithm
   - Data-Processing Inequality
   - Laboratory: Blahut-Arimoto algorithm
   - Continuous communication channel
   - Additive Gaussian channel
   - Weighted Water-Filling
   - Laboratory: Continuous channel capacity

Communication channels

- Communication channels model the transmission of information data from point to point in the space.
- The typical communication channel model is given by the following block diagram:

- X and Y may represent, for example, m-bit vectors encoded as integer numbers in $\{1 : 2^m\}$.
- X and Y are related by the conditional probability distribution $P(Y = y|X = x)$.
- The channel can be used sequentially so that at discrete time n (an integer number) the channel input is X_n and the output is Y_n.

Communication channels (cont.)

- An ideal channel reproduces the input at the output: $Y = X$.
- A real channel output is not always equal to the input.
- If it is not, we have a channel error.
- In order to estimate the channel input X from the channel output $Y = y$, known at the receiver, the Maximum A Posteriori (MAP) probability rule is usually implemented:

$$\widehat{X} = \arg\max_{x \in \mathcal{X}} P(X = x | Y = y).$$

- The a posteriori probability is obtained from the Bayes rule:

$$P(X = x | Y = y) = \frac{P(X = x) P(Y = y | X = x)}{\sum_{\check{x} \in \mathcal{X}} P(X = \check{x}) P(Y = y | X = \check{x})}$$

Communication channels (cont.)

- In the previous formula we need the probabilities $P(X = x)$ and $P(Y = y|X = x)$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.
- The probabilities $P(X = x)$ are called a priori probabilities and characterize the transmitting source.
- The conditional probabilities $P(Y = y|X = x)$ characterize the channel.
- In most cases, the a priori probabilities are constant, so that $P(X = x) = 1/|\mathcal{X}|$ (equiprobable distribution).
- In the case of equiprobable a priori distribution, the Bayes rule simplifies as follows:

$$P(X = x|Y = y) = \frac{P(Y = y|X = x)}{\sum_{\check{x}\in\mathcal{X}} P(Y = y|X = \check{x})}$$

Communication channels (cont.)

- The receiver operation is called MAP decision or detection and minimizes the error probability when decisions are taken on a symbol-by-symbol basis.
- Channel codes are used to minimize the error probability.
- They are based on the transmission of long symbol sequences chosen with appropriate design criteria.

Channel codes

- An $[M, n]$ channel code over $\mathcal{X}^n$, $\mathcal{C}$, is an invertible mapping $\mathcal{W} \to \mathcal{X}^n$ between a set of messages $\mathcal{W} = \{1, \ldots, M\}$ and the set of different n-tuples $(x_1, \ldots, x_n) \in \mathcal{X}^n$, which are called code words.
- The set of all possible code words is called the codebook.
- Transmitting one message from a set of M possible messages is equivalent to transmitting $\log_2 M$ bits.
- For example, if $M = 4$, we can make the following associations:

$$1 \leftrightarrow (0,0) \qquad 2 \leftrightarrow (0,1) \qquad 3 \leftrightarrow (1,0) \qquad 4 \leftrightarrow (1,1)$$

Channel codes (cont.)

- We define the code rate as:

$$R = \frac{\log_2 M}{n} \frac{\text{information bits}}{\text{channel symbols}}. \tag{15}$$

- Since M cannot be larger than the number of all possible words from $\mathcal{X}^n$ (i.e., $|\mathcal{X}|^n$), we have the inequality

$$R \leq \frac{\log_2(|\mathcal{X}|^n)}{n} = \log_2 |\mathcal{X}|.$$

- A binary channel code with $M = 2^k$ and $\mathcal{X} = \{0, 1\}$ is called an (n, k) code.
- Its rate is $R = \log_2(2^k)/n = k/n \leq \log_2 |\mathcal{X}| = 1$.

Channel codes (cont.)

- Examples of binary codes.
  - Repetition code $(n, 1)$: $\{(0 \ldots 0), (1 \ldots 1)$ with rate $R = 1/n$.
  - Single parity-check code $(n, n-1)$. The code words are obtained by concatenating a parity bit to the first $(n-1)$ bits. For example, the (3,2) code has the following words: $\{(000), (011), (101), (110)\}$. The rate is $R = (n-1)/n$.
  - A code can have no structure at all. For example, this is a $[4, 5]$ binary code with rate $R = 2/5$:

$$\boldsymbol{c}_1 = (10000),\ \boldsymbol{c}_2 = (01010),\ \boldsymbol{c}_3 = (10110),\ \boldsymbol{c}_4 = (11001).$$

Shannon Theorem

Shannon Theorem for communication channels

- Consider a communication channel with conditional probability distribution $p_{Y|X}(y|x)$.
- Assume that sequential channel transmission is conditionally independent:

$$\begin{aligned} P(Y_1 = y_1, \ldots, Y_n = y_n \mid X_1 = x_1, \ldots, X_n = x_n) \\ = \prod_{i=1}^{n} p_{Y|X}(y_i|x_i) \end{aligned}$$

for every $n \geq 1$.

Shannon Theorem (cont.)

- Shannon Theorem proves that,
  - given a target code rate R smaller than the channel capacity

$$C \triangleq \max_{p_X(x)} I(X;Y), \tag{16}$$

  - there is a code sequence $\mathcal{C}_n = \{\boldsymbol{x}_n(m) \in \mathcal{X}^n, m \in \{1 : M_n\}\}$ with code rates

$$R_n = \frac{\log_2 M_n}{n} \to R$$

    as $n \to \infty$, such that the maximum error probability

$$\left\{ P_{n,\max}(e) \triangleq \max_{1 \le m \le M_n} P\Big\{ \mathcal{D}_n(\boldsymbol{y}) \neq m \mid \boldsymbol{x}_n(m) \text{ transmitted} \Big\} \right\} \to 0$$

    as $n \to \infty$ for some decoding function $\mathcal{D}_n : \mathcal{Y}^n \mapsto \{1 : M_n\}$.

Shannon Theorem (cont.)

Simplified statement and consequences:

- According to Shannon Theorem, by using very long channel codes, we can transmit reliably (i.e., with vanishing error probability) up to a code rate equal to the channel capacity C given in (16).
- One codeword of n channel symbols carries $\approx nC$ information bits.
- If T is the time required to transmit one channel symbol, the maximum information bit rate is

$$\frac{\text{N. of information bits/codeword}}{\text{Time to transmit one codeword}} = \frac{nC}{nT} = \frac{C}{T}\ \frac{\text{bit}}{\text{s}}$$

Shannon Theorem (cont.)

- Shannon Theorem has a converse, too.
- The converse says that if the error probability of a code sequence $\mathcal{C}_n$ with limit rate R tends to zero as $n \to \infty$, then it must be $R \leq C$.
- Equivalently, if $R > C$, there exists no code sequence such that the error probability tends to zero as $n \to \infty$.
- In other words, if $R > C$, it is impossible to transmit reliably.
- Therefore, the capacity is a hard limit in the sense that the achievable limiting error probability passes from 0 to 1 when the rate R crosses the capacity limit C.

Shannon Theorem (cont.)

- A question remains: How large must be the code length n in order that the error probability be smaller than ε?
- Recent studies by Verdù and Polyanskiy have shown that the Shannon Theorem approximation $\log_2 M_n \approx nC$ as $n \to \infty$ can be refined as

$$\log_2 M_n(\varepsilon) \approx nC - \sqrt{nV}Q^{-1}(\varepsilon) \text{ as } n \to \infty$$

where

- $M_n(\varepsilon)$ is he maximal cardinality of a codebook of length n which can be decoded with block error probability $\leq \varepsilon$;
- V is defined as the channel dispersion;
- $Q^{-1}(\cdot)$ is the inverse Q-function, which is defined as

$$Q(x) \triangleq \int_x^\infty \mathrm{e}^{-u^2/2} \frac{du}{\sqrt{2\pi}}$$

Shannon Theorem (cont.)

- Therefore, if we want to attain a fraction $\eta = \log_2 M_n(\varepsilon)/C$ of the channel capacity, then we have to solve the equation

$$\eta \approx 1 - \frac{1}{C}\sqrt{\frac{V}{n}}Q^{-1}(\varepsilon)$$

so that

$$n \approx \left(\frac{Q^{-1}(\varepsilon)}{1-\eta}\right)^2 \frac{V}{C^2}$$

Channel capacity

- Shannon Theorem outlines the role of channel capacity as the ultimate information bit rate which can be achieved over a channel with vanishing error probability.
- The channel capacity C depends on $M \times N$ channel matrix

$$\boldsymbol{P} \triangleq \Big(P(Y = y_j | X = x_i)\Big)_{i,j=1}^{M,N}, \qquad M = |\mathcal{X}|, N = |\mathcal{Y}|.$$

- It can be calculated analytically only in a limited number of cases, depending on the characteristics of $\boldsymbol{P}$:
  - if the rows and columns of $\boldsymbol{P}$ are permutations of each other;
  - in some cases when only the rows of $\boldsymbol{P}$ are permutations of each other;
  - in the case $M = N = 2$.
- In all other cases one can resort to numerical approximation by the Blahut-Arimoto algorithm.

Channel capacity (cont.)

- **Rows of $\boldsymbol{P}$ permutations of each other.**
  In this case,

$$\begin{aligned} H(Y|X) &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{XY}(x,y)\log_2 p_{Y|X}(y|x) \\ &= -\sum_{x\in\mathcal{X}} p_X(x)\sum_{y\in\mathcal{Y}} p_{Y|X}(y|x)\log_2 p_{Y|X}(y|x) \\ &= \sum_{x\in\mathcal{X}} p_X(x) H(Y|X=x) \end{aligned}$$

  where $H(Y|X=x) \triangleq -\sum_{y\in\mathcal{Y}} p_{Y|X}(y|x)\log_2 p_{Y|X}(y|x)$.

Channel capacity (cont.)

- The entropy $H(Y|X=x)$, which is the entropy function $H(\boldsymbol{\pi}_x)$, i.e., of the row of $\boldsymbol{P}$ corresponding to $X=x$.
- If the rows of $\boldsymbol{P}$ are permutations of each other, $H(Y|X=x)$ is constant for every row, say $H(\boldsymbol{\pi})$, where $\boldsymbol{\pi}$ is any of the $\boldsymbol{\pi}_x$, for example the first row of $\boldsymbol{P}$.
- Then,

$$H(Y|X) = \sum_{x\in\mathcal{X}} p_X(x) H(\boldsymbol{\pi}) = H(\boldsymbol{\pi}).$$

Channel capacity (cont.)

- As a result, since $I(X;Y) = H(Y) - H(Y|X)$,

$$C = \left\{ \max_{p_X(x)} H(Y) \right\} - H(\pi)$$

since the probability distribution of X affects $H(Y)$ and not $H(Y|X)$, in this case.

Channel capacity (cont.)

- **Rows and columns of $\boldsymbol{P}$ permutations of each other (strictly symmetric channel).**

  In this case, we can maximize $H(Y)$ by showing that the equiprobable input distribution gives an equiprobable output distribution, maximizing $H(Y)$.
- Since the columns of $\boldsymbol{P}$ are permutations of each other, $\sum_{x\in\mathcal{X}} p_{Y|X}(y|x)$ doesn't depend on the column index $y \in \mathcal{Y}$.
- Hence, if $p_X(x) = \frac{1}{M}$,

$$p_Y(y) = \sum_{x\in\mathcal{X}} p_X(x)p_{Y|X}(y|x) = \frac{1}{M}\sum_{x\in\mathcal{X}} p_{Y|X}(y|x)$$

  doesn't depend on $y \in \mathcal{Y}$ so that Y is equiprobable.
- $H(Y)$ reaches the maximum when both X and Y are equiprobable, and the maximum is $\log_2 |\mathcal{Y}|$.

Channel capacity (cont.)

- Thus, the capacity is

$$C = \boxed{\log_2 |\mathcal{Y}| - H(\boldsymbol{\pi})}$$

where $\boldsymbol{\pi}$ is any row of $\boldsymbol{P}$.

Channel capacity (cont.)

- It is also instructive to consider the matrix relationship between the input and the output probability distribution of a discrete channel through the channel matrix.
- Let the row-vectors $\boldsymbol{p}_x$ and $\boldsymbol{p}_y$ denote the input and output probability distribution, so that

$$P(X = x_i) = (\boldsymbol{p}_x)_i, \qquad P(Y = y_j) = (\boldsymbol{p}_y)_j$$

- Since

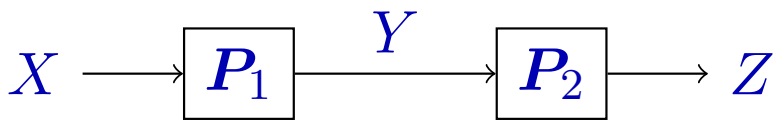
$$P(Y = y_j) = \sum_{i=1}^{|\mathcal{X}|} P(X = x_i) P(Y = y_j | X = x_i),$$

we have the following matrix equation:

$$\boldsymbol{p}_y = \boldsymbol{p}_x \boldsymbol{P}$$

Channel capacity (cont.)

- If we have two channels in cascade with matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ such that



  Then,

$$\boldsymbol{p}_z = \boldsymbol{p}_y \boldsymbol{P}_2 = (\boldsymbol{p}_x \boldsymbol{P}_1)\boldsymbol{P}_2 = \boldsymbol{p}_x(\boldsymbol{P}_1 \boldsymbol{P}_2)$$

- In other words, the channel matrix corresponding to the cascade of two channels with matrices $\boldsymbol{P}_1, \boldsymbol{P}_2$ is just $\boldsymbol{P}_1 \boldsymbol{P}_2$.
- This is called a Markov chain, $X \to Y \to Z$, because Z depends directly only on Y and not on X, like in the case of a probabilistic Markov chain where the state distribution at time n depends only on the state distribution at time $n-1$.

Channel capacity (cont.)

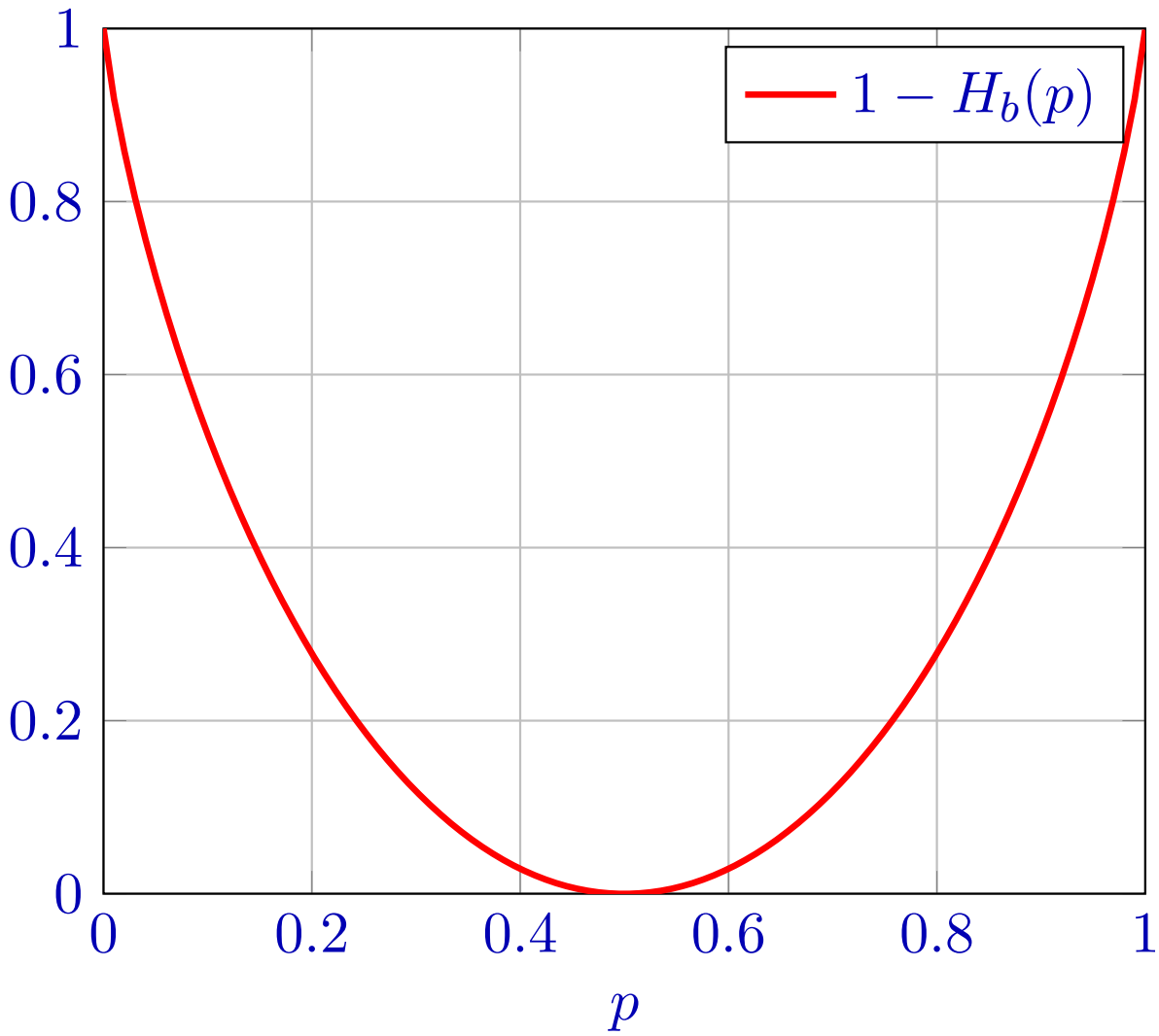
- Binary Symmetric Channel (BSC).
  The channel matrix is

$$\boldsymbol{P} = \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix}$$

  with $\mathcal{X} = \mathcal{Y} = \{0, 1\}$.
- The parameter p is called error probability of the channel.
- This matrix satisfies the strict symmetry condition stated above.
- Then, the channel capacity is

$$C = \log_2 |\mathcal{Y}| - H(1-p, p) = \boxed{1 - H_b(p)}. \tag{17}$$

Channel capacity (cont.)



Channel capacity (cont.)

- **Special case: Binary input symmetric output channel**

$$\boldsymbol{P} = \begin{pmatrix} p_1 & p_2 & \cdots & p_{N-1} & p_N \\ p_N & p_{N-1} & \cdots & p_2 & p_1 \end{pmatrix}$$

- We assume $p_X(\xi_1) = \alpha$ and $p_X(\xi_2) = \bar{\alpha} \triangleq 1 - \alpha$.
- We can see that $p_Y(\eta_j) = \alpha p_j + \bar{\alpha} p_{N+1-j}$.
- Then, applying the entropy inequality (2),

$$\begin{aligned} H(Y) &= H(\alpha p_1 + \bar{\alpha} p_N, \alpha p_2 + \bar{\alpha} p_{N-1}, \ldots, \alpha p_{N-1} + \bar{\alpha} p_2, \alpha p_N + \bar{\alpha} p_1) \\ &= H(\alpha p_1 + \bar{\alpha} p_N, \alpha p_N + \bar{\alpha} p_1, \alpha p_2 + \bar{\alpha} p_{N-1}, \alpha p_{N-1} + \bar{\alpha} p_2, \ldots) \\ &\leq H\left(\frac{p_1 + p_N}{2}, \frac{p_1 + p_N}{2}, \frac{p_2 + p_{N-1}}{2}, \frac{p_2 + p_{N-1}}{2}, \ldots\right) \end{aligned}$$

Channel capacity (cont.)

- The upper bound is attained when $\alpha = \frac{1}{2}$ and is therefore the maximum $H(Y)$.
- The capacity is

$$C = H\left(\frac{p_1 + p_N}{2}, \frac{p_1 + p_N}{2}, \frac{p_2 + p_{N-1}}{2}, \frac{p_2 + p_{N-1}}{2}, \dots\right) - H(p_1, \dots, p_N)$$

Channel capacity (cont.)

- For example (Binary Erasure Channel, BEC), if

$$\boldsymbol{P} = \begin{pmatrix} 1-p & p & 0 \\ 0 & p & 1-p \end{pmatrix},$$

we have

$$\begin{aligned} C &= H\left(\frac{1-p}{2}, \frac{1-p}{2}, p\right) - H(1-p, p) \\ &= 2 \times \frac{1-p}{2} \log_2 \frac{2}{1-p} + p \log_2 \frac{1}{p} \\ &\quad - (1-p) \log_2 \frac{1}{1-p} - p \log_2 \frac{1}{p} \\ &= 1-p \end{aligned}$$

Channel capacity (cont.)

- **Another example.**

$$\boldsymbol{P} = \begin{pmatrix} a & b & c & d \\ b & c & a & d \\ c & a & b & d \end{pmatrix}$$

- If $p_i = P(X = \xi_i), i = 1, 2, 3,$

$$\begin{aligned} H(Y) &= H(p_1 a + p_2 b + p_3 c, p_1 b + p_2 c + p_3 a, p_1 c + p_2 a + p_3 b, d) \\ &\leq H\left(\frac{a+b+c}{3}, \frac{a+b+c}{3}, \frac{a+b+c}{3}, d\right) \end{aligned}$$

and the maximum is attained for $p_1 = p_2 = p_3 = \frac{1}{3}$. Then,

$$C = H\left(\frac{a+b+c}{3}, \frac{a+b+c}{3}, \frac{a+b+c}{3}, d\right) - H(a, b, c, d)$$

Channel capacity (cont.)

- **Binary Asymmetric Channel (BAC).**

$$\boldsymbol{P} = \begin{pmatrix} 1-p & p \\ q & 1-q \end{pmatrix}$$

- Here, p, q are the error probabilities corresponding to the transmission of ξ_1, ξ_2, respectively.
- Let $\alpha \triangleq P(X = \xi_1)$, $\bar{\alpha} \triangleq 1 - \alpha$, $\bar{p} \triangleq 1 - p$, and $\bar{q} \triangleq 1 - q$.
- We express the mutual information as a function of α:

$$\begin{aligned} \varphi(\alpha) &\triangleq I(X;Y) \\ &= H(Y) - H(Y|X) \\ &= H_b(\alpha\bar{p} + \bar{\alpha}q) - \alpha H_b(p) - \bar{\alpha}H_b(q) \end{aligned}$$

Channel capacity (cont.)

- Then, we calculate the derivative of $H_b(p)$:

$$\begin{aligned} H_b'(p) &= \frac{1}{\ln 2}\frac{d}{dp}\Big(-p\ln p - (1-p)\ln(1-p)\Big) \\ &= \frac{1}{\ln 2}\left(-\ln p - \frac{p}{p} + \ln(1-p) - \frac{1-p}{1-p}(-1)\right) \\ &= \log_2 \frac{1-p}{p} \Longrightarrow p = \frac{1}{1+2^{H_b'(p)}} \end{aligned}$$

- Then we solve $\varphi'(\alpha) = 0$ to find the maximum:

$$\varphi'(\alpha) = H_b'(\alpha\bar{p} + \bar{\alpha}q)(\bar{p} - q) - H_b(p) + H_b(q) = 0$$

- The solution, for $\bar{p} \neq q$, with $\beta \triangleq 2^{(H_b(p)-H_b(q))/(\bar{p}-q)}$, is

$$\alpha\bar{p} + \bar{\alpha}q = \frac{1}{1+\beta}$$

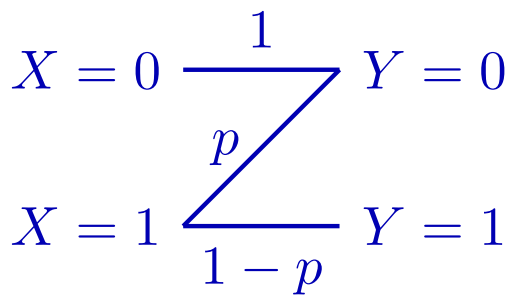
Channel capacity (cont.)

- Hence,
$$\alpha = \frac{\frac{1}{1+\beta} - q}{\bar{p} - q}, \qquad \bar{\alpha} = \frac{\bar{p} - \frac{1}{1+\beta}}{\bar{p} - q}$$
- Then,
$$C = H_b\left(\frac{1}{1+\beta}\right) - \frac{\frac{1}{1+\beta} - q}{\bar{p} - q} H_b(p) - \frac{\bar{p} - \frac{1}{1+\beta}}{\bar{p} - q} H_b(q)$$
$$= H_b\left(\frac{1}{1+\beta}\right) - \frac{1}{1+\beta} \boxed{\frac{H_b(p) - H_b(q)}{\bar{p} - q}} + \frac{qH_b(p) - \bar{p}H_b(q)}{\bar{p} - q}$$
$$= \frac{\log_2(1+\beta)}{1+\beta} + \frac{\beta}{1+\beta} \log_2 \frac{1+\beta}{\beta} - \boxed{\frac{\log_2 \beta}{1+\beta}} + \frac{qH_b(p) - \bar{p}H_b(q)}{\bar{p} - q}$$
$$= \log_2(1+\beta^{-1}) + \frac{qH_b(p) - \bar{p}H_b(q)}{\bar{p} - q}$$
$$= \boxed{\log_2\left(1 + 2^{(H_b(q) - H_b(p))/(\bar{p} - q)}\right) + \frac{qH_b(p) - \bar{p}H_b(q)}{\bar{p} - q}}$$

Channel capacity (cont.)

- Z Channel.
  The probability matrix is

$$\boldsymbol{P} = \begin{pmatrix} 1 & 0 \\ p & 1-p \end{pmatrix} \quad \Longrightarrow \quad$$

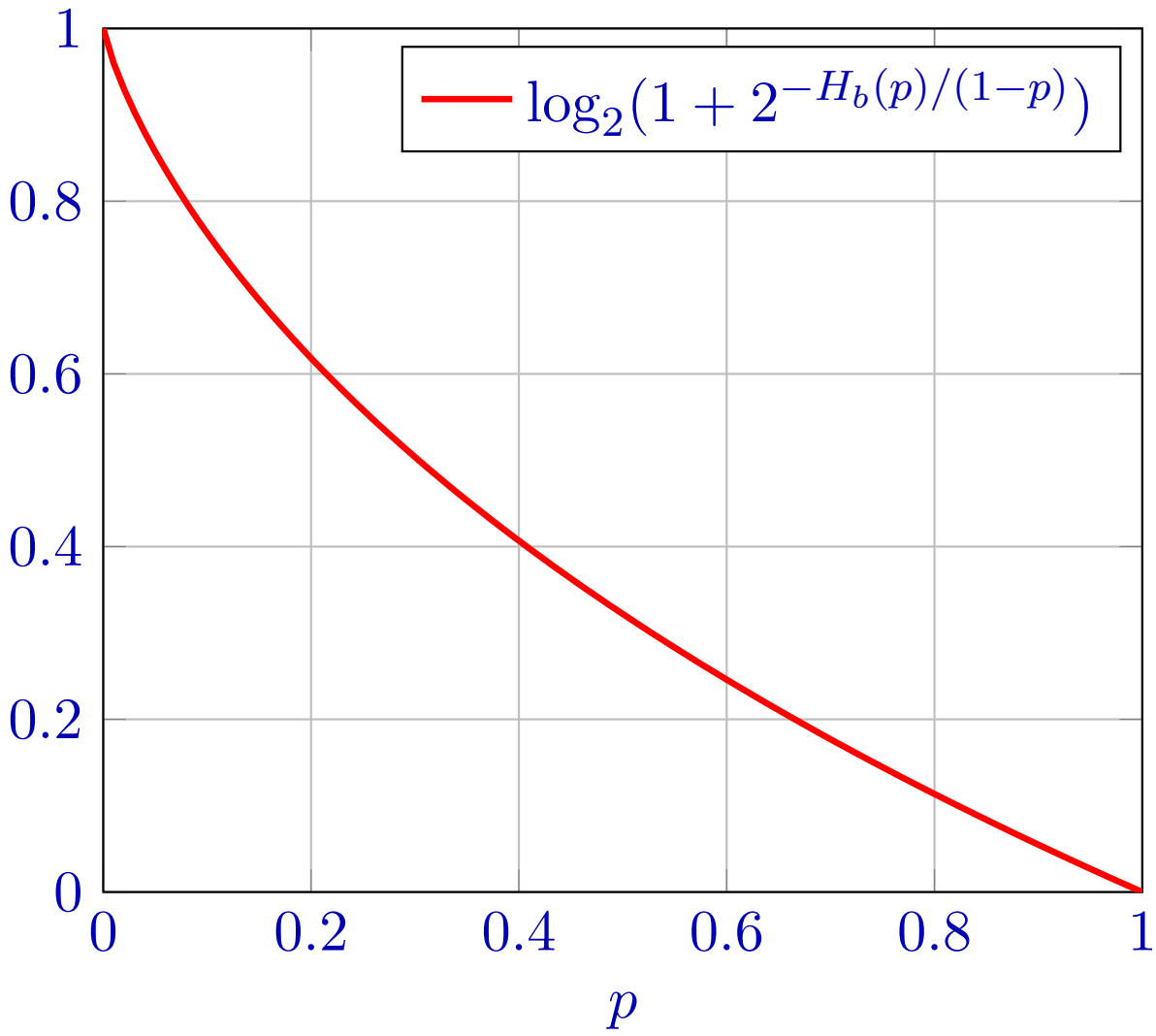


Channel capacity (cont.)

- We can apply the result obtained for the BAC with $\bar{p} \to 1, q \to p$:

$$
\begin{aligned}
C &= \log_2\left(1+2^{(H_b(q)-H_b(p))/(\bar{p}-q)}\right)+\frac{qH_b(p)-\bar{p}H_b(q)}{\bar{p}-q}\\
&= \log_2\left(1+2^{(H_b(p)-H_b(0))/(1-p)}\right)+\frac{pH_b(0)-1\cdot H_b(p)}{1-p}\\
&= \log_2\left(1+2^{H_b(p)/(1-p)}\right)-\frac{H_b(p)}{1-p}\\
&= \boxed{\log_2\left(1+2^{-H_b(p)/(1-p)}\right)}
\end{aligned}
$$

Channel capacity (cont.)



Channel capacity (cont.)

- The previous examples show that it is very difficult to calculate the capacity by analytic methods.
- In some cases, it is just impossible.
- However, it is always possible to calculate numerically the capacity of a discrete channel by resorting to the Blahut-Arimoto algorithhm.

Blahut-Arimoto algorithm

- The Blahut-Arimoto algorithm is used to calculate the channel capacity numerically and not analytically.
- The algorithm is iterative and based on a double maximization stemming from the following result.
- **Theorem.**
  The capacity of a discrete channel can be obtained as a double maximization:

$$C = \max_{p_X(x)} \max_{q(x|y)} \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{XY}(x,y) \log_2 \frac{q(x|y)}{p_X(x)} \tag{18}$$

  where $q(x|y)$ is an arbitrary probability distribution satisfying $\sum_{x\in\mathcal{X}} q(x|y) = 1$.

Blahut-Arimoto algorithm (cont.)

- **Proof.**
  Following the definition of channel capacity in eq. (16), it is sufficient to prove the inner maximization result:

$$I(X;Y) = \max_{q(x|y)} \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{XY}(x,y) \log_2 \frac{q(x|y)}{p_X(x)}.$$

- Since $I(X;Y) = H(X) - H(X|Y)$, it is sufficient to prove that the maximum is attained when $q(x|y) = p_{X|Y}(x|y)$.
- To this purpose we consider the difference

$$\sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{XY}(x,y) \log_2 \frac{q(x|y)}{p_X(x)} - I(X;Y)$$

Blahut-Arimoto algorithm (cont.)

- We have

$$\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{XY}(x,y)\log_2\frac{q(x|y)}{p_X(x)} - I(X;Y)$$

$$= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{XY}(x,y)\log_2\frac{q(x|y)}{p_X(x)} - \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_{XY}(x,y)\log_2\frac{p_{X|Y}(x|y)}{p_X(x)}$$

$$= \log_2 e \sum_{y\in\mathcal{Y}} p_Y(y) \sum_{x\in\mathcal{X}} p_{X|Y}(x|y)\ln\frac{q(x|y)}{p_{X|Y}(x|y)}$$

$$\leqslant \log_2 e \sum_{y\in\mathcal{Y}} p_Y(y) \sum_{x\in\mathcal{X}} p_{X|Y}(x|y)\left(\frac{q(x|y)}{p_{X|Y}(x|y)} - 1\right)$$

$$= \log_2 e \sum_{y\in\mathcal{Y}} p_Y(y) \sum_{x\in\mathcal{X}} \Big(q(x|y) - p_{X|Y}(x|y)\Big) = 0$$

Blahut-Arimoto algorithm (cont.)

- Therefore,

$$I(X;Y) = \max_{q(x|y)} \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} p_{XY}(x,y) \log_2 \frac{q(x|y)}{p_X(x)}$$

and the maximum is attained when

$$q(x|y) = p_{X|Y}(x|y)$$

for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. □

Data-Processing Inequality

- Consider the cascade of two channels represented by $X \to Y \to Z$ forming a Markov chain, i.e., such that

$$p_{Z|XY}(z|x,y) = p_{Z|Y}(z|y).$$

- The data-processing inequality is as follows:

$$I(X;Z) \leq I(X;Y). \tag{19}$$

- We shall use the conditional mutual information

$$\begin{aligned} I(X;Y|Z) &= H(X|Z) + H(Y|Z) - H(X,Y|Z) \\ &= H(X|Z) - H(X|Y,Z) \\ &= H(Y|Z) - H(Y|X,Z). \end{aligned} \tag{20}$$

Data-Processing Inequality (cont.)

- Proof of the Data-Processing Inequality.
- We first notice from the chain rule of mutual information that

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

- In a similar way, we can see that $I(X;Y,Z) = I(X;Z) + I(X;Y|Z)$.
- Thus, we have the identity

$$I(X;Y) + I(X;Z|Y) = I(X;Z) + I(X;Y|Z)$$

Data-Processing Inequality (cont.)

- Next we show that X and Z are conditionally independent given Y, i.e.,

$$p_{XZ|Y}(x,z|y) = p_{X|Y}(x|y)p_{Z|Y}(z|y).$$

- This is proved from the Markov property $p_{Z|XY}(z|x,y) = p_{Z|Y}(z|y)$:

$$\begin{aligned} p_{XZ|Y}(x,z|y) &= \frac{p_{XZY}(x,z,y)}{p_Y(y)} \\ &= \frac{p_{XZY}(x,z,y)}{p_{XY}(x,y)} \frac{p_{XY}(x,y)}{p_Y(y)} \\ &= p_{Z|XY}(z|x,y)p_{X|Y}(x|y) \\ &= p_{Z|Y}(z|y)p_{X|Y}(x|y) \end{aligned}$$

Data-Processing Inequality (cont.)

- The conditional independence property implies that

$$\begin{aligned} I(X;Z|Y) &= H(X|Y) + H(Z|Y) - H(X,Z|Y) \\ &= \mathbb{E}\left[\log_2\left(\frac{p_{XZ|Y}(X,Z|Y)}{p_{X|Y}(X|Y)p_{Z|Y}(Z|Y)}\right)\right] \\ &= \mathbb{E}[\log_2 1] \\ &= 0. \end{aligned}$$

- The previous result, $I(X;Z|Y) = 0$, implies that

$$I(X;Y) + 0 = I(X;Z) + I(X;Y|Z)$$

- Since $I(X;Y|Z) \geq 0$, we get

$$I(X;Z) \leq I(X;Y).$$

Data-Processing Inequality (cont.)

- In a similar way as before we have:

$$\begin{aligned} I(Z;X,Y) &= I(Z;Y) + I(Z;X|Y) \\ &= I(Z;X) + I(Z;Y|X) \end{aligned}$$

- Since $I(X;Z|Y) = I(Z;X|Y) = 0$, we get

$$I(Z;Y) + 0 = I(Z;X) + I(Z;Y|X).$$

- Since $I(Z;Y|X) \geq 0$, we get

$$I(X;Z) \leq I(Y;Z).$$

Data-Processing Inequality (cont.)

- The data processing inequality was originally derived to show that no type of data processing can increase the mutual information in a communication channel.
- Let the communication channel be $X \to Y$ with mutual information $I(X;Y)$.
- The channel output can be subject to a data-processing operation that maps every possible output Y to another output Z according to a deterministic data-processing function $Z = f(Y)$.
- This is a special case of Markov chain where the channel $Y \to Z$ is deterministic but the data-processing inequality still applies, as a special case.

Data-Processing Inequality (cont.)

- Therefore, since $I(X;Z) \leq I(X;Y)$ we conclude that there is no way to increase the amount of information carried through a channel by using data processing.
- Obviously, a possible increase in the mutual information would lead to a higher achievable rate, which is impossible to attain.
- This property confirms the key role of the mutual information and of its maximum, the channel capacity, because there is no deterministic (or random) way to increase it.
- The bottom line is the following fundamental result:
  the information loss due to channel transmission cannot be recovered.

Laboratory: Blahut-Arimoto algorithm

- Implement in Matlab the following iterative algorithm:
  1. Initialize $p_X(x) = p_0(x)$ (typically, the equiprobable distribution).
  2. For $k = 0, 1, 2, \ldots$, calculate

$$q_k(x|y) \triangleq \frac{p_k(x)p_{Y|X}(y|x)}{\sum_{\check{x}\in\mathcal{X}} p_k(\check{x})p_{Y|X}(y|\check{x})}$$

$$C_k \triangleq \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_k(x)p_{Y|X}(y|x)\log_2\frac{q_k(x|y)}{p_k(x)}$$

$$p_{k+1}(x) \triangleq \frac{\prod_{y\in\mathcal{Y}} q_k(x|y)^{p_{Y|X}(y|x)}}{\sum_{\check{x}\in\mathcal{X}}\prod_{y\in\mathcal{Y}} q_k(\check{x}|y)^{p_{Y|X}(y|\check{x})}}$$

  3. Terminate the iterations when $C_{k+1} - C_k$ is sufficiently small.

Laboratory: Blahut-Arimoto algorithm (cont.)

- In some cases, the capacity is subject to an additional constraint on the input probability distribution:

$$\sum_{x \in \mathcal{X}} p(x) w(x) \leq P_{tot}$$

  where $w(x) \geq 0$ can represent, for example, the power associated to the symbol $x \in \mathcal{X}$.
- The optimization problem must be modified accordingly.
- Implement in Matlab the following weighted Blahut-Arimoto algorithm:

Laboratory: Blahut-Arimoto algorithm (cont.)

1. Initialize $p_X(x) = p_0(x)$.
2. For $k = 0, 1, 2, \ldots$, calculate

$$q_k(x|y) \triangleq \frac{p_k(x)p_{Y|X}(y|x)}{\sum_{\check{x}\in\mathcal{X}} p_k(\check{x})p_{Y|X}(y|\check{x})}$$

$$C_k \triangleq \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p_k(x)p_{Y|X}(y|x)\log_2\frac{q_k(x|y)}{p_k(x)}$$

Solve the following equation for μ and set $\mu = 0$ if negative solution

$$P_{tot} = \frac{\sum_{x\in\mathcal{X}} w(x)\mathrm{e}^{-\mu w(x)}\prod_{y\in\mathcal{Y}} q_k(x|y)^{p_{Y|X}(y|x)}}{\sum_{x\in\mathcal{X}} \mathrm{e}^{-\mu w(x)}\prod_{y\in\mathcal{Y}} q_k(x|y)^{p_{Y|X}(y|x)}}$$

$$p_{k+1}(x) \triangleq \frac{\mathrm{e}^{-\mu w(x)}\prod_{y\in\mathcal{Y}} q_k(x|y)^{p_{Y|X}(y|x)}}{\sum_{\check{x}\in\mathcal{X}} \mathrm{e}^{-\mu w(\check{x})}\prod_{y\in\mathcal{Y}} q_k(\check{x}|y)^{p_{Y|X}(y|\check{x})}}$$

3. Terminate the iterations when $C_{k+1} - C_k$ is sufficiently small.

Laboratory: Blahut-Arimoto algorithm (cont.)

- The initialization of $p_X(x) = p_0(x)$ is not straightforward and depends on both $w(x)$ and P_{tot}.
- Let $(p_1, p_2, \ldots, p_M)$ be the probabilities in $p_0(x)$ and $(w_1, w_2, \ldots, w_M)$ the corresponding weights so that we have the two linear equations

$$p_1 + p_2 + \cdots + p_M = 1$$
$$w_1 p_1 + w_2 p_2 + \cdots + w_M p_M = P_{tot}$$

- Assume further that $w_1 \leq w_2 \leq \cdots \leq w_M$.
- It is plain to see that the constraint is feasible only if

$$w_1 \leq P_{tot} \leq w_M$$

Laboratory: Blahut-Arimoto algorithm (cont.)

- Let's look for a probability distribution such that $p_{3:M} = \bar{p}$, constant.
- Then, the linear equations become

$$p_1 + p_2 = 1 - (M-2)\bar{p}$$

$$w_1 p_1 + w_2 p_2 = P_{tot} - \sum_{i=3}^{M} w_i \bar{p}$$

- Their solution is

$$p_1 = \frac{w_2 - P_{tot} + \sum_{i=3}^{M}(w_i - w_2)\bar{p}}{w_2 - w_1}$$

$$p_2 = \frac{P_{tot} - w_1 - \sum_{i=3}^{M}(w_i - w_1)\bar{p}}{w_2 - w_1}$$

Laboratory: Blahut-Arimoto algorithm (cont.)

- The solution yields a valid probability distribution if $p_1, p_2, \bar{p} \geq 0$.
- This condition requires that

$$\frac{P_{tot} - w_2}{\sum_{i=3}^{M}(w_i - w_2)} < \bar{p} < \frac{P_{tot} - w_1}{\sum_{i=3}^{M}(w_i - w_1)}$$

- Equivalently,

$$P_{tot} < \frac{1}{M-2}\sum_{i=3}^{M} w_i$$

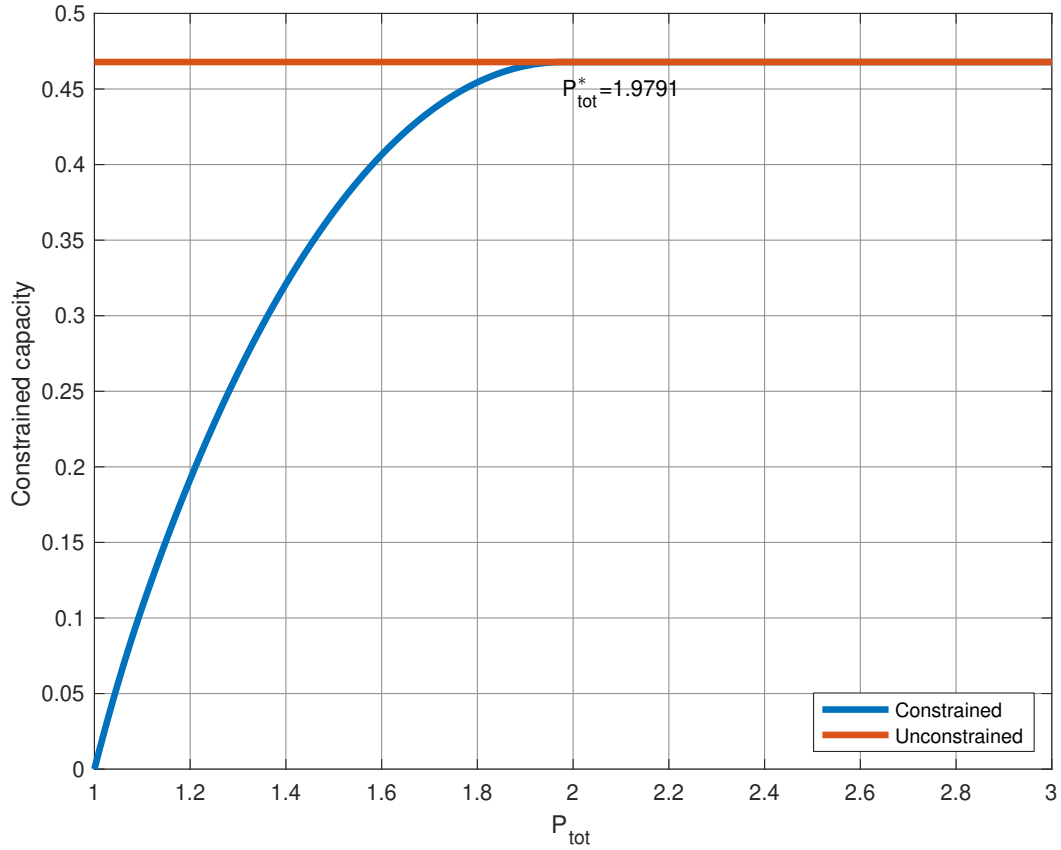
- The upper limit to P_{tot} corresponds to $\sum_{i=1}^{M} w_i \tilde{p}_i$ if $(\tilde{p}_i)_{i=1}^{M}$ is the capacity achieving distribution in the unconstrained problem since it is impossible to increase the mutual information above that limit.

Laboratory: Blahut-Arimoto algorithm (cont.)

- To illustrate this fact we consider the following example:

$$\boldsymbol{P} = \begin{pmatrix} 0.8 & 0.1 & 0.1 \\ 0.2 & 0.3 & 0.5 \\ 0.1 & 0.8 & 0.1 \\ 0.3 & 0.3 & 0.4 \end{pmatrix} \qquad \boldsymbol{w} = (1, 2, 3, 4)$$

Laboratory: Blahut-Arimoto algorithm (cont.)



Continuous communication channels

- The previous results, relevant to discrete channels, can be extended to the case when the channel input and/or output are modeled by a continuous distribution.
- We need to find the mutual information between random variables that have continuous pdf instead than a discrete probability distribution.
- To this purpose, we first show the effect of the discretization of a continuous random variable on its entropy.
- Next, we extend the argument to jointly distributed random variables.
- Finally, we derive the mutual information in the case of continuous input and output distributions.

Continuous communication channels (cont.)

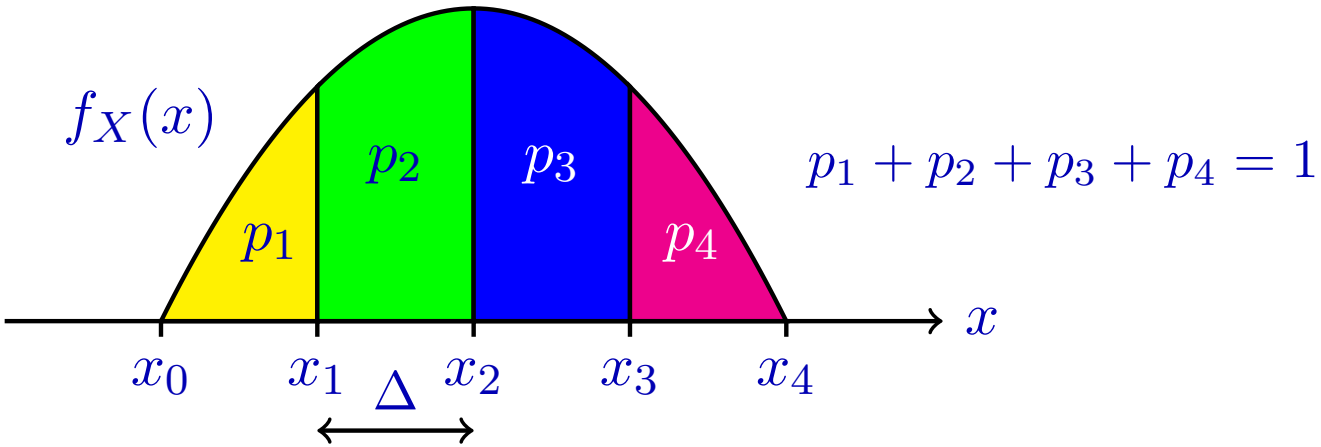
- Let X be a continuous random variable, i.e., characterized by a pdf which is a smooth function $f_X(x)$.
- We can discretize X at the precision level Δ as follows.
- Let us assume that $P(x_0 < X < x_N) = 1$, i.e., X is included in the interval (x_0, x_N) with probability 1.
- Then, we map X to a discrete random variable, X_Δ, defined by the following probability distribution:

$$p_i \triangleq P(X_\Delta = i) = P(x_i < X < x_{i+1}) = \int_{x_i}^{x_{i+1}} f_x(x)dx$$

with $\Delta \triangleq x_{i+1} - x_i$ for $i = 0, 1, \ldots, N-1$.

Continuous communication channels (cont.)

- For example, consider the case $N = 4$ and the pdf given in the following diagram:



- We can see that $p_1 = \int_{x_0}^{x_1} f_X(x)dx$ and so on, as in the definition of the previous slide.

Continuous communication channels (cont.)

- Now, we recall the approximation of an integral as a sum:

$$\int_{x_0}^{x_N} g(x)dx \approx \Delta \times \sum_{i=1}^{N} g(\xi_i)$$

- Here, ξ_i is the middle point of the interval (x_{i-1}, x_i) for $i = 1, 2, \ldots, N$.
- The approximation is accurate if the function $g(x)$ is almost constant over each interval (x_{i-1}, x_i).
- This condition is satisfied if the size of the intervals, Δ, is sufficiently small, that is, when $\Delta \to 0^+$.

Continuous communication channels (cont.)

- Let us use the previous approximation to calculate the probabilities of X_Δ:

$$p_i = \int_{x_{i-1}}^{x_i} f_X(x)dx \approx \Delta f_X(\xi_i), \qquad i = 1, \ldots, N$$

- Recall that ξ_i is the middle point of the interval (x_{i-1}, x_i), so that

$$\xi_i = \frac{x_{i-1} + x_i}{2}$$

Continuous communication channels (cont.)

- Moreover, we have

$$
\begin{aligned}
h(X) &\triangleq \int_{x_0}^{x_N} f_X(x) \log_2 \frac{1}{f_X(x)} dx \\
&\approx \Delta \sum_{i=1}^{N} f_X(\xi_i) \log_2 \frac{1}{f_X(\xi_i)} \\
&= \sum_{i=1}^{N} (\Delta f_X(\xi_i)) \log_2 \frac{\Delta}{\Delta f_X(\xi_i)} \\
&\approx \sum_{i=1}^{N} p_i \log_2 \Delta + \sum_{i=1}^{N} p_i \log_2 \frac{1}{p_i} \\
&= \log_2 \Delta + H(p_1, p_2, \ldots, p_N) \qquad = \log_2 \Delta + H(X_\Delta)
\end{aligned}
$$

Continuous communication channels (cont.)

- We shall call $h(X)$ the differential entropy of X:

$$h(X) \triangleq -\int f_X(x) \log_2 f_X(x) dx = \mathbb{E}[-\log_2 f_X(X)]$$

- The previous approximation can also be written as

$$H(X_\Delta) \approx h(X) - \log_2 \Delta.$$

- Usually, the integral defining $h(X)$ is assumed to converge, so that the differential entropy $h(X)$ is finite.
- However, we notice that $H(X_\Delta) \to \infty$ since $-\log_2 \Delta \to \infty$ as $\Delta \to 0^+$.

Continuous communication channels (cont.)

- Example. Let us consider a random variable X distributed over $(-1, 1)$ with the following pdf and cumulative distribution function (cdf):

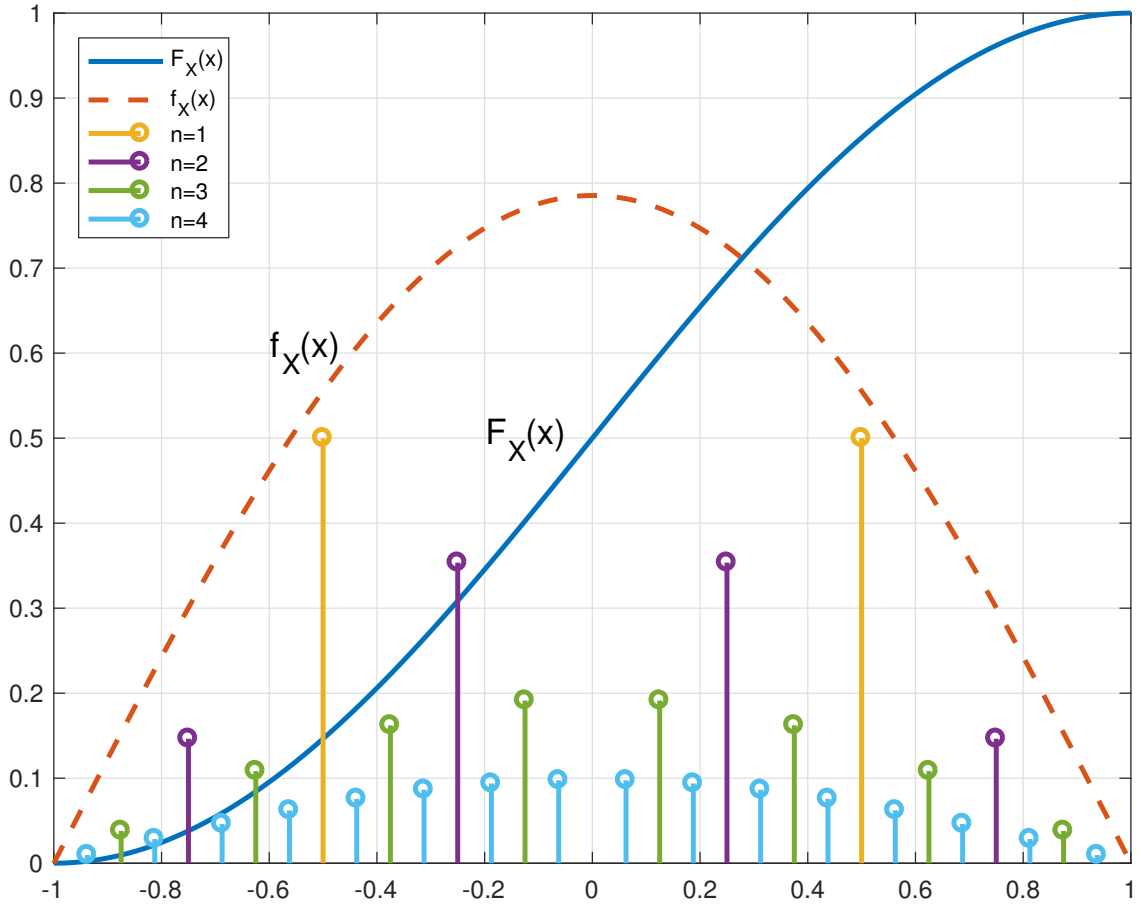
$$f_X(x) = \frac{\pi}{4} \cos\left(\frac{\pi x}{2}\right)$$
$$F_X(x) = \frac{1 + \sin(\pi x/2)}{2}$$

- In spite of the apparent difficulty, it is possible to calculate the differential entropy of X in closed form:

$$h(X) = \log_2 \frac{2\mathrm{e}}{\pi} = 0.791199$$

Continuous communication channels (cont.)

- Next, we calculate the discrete distributon of X_Δ for different values of $\Delta = 2/2^n$ and obtain the results reported in the following figure.



Continuous communication channels (cont.)

- The following table compares the discrete entropy $H(X_\Delta)$ with its approximation $h(X) - \log_2 \Delta$ for $\Delta = 2/2^n$ and $n = 1, \ldots, 8$:

n	Δ	$H(X_\Delta)$	$h(X) - \log_2 \Delta$
1	1	1	0.7912
2	0.5	1.8724	1.7912
3	0.25	2.8182	2.7912
4	0.125	3.7996	3.7912
5	0.0625	4.7937	4.7912
6	0.03125	5.7919	5.7912
7	0.015625	6.7914	6.7912
8	0.0078125	7.7913	7.7912

Continuous communication channels (cont.)

- The concept of discretization can be extended to pairs of random variables (X, Y) with joint pdf $f_{XY}(x, y)$.
- Assume again $P(x_0 < X < x_N) = 1$ and $P(y_0 < Y < y_N) = 1$.
- The joint probability distribution of (X_Δ, Y_Δ) is given by

$$\begin{aligned} p_{ij} &\triangleq P(x_i < X < x_{i+1}, y_j < Y < y_{j+1}) \\ &\approx \Delta^2 f_{XY}(\xi_i, \eta_j) \end{aligned}$$

where

$$\xi_i \triangleq \frac{x_{i-1} + x_i}{2}, \qquad \eta_j \triangleq \frac{y_{i-1} + y_i}{2}$$

Continuous communication channels (cont.)

- In this case, to approximate the joint entropy, we need a two-dimensional integral approximation:

$$\int_{x_0}^{x_N}\int_{y_0}^{y_N} g(x,y)dxdy \approx \Delta^2 \sum_{i=1}^{N}\sum_{j=1}^{N} g(\xi_i,\eta_j)$$

- Proceding as in the one-dimensional case, we get the joint differential entropy

$$\begin{aligned} h(X,Y) &\triangleq \int_{x_0}^{x_N}\int_{y_0}^{y_N} f_{XY}(x,y)\log_2\frac{1}{f_{XY}(x,y)}dxdy \\ &\approx \log_2(\Delta^2) + H(X_\Delta, Y_\Delta) \end{aligned}$$

Continuous communication channels (cont.)

- Hence,

$$H(X_\Delta, Y_\Delta) \approx h(X,Y) - \log_2(\Delta^2)$$

with increasing accuracy as $\Delta \to 0^+$.

- Now, we can use the one- and two-dimensional approximation and obtain

$$\begin{aligned} I(X_\Delta, Y_\Delta) &= H(X_\Delta) + H(Y_\Delta) - H(X_\Delta, Y_\Delta) \\ &\approx h(X) - \log_2 \Delta + h(Y) - \log_2 \Delta \\ &\quad - [h(X,Y) - \log_2(\Delta^2)] \\ &= h(X) + h(Y) - h(X,Y) \end{aligned}$$

since $-\log_2 \Delta - \log_2 \Delta + \log_2(\Delta^2) = 0$.

Continuous communication channels (cont.)

- Thus, we have this interesting result: as long as the differential entropies are finite, as $\Delta \to 0^+$,
  - the entropy of the discretized random variables goes to infinity;
  - the mutual information between discretized random variables is finite.

Continuous communication channels (cont.)

- This fact can be explained as follows.
  - The entropy of a random variable is the average information content. Assuming $\Delta \to 0^+$ is equivalent to using an infinite number of digits to represent X_Δ. Thus, the average information content must diverge.
  - The mutual information gives the capacity of a communication channel. Letting $\Delta \to 0^+$ is equivalent to using a huge precision for the channel input and output.

    If the capacity would go to infinity, that would mean we could transmit at any arbitrarily large rate just by increasing the precision of the channel output, as we had a communication philosopher's stone.

Continuous communication channels (cont.)

- In view of the previous analysis, the mutual information between two continuous random variables X and Y is defined as

$$I(X;Y) \triangleq h(X) + h(Y) - h(X,Y)$$

Additive Gaussian channel

- The additive Gaussian channel is represented by the following channel equation:

$$Y = X + Z$$

- Here, X is the channel input, Y is the output, and Z is the additive noise.
- The channel is Gaussian when the noise is Gaussian-distributed, i.e., $Z \sim \mathcal{N}(0, N)$, where N is the average noise power.
- The channel input and the noise are always independent.
- The capacity is calculated under an average input power constraint:

$$\mathbb{E}[X^2] \leq S.$$

Additive Gaussian channel (cont.)

- To derive the capacity of the additive Gaussian channel we need some properties of the differential entropy:
  1. For any constant a, $h(a+X) = h(X)$.
  2. For any two independent random variables X, Z, we have
  $$h(X+Z|X) = h(X)$$
  3. If σ_X^2 is the variance of X, then
  $$h(X) \leq h(X_G) = \frac{1}{2}\log_2(2\pi \mathrm{e}\,\sigma_X^2) \tag{21}$$
  where X_G is a Gaussian random variable with zero mean and variance σ_X^2.

Additive Gaussian channel (cont.)

- The proofs of the first two results are omitted.
- Intuitively, adding a constant to a random variable means that we shift horizontally the pdf $f_X(x)$.
- Since $h(X) = -\int f_X(x) \log_2 f_X(x) dx$ is obtained by integrating a function of the pdf itself, a horizontal shift does not change the result.
- The second result derives from the fact that $h(X+Z|X)$ corresponds to averaging with respect to X the differential entropy of $X+Z$ where X can be considered constant before averaging so that $h(X+Z|X=x) = h(x+Z) = h(Z)$ and then averaging with respect to X has no effect.

Additive Gaussian channel (cont.)

- Proof of $h(X) \leq h(X_G) = \frac{1}{2}\log_2(2\pi \mathrm{e}\,\sigma_X^2)$.
  - Since $h(X) = h(X - \mathbb{E}[X])$ we assume (without loss of generality) that X and X_G have both zero mean.
  - Moreover, they have the same variance:

$$\sigma_X^2 = \mathbb{E}[X^2] = \mathbb{E}[X_G^2].$$

  - Then, let $f(x)$ and $g(x)$ be the pdfs of X and X_G, so that

$$g(x) = \frac{1}{\sqrt{2\pi\sigma_X^2}} \mathrm{e}^{-x^2/(2\sigma_X^2)}$$

Additive Gaussian channel (cont.)

- The following identity is a basic ingredient of the proof:

$$\begin{aligned}
\mathbb{E}[\log_2 g(X)] &= \int f(x) \log_2 g(x) dx \\
&= \mathbb{E}[-\log_2(\sqrt{2\pi\sigma^2}) - X^2 \log_2 \mathrm{e} / (2\sigma^2)] \\
&= \mathbb{E}[-\log_2(\sqrt{2\pi\sigma^2}) - X_G^2 \log_2 \mathrm{e} / (2\sigma^2)] \\
&= \int g(x) \log_2 g(x) dx \\
&= \mathbb{E}[\log_2 g(X_G)]
\end{aligned}$$

Additive Gaussian channel (cont.)

- By using the logarithmic inequality $\ln(1+x) \leq x$ we can see that

$$
\begin{aligned}
h(X) - h(X_G) &= \mathbb{E}[-\log_2 f(X)] - \mathbb{E}[-\log_2 g(X_G)] \\
&= \mathbb{E}[-\log_2 f(X)] - \mathbb{E}[-\log_2 g(X)] \\
&= \mathbb{E}\left[\log_2 \frac{g(X)}{f(X)}\right] \\
&\leq \log_2 \mathrm{e} \times \mathbb{E}\left[\frac{g(X)}{f(X)} - 1\right] \\
&= \log_2 \mathrm{e} \times \left\{\int \frac{g(x)}{f(x)} f(x) dx - 1\right\} \\
&= \log_2 \mathrm{e} \times \left\{\int g(x) dx - 1\right\} \qquad = \quad 0
\end{aligned}
$$

Additive Gaussian channel (cont.)

- Finally,

$$\begin{aligned} h(X_G) &= -\mathbb{E}[\log_2 g(X_G)] \\ &= \mathbb{E}[+\log_2(\sqrt{2\pi\sigma^2}) + X_G^2 \log_2 \mathrm{e}\,/(2\sigma^2)] \\ &= \frac{1}{2}\log_2(2\pi \mathrm{e}\,\sigma^2). \end{aligned}$$

Additive Gaussian channel (cont.)

- Now, we apply the previous results to calculate the capacity of the additive Gaussian channel $Y = X + Z$, where $Z \sim \mathcal{N}(0, N)$, under the average input power contrraint $\mathbb{E}[X^2] \leq S$.
- We can write the mutual information as

$$I(X;Y) = h(Y) - h(Z). \tag{22}$$

- We search the maximum $I(X;Y)$ for all possible $f_X(x)$ with the constraint $\mathbb{E}[X^2] \leq S$.

Additive Gaussian channel (cont.)

- Since $Y = X + Z$ and Z is independent of X,

$$\begin{aligned} \sigma_Y^2 \leq \mathbb{E}[Y^2] &= \mathbb{E}[(X+Z)^2] \\ &= \mathbb{E}[X^2] + 2\mathbb{E}[XZ] + \mathbb{E}[Z^2] \\ &= \mathbb{E}[X^2] + 2\mathbb{E}[X]\mathbb{E}[Z] + N \\ &= \mathbb{E}[X^2] + N \\ &\leq S + N \end{aligned}$$

- The upper limit is attained when $\mathbb{E}[X] = 0$ and $\mathbb{E}[X^2] = S$.
- In fact, if $\mathbb{E}[X] = 0$, $\mathbb{E}[Y] = \mathbb{E}[X] + \mathbb{E}[Z] = 0$, so that $\sigma_Y^2 = \mathbb{E}[Y^2]$.
- Moreover, if $\mathbb{E}[X] = 0$, $\mathbb{E}[X^2] = \sigma_X^2 = S$.

Additive Gaussian channel (cont.)

- Therefore, applying (21), we obtain

$$h(Y) \leq h(Y_G) = \frac{1}{2}\log_2(2\pi e(S+N)) \tag{23}$$

where the maximum is attained when $X \sim \mathcal{N}(0, S)$ since the sum of two Gaussian random variables, $Y = X + Z$, is Gaussian.

- This condition, $X \sim \mathcal{N}(0, S)$, maximizes $h(Y)$ and hence $I(X;Y) = h(Y) - h(Z)$ over the possible input distributions $f_X(x)$ satisfying the power constraint $\mathbb{E}[X^2] \leq S$.

Additive Gaussian channel (cont.)

- Then, the channel capacity is achieved when $X \sim \mathcal{N}(0, S)$ and its value is

$$C = \frac{1}{2}\log_2(2\pi e(S+N)) - \frac{1}{2}\log_2(2\pi eN)$$
$$= \boxed{\frac{1}{2}\log_2\left(1+\frac{S}{N}\right)}$$

which is Shannon's capacity formula.

Weighted Water-Filling

- Some transmission techniques correspond to the simultaneous transmission of a single data stream multiplexed over a number of independent (parallel) additive Gaussian channels (as an example, OFDM: Orthogonal Frequency Division Multiplexing used in 4G cellular communications and terrestrial Digital Video Broadcasting).
- Consider a set of K independent additive Gaussian channels

$$Y_k = A_k X_k + Z_k, \qquad k = 1, \ldots, K$$

- The transmitted symbols are independent special complex Gaussian distributed: $X_k \sim \mathcal{N}(0, P_k)$.
- The noise samples are also independent special complex Gaussian distributed: $Z_k \sim \mathcal{N}(0, N_k)$.

Weighted Water-Filling (cont.)

- The channel gains A_k are constant and the channels are called parallel Gaussian channels.
- According to this assumptions, by direct application of the Shannon capacity formula, the achievable rate of each channel is

$$R_k = \frac{1}{2}\log_2(1+\mathrm{SNR}_k) = \frac{1}{2}\log_2(1+\gamma_k P_k)$$

where $\gamma_k \triangleq A_k^2/N_k$

Weighted Water-Filling (cont.)

- In some situations, we are interested in the maximum weighted sum rate

$$R \triangleq \sum_{k=1}^{K} w_k \log_2(1 + \gamma_k P_k)$$

with $w_k > 0$ for $k = 1, \ldots, K$, under a constraint on the total power:

$$\sum_{k=1}^{K} P_k \leq P.$$

- In other words, our goal is to find the power allocation $(P_k)_{k=1}^{K}$ which maximizes the weighted sum rate R under the total power constraint.
- To solve this optimization problem we note that the transmitted powers are nonnegative numbers: $P_k \geq 0$.

Weighted Water-Filling (cont.)

- The power constraint inequality can be replaced by a strict identity since the maximum weighted sum rate is surely achieved when $\sum_{k=1}^{K} P_k = P$, otherwise it would be possible to increase one of the powers and hence increase the weighted sum rate.
- To implement the equality constraint we assume

$$P_k = P \frac{x_k^2}{\sum_{\ell=1}^{K} x_\ell^2}$$

where $x_k \in \mathbb{R}$.

Weighted Water-Filling (cont.)

- Defining the function

$$\varphi(x_1,\ldots,x_K) \triangleq \sum_{k=1}^{K} w_k \log_2\left(1+\gamma_k P \frac{x_k^2}{\sum_{\ell=1}^{K} x_\ell^2}\right)$$

the optimization problem becomes

$$\max_{x_1,\ldots,x_K} \varphi(x_1,\ldots,x_K)$$

without any additional constraint.

- The maximum is then a stationary point of $\varphi(x_1,\ldots,x_K)$ corresponding to setting of all the partial derivatives with respect to x_k for $k=1,\ldots,K$ equal to 0.

Weighted Water-Filling (cont.)

- Since $P_k = P\frac{x_k^2}{x_1^2+\cdots+x_K^2}$ we have

$$\frac{\partial \varphi}{\partial x_k} = \sum_{\ell=1}^{K} \frac{w_\ell \gamma_\ell}{1+\gamma_\ell P_\ell} \frac{\partial P_\ell}{\partial x_k} \log_2 \mathrm{e}$$

$$= 2P \sum_{\ell=1}^{K} \frac{w_\ell \gamma_\ell}{1+\gamma_\ell P_\ell} \left(\frac{x_k \delta_{k,\ell}}{x_1^2+\cdots+x_K^2} - \frac{x_\ell^2 x_k}{(x_1^2+\cdots+x_K^2)^2} \right) \log_2 \mathrm{e}$$

$$= 2P \frac{x_k}{x_1^2+\cdots+x_K^2} \sum_{\ell=1}^{K} \frac{w_\ell \gamma_\ell}{1+\gamma_\ell P_\ell} \left(\delta_{k,\ell} - \frac{x_\ell^2}{x_1^2+\cdots+x_K^2} \right) \log_2 \mathrm{e}$$

$$= 2P \frac{x_k}{x_1^2+\cdots+x_K^2} \left\{ \frac{w_k \gamma_k}{1+\gamma_k P_k} - \frac{1}{P} \sum_{\ell=1}^{K} \frac{w_\ell \gamma_\ell P_\ell}{1+\gamma_\ell P_\ell} \right\} \log_2 \mathrm{e}$$

Weighted Water-Filling (cont.)

- Thus, setting

$$\lambda \triangleq \left\{ \frac{1}{P} \sum_{\ell=1}^{K} \frac{w_\ell \gamma_\ell P_\ell}{1+\gamma_\ell P_\ell} \right\}^{-1}$$

we get the following equations:

$$x_k \left\{ \frac{w_k \gamma_k}{1+\gamma_k P_k} - \frac{1}{\lambda} \right\} = 0, \qquad k = 1, \ldots, K$$

- These equations have two possible solutions:
  1. $x_k = 0 \Longrightarrow P_k = 0$
  2. $P_k = \lambda w_k - \frac{1}{\gamma_k}$

  for $k = 1, \ldots, K$.
- Now assume we know λ.

Weighted Water-Filling (cont.)

- If $\lambda w_k - \frac{1}{\gamma_k} < 0$, the second solution is impossible because it would correspond to a negative power: $P_k < 0$.
- Then, we discard it and keep the first solution, $x_k = 0$. That solution gives $P_k = 0$.
- On the contrary, if $\lambda w_k - \frac{1}{\gamma_k} > 0$, there are two possible solutions:
  1. $P_k = 0$
  2. $P_k = \lambda w_k - \frac{1}{\gamma_k}$
- Since our goal is to maximize the weighted sum rate and each component rate R_k is an increasing function of P_k, the second solution has to be chosen because it leads to a greater R_k (the first solution would give $R_k = 0$).

Weighted Water-Filling (cont.)

- Summarizing we write the solution as

$$P_k = \left(\lambda w_k - \frac{1}{\gamma_k}\right)_+$$

where we use the notation $(x)_+ \triangleq \max(0, x)$.

- The remaining problem is determining the parameter λ.
- The problem can be solved by finding the solution of the following water-filling equation:

$$\sum_{k=1}^{K} \left(\lambda w_k - \frac{1}{\gamma_k}\right)_+ = P.$$

- The following example outlines a procedure to solve the water-filling equation in a specific case.

Example of Weighted Water-Filling solution

- Let us find the weighted water-filling solution in the case corresponding to the following parameters:
  - $K = 3$
  - $(w_k)_{k=1}^K = (1, 2, 2)$
  - $(A_k)_{k=1}^K = (1, 1, 1)$
  - $(N_k)_{k=1}^K = (0.1, 0.1, 0.5)$
  - $P = 0.4$
- Our previous analysis provides the following expressions of the three optimum powers:

$$P_1 = (\lambda - 0.1)_+, \quad P_2 = (2\lambda - 0.1)_+, \quad P_3 = (2\lambda - 0.5)_+$$

- Our goal is to find λ so that $P_1 + P_2 + P_3 = P = 0.4$.

Example of Weighted Water-Filling solution (cont.)

- To this purpose we find the minimum value of λ which makes $P_k \geq 0$:
  - $P_1 \geq 0 \Longrightarrow \lambda \geq 0.1$
  - $P_2 \geq 0 \Longrightarrow \lambda \geq 0.05$
  - $P_3 \geq 0 \Longrightarrow \lambda \geq 0.25$
- As a result of the previous analysis we can write the sum of powers P_k as follows:

$$P_1+P_2+P_3 = \begin{cases} 0 & \lambda \leq 0.05 \\ (2\lambda - 0.1) & 0.05 \leq \lambda \leq 0.1 \\ (\lambda - 0.1) + (2\lambda - 0.1) & 0.1 \leq \lambda \leq 0.25 \\ (\lambda - 0.1) + (2\lambda - 0.1) + (2\lambda - 0.5) & \lambda \geq 0.25 \end{cases}$$

Example of Weighted Water-Filling solution (cont.)

- Then we proceed by $i)$ assuming that λ belongs to each of the above specified intervals, $ii)$ solving the corresponding equation for λ, and $iii)$ checking if λ effectively belongs to the interval assumed.
  1. Assume $\lambda \in [0.05, 0.1]$; solve $2\lambda - 0.1 = 0.4$, yielding $\lambda = 0.25$; discard this solution since $\notin [0.05, 0.1]$.
  2. Assume $\lambda \in [0.1, 0.25]$; solve $3\lambda - 0.2 = 0.4$, yielding $\lambda = 0.2$; accept this solution since $\in [0.1, 0.25]$.
  3. Assume $\lambda \in [0.25, \infty)$; solve $5\lambda - 0.7 = 0.4$, yielding $\lambda = 0.22$; discard this solution since $\notin [0.25, \infty)$.

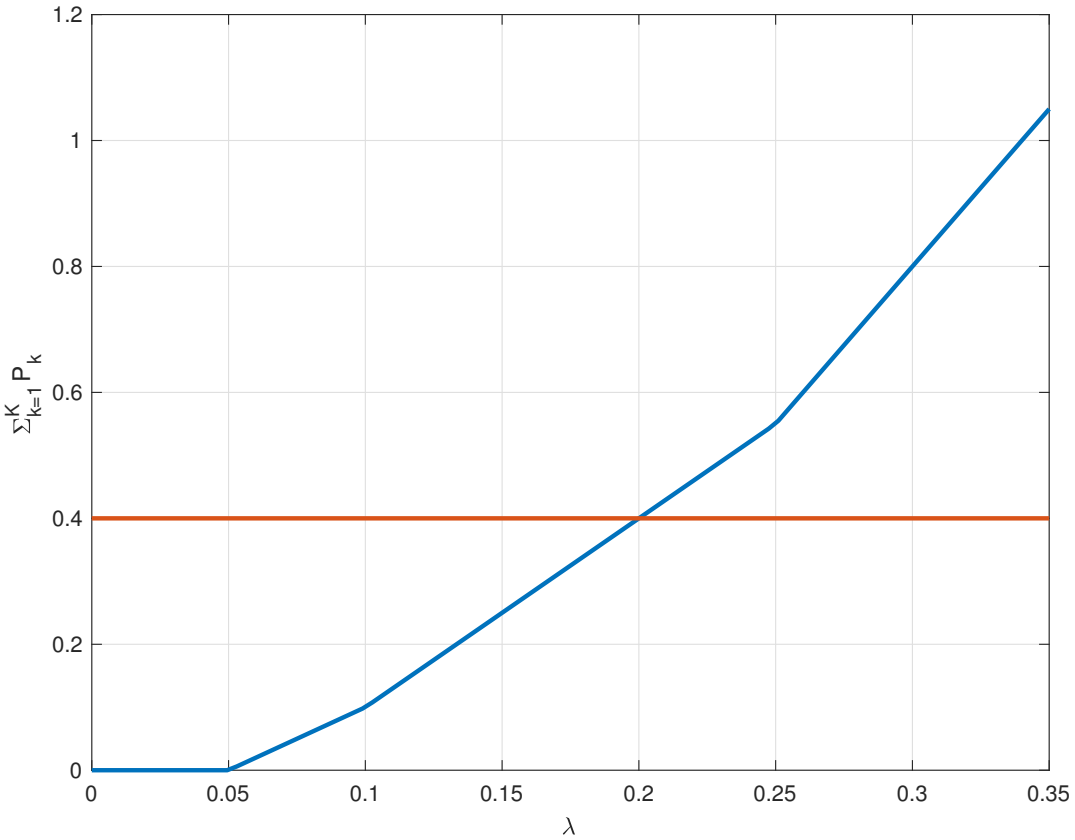
  The resulting solution is as follows:
  - The transmitted powers are $(P_k)_{k=1}^K = (0.1, 0.3, 0)$
  - The channel rates are $(R_k)_{k=1}^K = (1, 2, 0)$ bit/s/Hz
  - The weighted sum rate is $R = 5$ bit/s/Hz.

Example of Weighted Water-Filling solution (cont.)

- Every other power allocation yields a smaller weighted sum rate.
  - Consider for example $(P_k)_{k=1}^K = (0.2, 0.1, 0.1)$
  - The channel rates are $(R_k)_{k=1}^K = (1.585, 1, 0.263)$ bit/s/Hz
  - The weighted sum rate is $R = 4.111$ bit/s/Hz, which is smaller than the optimum rate of 5 bit/s/Hz
- We note that the solution is unique because of the following facts:
  - Every P_k is a monotonically increasing function of λ.
  - Therefore, the sum $\sum_{k=1}^K P_k$ is a monotonically increasing function of λ.
  - Thus, the solution of the equation $\sum_{k=1}^K P_k = P$ is unique for $P > 0$.

Example of Weighted Water-Filling solution (cont.)

- In our example we can visualize the solution by the following picture:



Laboratory: Continuous channel capacity

- Implement the weighted water-filling algorithm and test it against the results of the previous example.

Laboratory: Continuous channel capacity (cont.)

- Consider an additive channel $Y = X + Z$ where Z is uniformly distributed over $(-A, A)$.
- Discretize the channel model by assuming that
  - the signal X is distributed over an interval $(-KA, KA)$ for $K = 3$ or 5 partitioned in N equally intervals of size $\Delta = 2KA/N$;
  - the output is distributed over an interval $(-(K+1)A, (K+1)A)$ partitioned in N equally intervals of size $\Delta = 2(K+1)A/N$;
- Derive the conditional probabilities

$$P(Y \in \mathcal{Y}_j | X = x_i)$$

where $x_i = -KA + 2KA(i - 0.5)/(N-1)$ and

$$\mathcal{Y}_j = \left(\frac{(2j - N - 2)(K+1)A}{N}, \frac{(2j - N)(K+1)A}{N} \right)$$

Laboratory: Continuous channel capacity (cont.)

- Evaluate the channel capacity under a power constraint $P = \mathbb{E}[X^2]$ by using the weighted Blahut-Arimoto algorithm on the equivalent discretized channel.
- Compare the result with the capacity of an additive Gaussian channel with equivalent noise power.

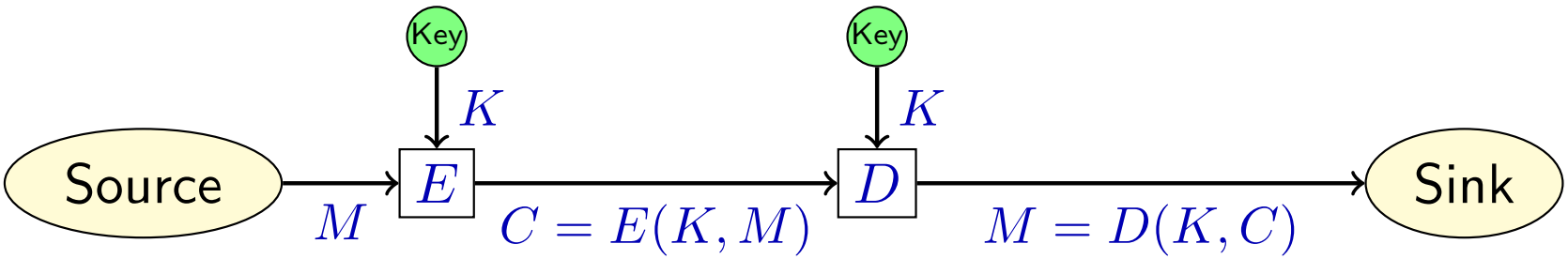
Outline

Section Outline

5 Theoretical Security
- Perfect Secrecy
- One-Time Pad
- Maurer Scheme
- LFSR
- LFSR as Stream Cipler
- A5/1
- Laboratory: The A5/1 Algorithm
- Unicity Distance
- Wiretap Channel
- Laboratory: Secrecy Capacity of the Wiretap Channel

Perfect Secrecy

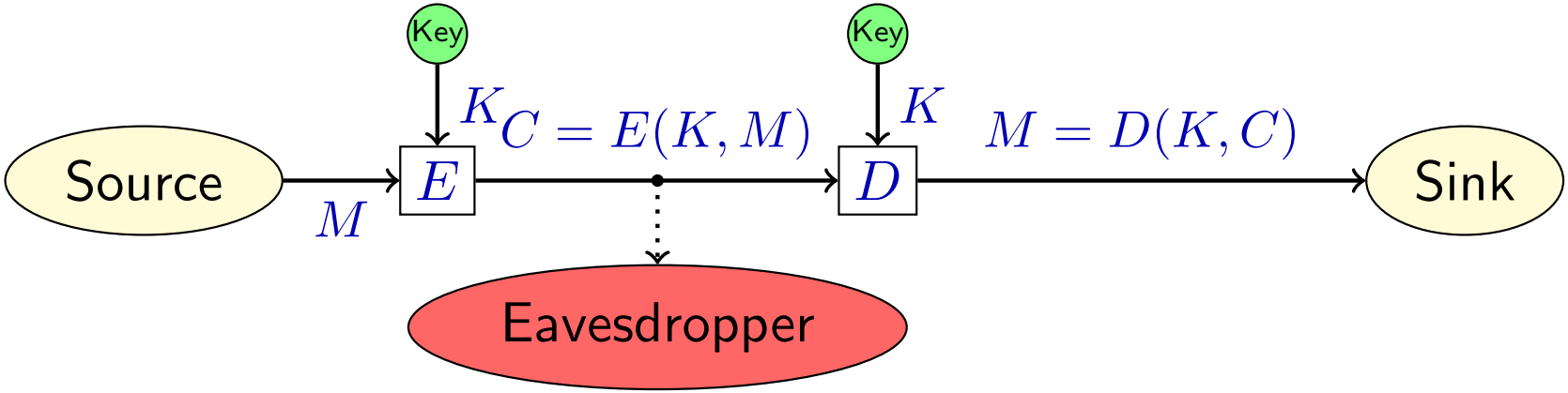
- Security is a basic requirement of a digital communication systems and involves many aspects in the design, implementation, and usage of the system.
- Here we focus on a specific aspect of security, secrecy, which affects the key exchange in symmetric encryption:



  - M: plaintext message
  - K: key, same at transmitter and receiver
  - $E(K, M)$: encrypting function, *e.g.*, $K \oplus M$
  - $D(K, C)$: decrypting function, *e.g.*, $K \oplus C$

Perfect Secrecy (cont.)

- The critical aspect in symmetric encryption is key distribution: how to transmit securely the secret key from the source to the destination.
- In the previous model we need to add a third player, the eavesdropper:



- The eavesdropper has access to the transmission channel (e.g., in a wireless system) and to the encrypted message but does not possess the key.
- Her goal is to recover the plaintext message by cryptanalytic attacks of many possible types.

Perfect Secrecy (cont.)

- Hereafter we focus on the information theoretical security of a cryptographic system.
- We model the plaintext message M, the key K, and the encrypted message C as random variables with alphabets $\mathcal{M}, \mathcal{K}$, and $\mathcal{C}$, respectively.
- Since $M = D(K, C)$, the conditional entropy $H(M|K, C) = 0$, i.e., if we know the key and the encrypted message, decryption must be always possible.
- On the contrary, if we only know C but not K, decryption must be very difficult, i.e., $H(M|C)$ must be as great as possible.

Perfect Secrecy (cont.)

- From the general entropy inequality $H(M|C) \leq H(M)$, we realize that the best conditions occur when

$$H(M|C) = H(M) \quad \Leftrightarrow \quad I(M;C) = 0$$

- In this case, M and C are statistically independent and the cryptographic system has perfect secrecy.

Perfect Secrecy (cont.)

- Since

$$H(M|K,C) = 0 = H(M,K,C) - H(K,C),$$

we have

$$\begin{aligned} H(K) &\geq H(K|C) \\ &= H(K,C) - H(C) \\ &= H(M,K,C) - H(C) \\ &= H(M,K,C) - H(M,C) + H(M,C) - H(C) \\ &= H(K|M,C) + H(M|C) \\ &\geq H(M|C) \\ &= H(M) \qquad \text{(with perfect secrecy)} \end{aligned}$$

Perfect Secrecy (cont.)

- Therefore, in order to achieve information theoretical perfect secrecy, the key entropy should exceed the plaintext message entropy so that transmitting the key would be at least as difficult as transmitting the plaintext message itself.
- The one-time pad is a cryptographic system achieving perfect secrecy but very impractical.
- It consists of using a truly random binary key sequence of the same length as the binary plaintext message without reuse.
- Trusted messengers are required to carry the key from the source to the destination before its use.

One-Time Pad

- With the one-time pad, the key is a truly random bit sequence, i.e., K is a bit sequence, $K = (K_1, \ldots, K_N)$, where the K_i's are independent equiprobable binary random variables with

$$P(K_i = 0) = P(K_i = 1) = \frac{1}{2}$$

- Equivalently, all the possible 2^N key sequences have the same probability, 2^{-N}.
- Let the plaintext message and the encrypted message be represented by the bit sequences $(M_1, \ldots, M_N)$ and $(C_1, \ldots, C_N)$, respectively.
- Encryption consists in setting $C_i = M_i \oplus K_i$ for $i = 1, \ldots, N$, where the symbol $\oplus$ denotes modulo-2 addition.
- Whatever the distribution of the plaintext message $M = (M_1, \ldots, M_N)$, the encrypted message $C = (C_1, \ldots, C_N)$ is independent of M.

One-Time Pad (cont.)

- In fact,

$$P(C = c|M = m) = P(m \oplus K = c) = P(K = m \oplus c) = 2^{-N}$$

since all the keys are equiprobable.

- Also,

$$\begin{aligned} P(C = c) &= \sum_m P(M = m)P(C = c|M = m) \\ &= 2^{-N} \sum_m P(M = m) \\ &= 2^{-N} \end{aligned}$$

- Therefore,

$$P(C = c|M = m) = P(C = c)$$

for every m, c, so that M and C are independent.

Maurer Scheme

- The Maurer cryptographic scheme is based on an ideal network of 2^L phones whose numbers are L-bit vectors.
- The ℓ-th phone ($0 \leq \ell \leq 2^L - 1$) stores a random N-bit vector R_ℓ.
- The plaintext and encrypted messages are also N-bit vectors, denoted by M and C.
- The encryption key is an L-bit vector K, which is used to call the K-th phone and obtain the N-bit vector R_K in response, which is used to encrypt the plaintext message:

$$C = M \oplus R_K$$

Maurer Scheme (cont.)

- The designed receiver knows the key K and can recover R_K and the plaintext message from the encrypted message:

$$M = C \oplus R_K$$

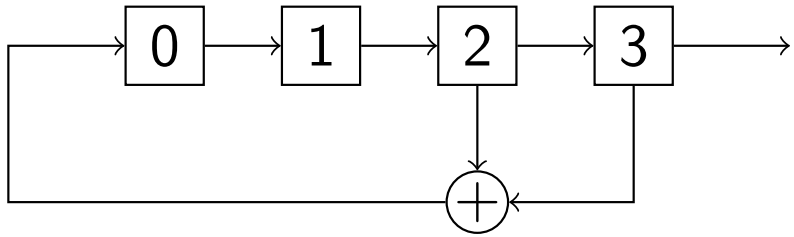
- An attacker can attempt to decrypt C only by guessing the phone number.
- If she makes 2^T random attempts, she can break the cipher with probability 2^{T-L}, otherwise, with probability $1 - 2^{T-L}$, the cryptographic scheme has perfect secrecy as the one-time pad.
- If we set $L = 100$, $2^{50} \approx 10^{15}$ attempts would break the cipher with probability $2^{50-100} \approx 10^{-15}$.

Maurer Scheme (cont.)

- The implementation of the Muarer scheme is complicated by the fact that $2^L \times N$ bits must be stored in the phone network (or in a computer memory).
- However, the principle behind it is that a cryptographic scheme requires some algorithm to convert a relatively short key K into a very long enciphering sequence R_K.
- Eventually, the principle leads to the concept of stream cipher, whose implementation uses a short key driving a pseudorandom key stream generator like the Linear Feedback Shift Register (LFSR).

LFSR

- One possible way to implement a stream cipher is using a LFSR such as the following:



- The boxes are registers with content $R_i[n]$ at time n, $i = 0, \ldots, 3$.
- At every clock time the contents propagate through the paths indicated by the arrows, so that we have the equations

$$\begin{cases} R_0[n+1] &= R_2[n] + R_3[n] \\ R_1[n+1] &= R_0[n] \\ R_2[n+1] &= R_1[n] \\ R_3[n+1] &= R_2[n] \end{cases} \qquad n = 0, 1, 2, \ldots$$

LFSR (cont.)

- The previous equations can be written as

$$\begin{cases} \sum_{n=0}^{\infty} D^{n+1} R_0[n+1] &= \sum_{n=0}^{\infty} D^{n+1}\{R_2[n] + R_3[n]\} \\ \sum_{n=0}^{\infty} D^{n+1} R_1[n+1] &= \sum_{n=0}^{\infty} D^{n+1} R_0[n] \\ \sum_{n=0}^{\infty} D^{n+1} R_2[n+1] &= \sum_{n=0}^{\infty} D^{n+1} R_1[n] \\ \sum_{n=0}^{\infty} D^{n+1} R_3[n+1] &= \sum_{n=0}^{\infty} D^{n+1} R_2[n] \end{cases}$$

- Then, defining $R_i(D) \triangleq \sum_{n=0}^{\infty} R_i[n] D^n$, we get

$$\begin{cases} R_0(D) - R_0[0] &= D[R_2(D) + R_3(D)] \\ R_1(D) - R_1[0] &= D R_0(D) \\ R_2(D) - R_2[0] &= D R_1(D) \\ R_3(D) - R_3[0] &= D R_2(D) \end{cases}$$

LFSR (cont.)

- The equations can be rewritten as

$$\begin{cases} D^3R_0(D) & = D^3R_0[0] + D^4[R_2(D) + R_3(D)] \\ R_1(D) & = R_1[0] + DR_0(D) \\ R_2(D) & = R_2[0] + DR_1[0] + D^2R_0(D) \\ R_3(D) & = R_3[0] + DR_2[0] + D^2R_1[0] + D^3R_0(D) \end{cases}$$

- Then, after eliminating $R_0(D)$, $R_1(D)$, $R_2(D)$, we have:

$$\begin{aligned} D^3R_0(D) = R_3(D) - R_3[0] - DR_2[0] - D^2R_1[0] \\ + D^3R_0[0] + D^3\{R_3(D) - R_3[0]\} + D^4R_3(D) \end{aligned}$$

LFSR (cont.)

- Solving for $R_0(D)$, we obtain

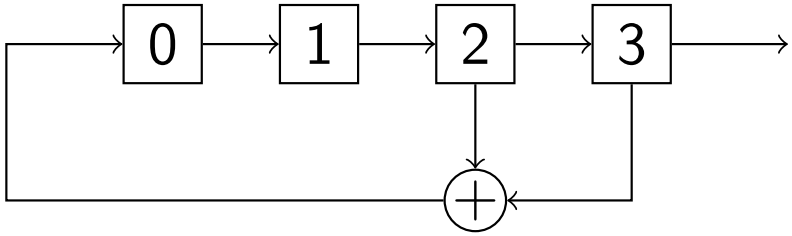
$$R_3(D) = \frac{R_3[0] + DR_2[0] + D^2R_1[0] + D^3(R_0[0] - R_3[0])}{1 - D^3 - D^4}$$

- Typically, the LFSR registers are binary, so that modulo-2 arithmetic rules apply and we get

$$R_3(D) = \frac{R_3[0] + DR_2[0] + D^2R_1[0] + D^3(R_0[0] + R_3[0])}{1 + D^3 + D^4}$$

LFSR (cont.)

- Assuming $R_0[0] = 1$ and $R_i[0] = 0$ for $i = 1, 2, 3$ in our example,



we determine the state sequence:

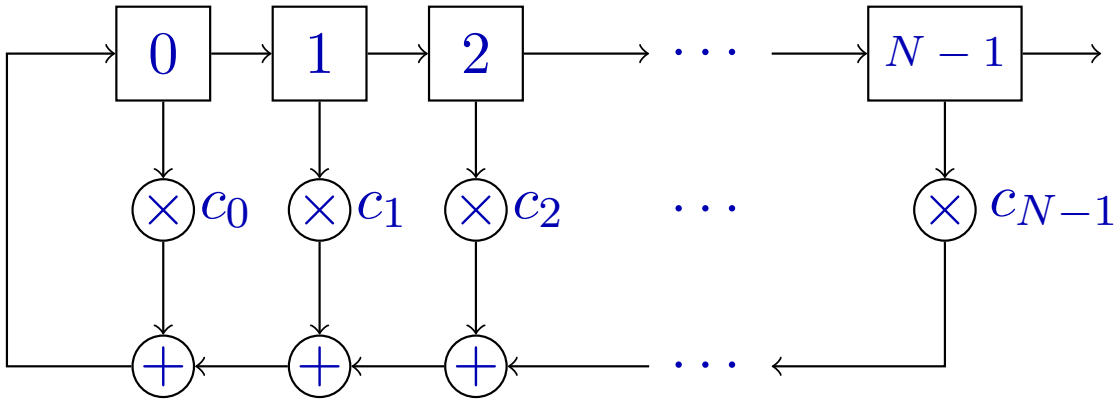
n	state	n	state
0	1000	8	1010
1	0100	9	1101
2	0010	10	1110
3	1001	11	1111
4	1100	12	0111
5	0110	13	0011
6	1011	14	0001
7	0101	15	1000

LFSR (cont.)

- As we can see, the state sequence is periodic with period 15.
- This is a general property of the LFSR's: the number of possible states is finite and must repeat itself as a sequence.
- The all-zero state repeats itself immediately so that it is excluded.
- Therefore, the maximum period of a LFSR is $2^N - 1$ where N is the number of registers.
- The period reaches the maximum if the connection polynomial ($1 + D^3 + D^4$ in this case) is primitive.
- The connection polynomial can be obtained for a general LFSR as the one indicated in the picture:

LFSR (cont.)

- General LFSR:



- The LFSR equations are summarized by

$$R_0[n+1] = \sum_{i=0}^{N-1} c_i R_i[n]$$

- We assume that $R_i[n] = 0$ for any $n < 0$.
- We assume that all $R_i[0]$ are known (initial state).

LFSR (cont.)

- We note that $R_i[n] = R_{N-1}[n+N-i-1]$ (it takes $(N-i-1)$ clock cycles to move the output of cell i to cell $N-1$).
- The LFSR equations can be transformed into equations in $R_{N-1}[n]$ as follows:

$$R_{N-1}[n+N] = \sum_{i=0}^{N-1} c_i R_{N-1}[n+N-1-i]$$

- Each equation is multiplied by D^{n+N} and added for $n=0$ to ∞:

$$\sum_{n=0}^{\infty} D^{n+N} R_{N-1}[n+N] = \sum_{i=0}^{N-1} c_i \sum_{n=0}^{\infty} D^{n+N} R_{N-1}[n+N-1-i]$$

- The previous equations can be rewritten in the following form:

$$\sum_{n=N}^{\infty} R_{N-1}[n] D^n = \sum_{i=0}^{N-1} c_i D^{i+1} \sum_{n=N-1-i}^{\infty} R_{N-1}[n] D^n$$

LFSR (cont.)

- Defining $R_i(D) \triangleq \sum_{n=0}^{\infty} R_i[n]D^n$ we can write the LFSR equations in the following finite form:

$$\left(1 - \sum_{i=0}^{N-1} c_i D^{i+1}\right) R_{N-1}(D)$$

$$= \sum_{n=0}^{N-1} R_{N-1}[n]D^n - \sum_{i=0}^{N-1} c_i \sum_{n=0}^{N-2-i} R_{N-1}[n]D^{n+i+1}$$

$$= \sum_{n=0}^{N-1} R_{N-1-n}[0]D^n - \sum_{i=0}^{N-1} c_i \sum_{n=0}^{N-2-i} R_{N-1-n}[0]D^{n+i+1}$$

LFSR (cont.)

- Therefore, $R_{N-1}(D) =$

$$\frac{\sum_{n=0}^{N-1} R_{N-1-n}[0]D^n - \sum_{i=0}^{N-1} c_i \sum_{n=0}^{N-2-i} R_{N-1-n}[0]D^{n+i+1}}{1 - \sum_{i=0}^{N-1} c_i D^{i+1}}$$

- Taking into account the binary characteristics of these expressions, the connection polynomial is

$$1 + \sum_{i=0}^{N-1} c_i D^{i+1}$$

- In our example, $N = 3$ and $c_0 = c_1 = 0, c_2 = c_3 = 1$, so that the connection polynomial is $1 + D^3 + D^4$.

LFSR as Stream Cipher

- AN LFSR is not usable as a stream cipher because it is vulnerable to a known plaintext attack of length $2N$.
- A known plaintext attack consists of knowing a sequence of plaintext and encrypted message bits, which is equivalent to knowing $2N$ stream cipher bits.
- Let's go back to our LFSR with $N = 4$ registers and write its equations as if we didn't know the connection coefficients.

$$\begin{cases} R_0[n+1] &= c_0 R_0[n] + c_1 R_1[n] + c_2 R_2[n] + c_3 R_3[n] \\ R_1[n+1] &= R_0[n] \\ R_2[n+1] &= R_1[n] \\ R_3[n+1] &= R_2[n] \end{cases}$$

- Let the attacker know $R_3[n]$ for $n = 0, \ldots, 2 \times 4 - 1$.

LFSR as Stream Cipher (cont.)

- Now, we want to rewrite the equations using only $R_3[n]$.
- To this purpose, we notice that, by using the last three equations of the previous group, we have

$$\begin{cases} R_0[n] &= R_1[n+1] = R_2[n+2] = R_3[n+3] \\ R_1[n] &= R_2[n+1] = R_3[n+2] \\ R_2[n] &= R_3[n+1] \end{cases}$$

- Thus, the first equation of the LFSR becomes

$$R_3[n+4] = c_0 R_3[n+3] + c_1 R_3[n+2] + c_2 R_3[n+1] + c_3 R_3[n]$$

LFSR as Stream Cipher (cont.)

- Setting $n = 0, 1, 2, 3$ in the previous equation and $K \equiv R_3$ for ease of notation, we get

$$\begin{cases} K[3]c_0 + K[2]c_1 + K[1]c_2 + K[0]c_3 = K[4] \\ K[4]c_0 + K[3]c_1 + K[2]c_2 + K[1]c_3 = K[5] \\ K[5]c_0 + K[4]c_1 + K[3]c_2 + K[2]c_3 = K[6] \\ K[6]c_0 + K[5]c_1 + K[4]c_2 + K[3]c_3 = K[7] \end{cases}$$

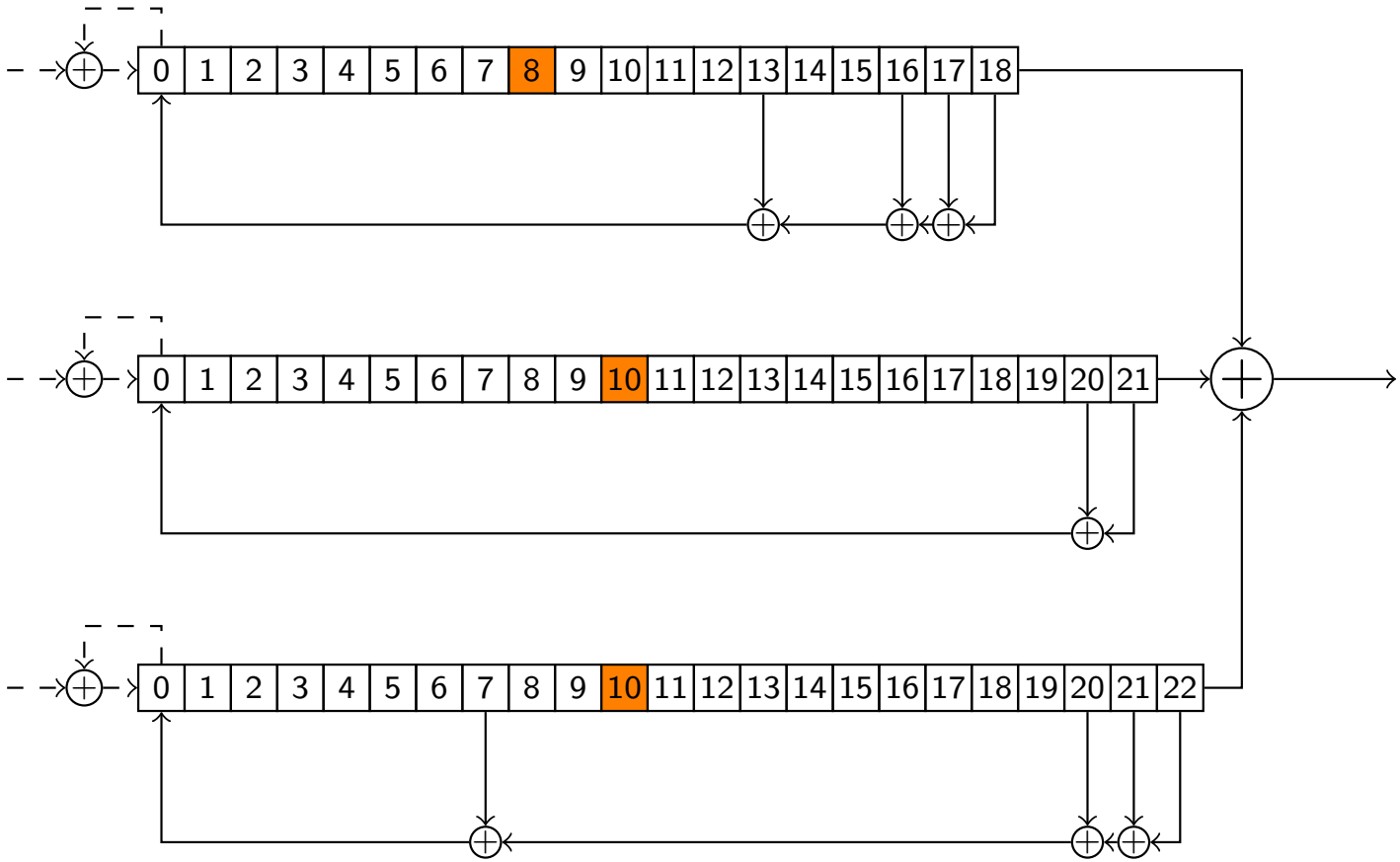
- Thus, breaking the stream cipher is equivalent to solving a linear equation system of four equations in the four unknown c_0, c_1, c_2, c_3.
- The only information required is a set of 2×4 consecutive values of $K[n]$.
- Therefore, a cryptographic system cannot be based on a LFSR as a stream cipher.

LFSR as Stream Cipher (cont.)

- In order to address this issues, different classes of nonlinear feedback shift registers (NFSR's) have been studied, a class of them being based on irregular clocking like the A5/1 algorithm used in GSM.

A5/1

- A safer pseudorandom bit generator is the A5/1 stream cipher used for GSM transmission and described in the following diagram:

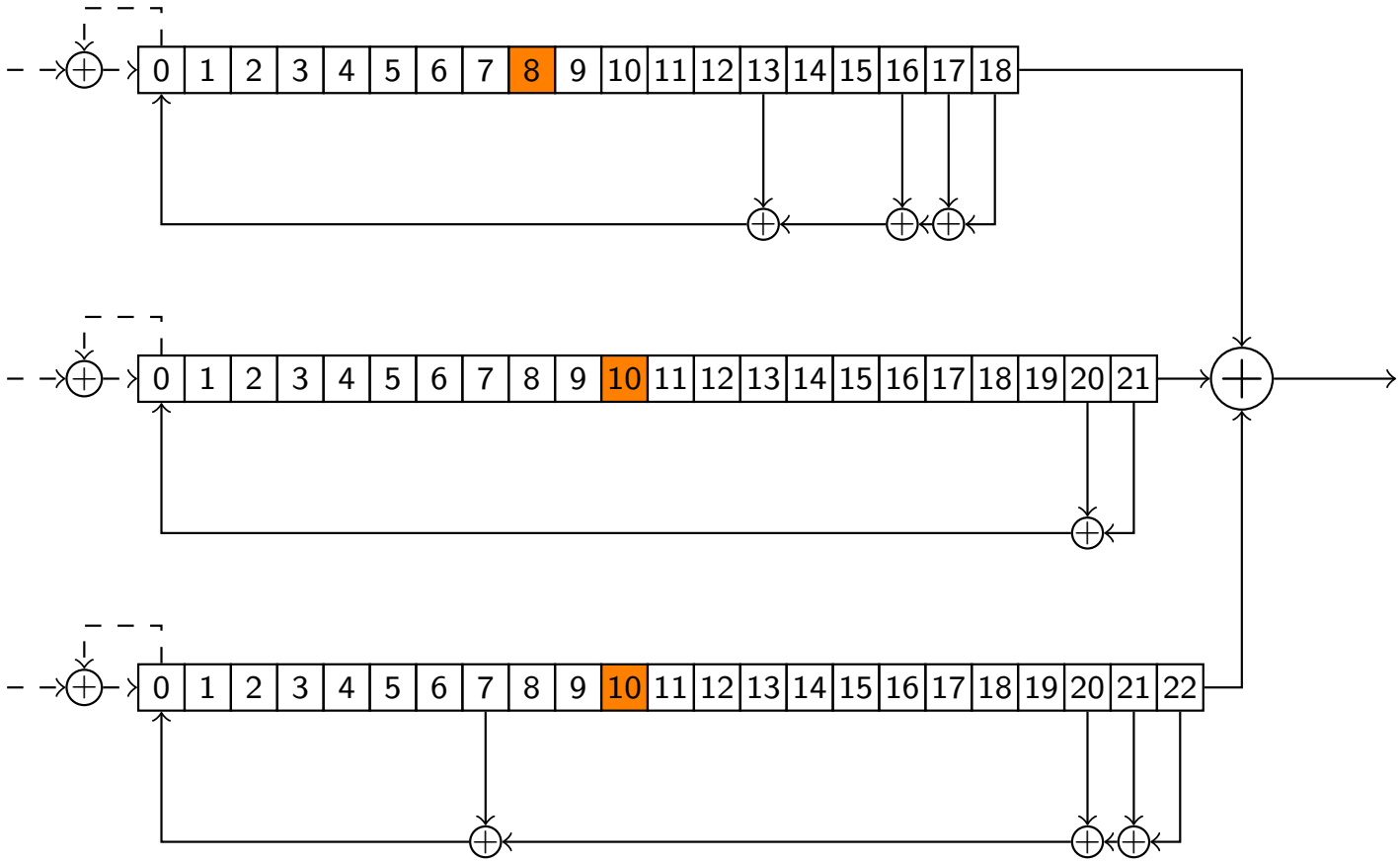


A5/1 (cont.)

- The A5/1 bit stream is the XOR of three LFSR outputs with irregular clocking.
- Each LFSR is clocked if its clocking bit (orange) agrees with the majority of the three clocking bits.
- For example, if the three clocking bits are $1, 1, 0$, the majority bit is 1, so that the first two LFSR's clock while the third doesn't.
- The initialization of the LFSR states is done by feeding their LSB's with the private key followed by the public session number.
- Next, the LFSR's output the 114 bit stream cipher for one GSM block.

Laboratory: The A5/1 Algorithm

- Implement the A5/1 algorithm in Matlab.



Unicity Distance

- The concept of unicity distance was introduced in the paper
  C. Shannon, "Communication theory of secrecy systems," *The Bell System Technical Journal*, 1949.
- The concept arises from the fact that the longer is the encrypted message known to an attacker, the higher is the chance that the encryption key can be recovered.
- Let us denote by M_L and C_L the plaintext and encrypted message of length L obtained by an encryption scheme with fixed key K.
- The key uncertainty is $H(K|C_L)$. In general, the key uncertainty decreases with the message length L.
- Since $C = E(K, M)$, C_L is a deterministic function of M_L given the key K. Thus,

$$H(C_L|K) = H(M_L|K) \quad \Rightarrow \quad H(K, C_L) = H(K, M_L)$$

Unicity Distance (cont.)

- Now, assume that the plaintext and encrypted message share the same alphabet $\mathcal{X}$.
- We have

$$\begin{aligned} H(K|C_L) &= H(K, C_L) - H(C_L) \\ &= H(K, M_L) - H(C_L) \\ &= H(K) + H(M_L) - H(C_L) \\ &\geq H(K) + H(M_L) - L \log_2 |\mathcal{X}| \end{aligned}$$

since the entropy of the encrypted message cannot exceed the case of iid equiprobable symbols over $\mathcal{X}$.

Unicity Distance (cont.)

- On the other hand, the message is typically from a spoken language and its symbols are not iid equiprobable.
- Rather, they have a limiting entropy rate $\bar{H} < \log_2 |\mathcal{X}|$.
- Hence, the previous inequality becomes

$$H(K|C_L) \geq H(K) - LD_X$$

  where $D_X \triangleq \log_2 |\mathcal{X}| - \bar{H}$ is called the redundancy of the source X.
- Thus, in order to have the possibility of recovering the encryption key K, the lower bound must be negative or 0, which occurs when

$$L \geq L_U \triangleq \frac{H(K)}{D_X}$$

- Then, L_U is called the unicity distance.

Unicity Distance (cont.)

- As an example of application of this concept, consider the encryption scheme called substitution cipher.
- The scheme corresponds to a permutation of the English alphabet, such as

$$A \to B, B \to G, C \to J, D \to B, \ldots, Z \to A$$

- Since the English alphabet has 26 letters, the number of possible permutations is $26! = 4.03 \cdot 10^{26}$.
- The key is the permutation and is selected randomly and equiprobably among $26!$ possibilities, so that the entropy of the key is

$$H(K) = \log_2(26!) = 88.38$$

Unicity Distance (cont.)

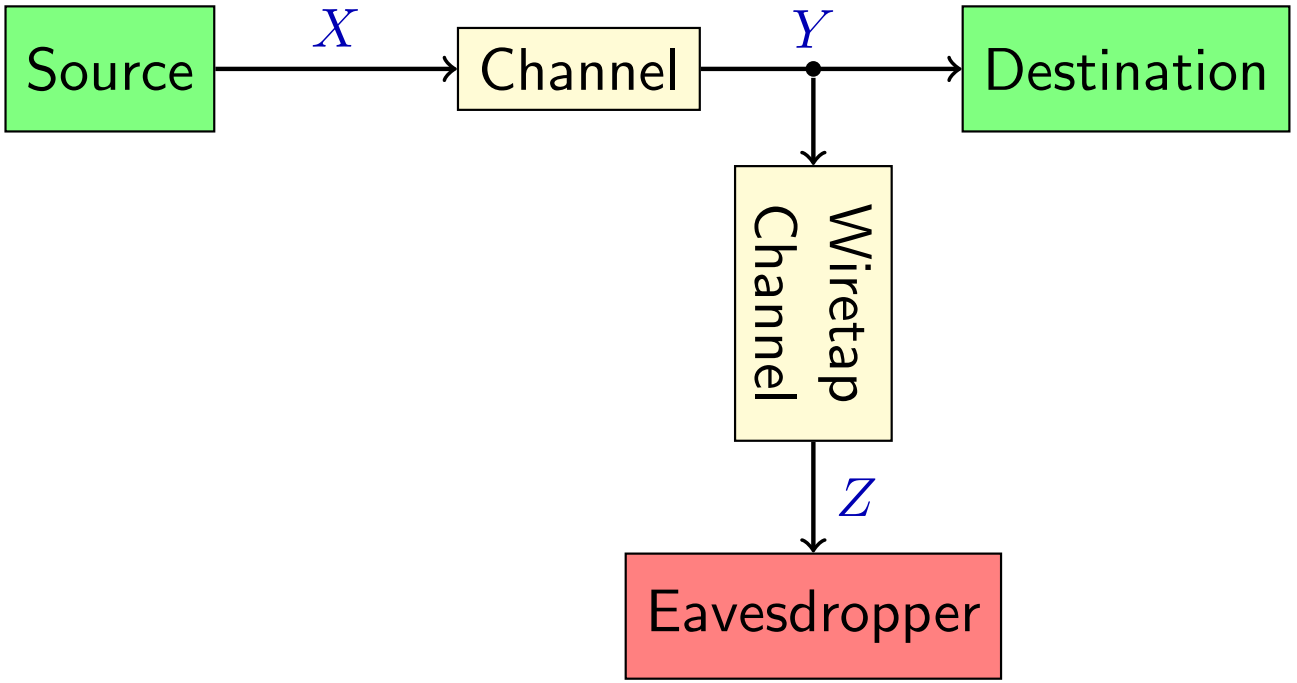
- The English language has an entropy rate approximately equal to 2.62 bit/character so that the redundancy is $\log_2 26 - 2.62 = 2.08$.
- As a result, the unique distance is

$$L_U = \frac{88.38}{2.08} = 42.48$$

- This means that a message of 43 character encrypted from the English language by a substitution cipher is potentially vulnerable to a known-ciphertext attack.
- This vulnerability can be circumvented by source encoding the plaintext message, so that the redundancy becomes very small and the unicity distance very large.

Wiretap Channel

- The concept of wiretap channel model was introduced in the paper
  A.D. Wyner, "The wire-tap channel,"
  The Bell System Technical Journal, 1975
- The block diagram of this channel is represented as follows:



Wiretap Channel (cont.)

- The situation depicted represents that of an unauthorized eavesdropper wiretapping the transmission channel at the receiver and obtaining a degraded (more noisy) version of the received signal.
- By the data-processing inequality we know that

$$I(X;Z) \leq I(X;Y)$$

- Therefore, the authorized receiver (signal Y) can achieve a better rate than the eavesdropper (signal Z).
- The transmitter can use a randomized encoder (known only to the authorized receiver) in order to set the leakage rate to the eavesdropper equal to zero.

Wiretap Channel (cont.)

- The resulting achievable rate to the authorized receiver, under the above condition, is called secrecy capacity and can be evaluated by

$$C_S = \max_{p_X(x)} [I(X;Y) - I(X;Z)] \tag{24}$$

- Example. Let the channels be BSC with error probabilities p_1 and p_E and calculate the secrecy capacity.
- The cascade of two discrete channels has a probability matrix that is the product of the two channels' probability matrices:

$$\boldsymbol{P}_{1+2} = \boldsymbol{P}_1 \boldsymbol{P}_2$$

Wiretap Channel (cont.)

- In the case of the cascade of two BSC's with error probabilities p_1 and p_E we obtain another BSC with error probability

$$p_2 = p_1(1-p_E) + (1-p_1)p_E = p_1 + p_E - 2p_1p_E$$

  We can see that if $0 \leq p_1, p_E \leq \frac{1}{2}$, then $0 \leq p_1 \leq p_2 \leq \frac{1}{2}$.
- Then, we assume that the transmitted signal X has distribution given by $P(X=0) = \alpha$ and $P(X=1) = \bar{\alpha} \triangleq 1-\alpha$.
- The mutual informations of the two channels are

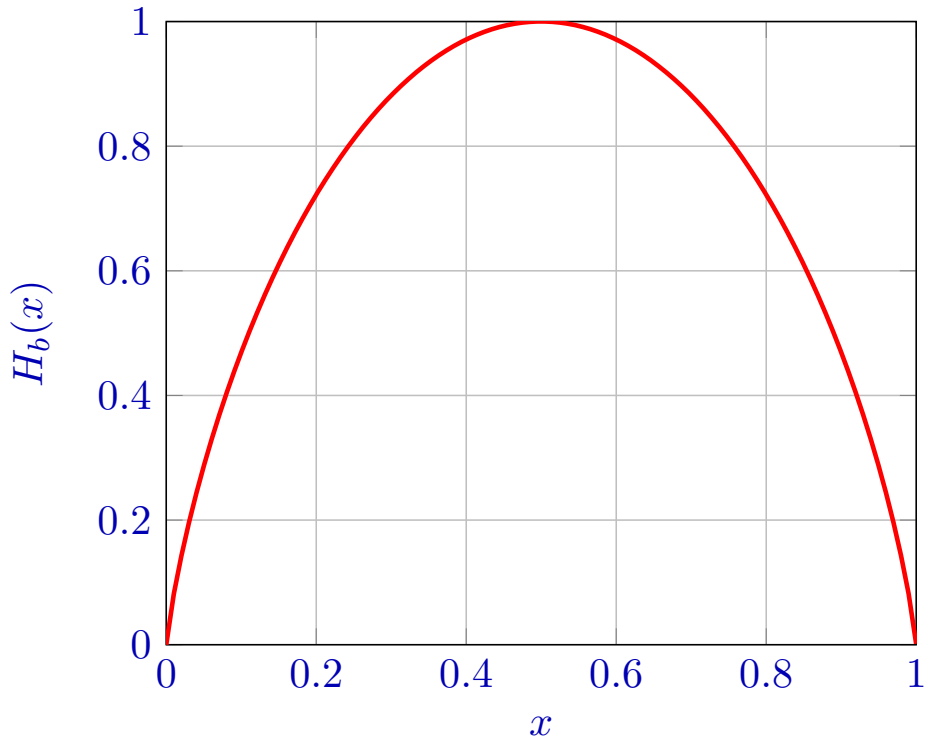
$$I(X;Y) = H_b(\alpha\bar{p}_1 + \bar{\alpha}p_1) - H_b(p_1)$$
$$I(X;Z) = H_b(\alpha\bar{p}_2 + \bar{\alpha}p_2) - H_b(p_2)$$

Wiretap Channel (cont.)

- Then,

$$C_S = \max_{0 \le \alpha \le 1} \{H_b(\alpha \bar{p}_1 + \bar{\alpha} p_1) - H_b(\alpha \bar{p}_2 + \bar{\alpha} p_2)\} - H_b(p_1) + H_b(p_2)$$

- Recall the behavior of $H_b(x)$:



Wiretap Channel (cont.)

- Since $0 \leq p_1 \leq p_2 \leq \frac{1}{2}$, we can see that, if $0 \leq \alpha \leq \frac{1}{2}$,

$$0 \leq \alpha\bar{p}_1 + \bar{\alpha}p_1 \leq \alpha\bar{p}_2 + \bar{\alpha}p_2 \leq \frac{1}{2}$$

and, if $\frac{1}{2} \leq \alpha \leq 1$,

$$1 \geq \alpha\bar{p}_1 + \bar{\alpha}p_1 \geq \alpha\bar{p}_2 + \bar{\alpha}p_2 \geq \frac{1}{2}$$

because

$$\begin{aligned}(\alpha\bar{p}_1 + \bar{\alpha}p_1) - (\alpha\bar{p}_2 + \bar{\alpha}p_2) &= \alpha(1-2p_1) + p_1 - [\alpha(1-2p_2) + p_2] \\ &= (p_1 - p_2)(1-2\alpha)\end{aligned}$$

Wiretap Channel (cont.)

- As a result, for any $0 \leq \alpha \leq 1$, we have

$$H_b(\alpha\bar{p}_1 + \bar{\alpha}p_1) \leq H_b(\alpha\bar{p}_2 + \bar{\alpha}p_2)$$

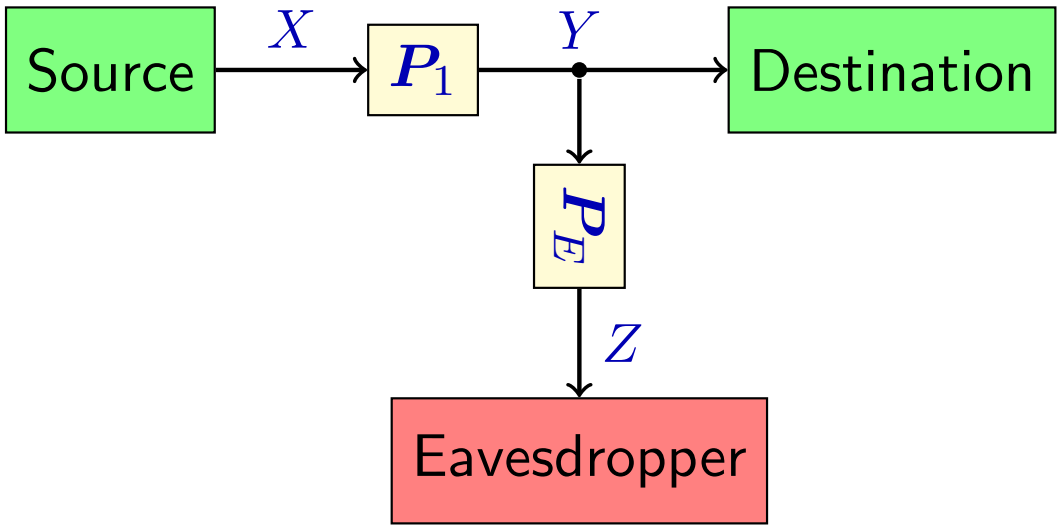
  with equality only if $\alpha = \frac{1}{2}$.
- Therefore, the secrecy capacity is given by

$$C_S = H_b(p_2) - H_b(p_1)$$

- Note that, in this case, the secrecy capacity is the difference between the capacity of the authorized channel $1 - H_b(p_1)$ and that of the eavesdropper channel $1 - H_b(p_2)$ (which is lower).

Laboratory: Wiretap Channel Secrecy Capacity

- We address the problem of calculating the secrecy capacity (24) of the following degraded wiretap channel:



Laboratory: Wiretap Channel Secrecy Capacity (cont.)

- Let $\boldsymbol{P}_1$ and $\boldsymbol{P}_E$ be two channel matrices of compatible sizes (so that $\boldsymbol{P}_2 \triangleq \boldsymbol{P}_1\boldsymbol{P}_E$ exists).
- Let $\boldsymbol{p} = (p_1, \ldots, p_{|\mathcal{X}|})$ be the probability row-vector of $X \in \mathcal{X}$, a finite alphabet.
- Then, $\boldsymbol{p}\boldsymbol{P}_1$ and $\boldsymbol{p}\boldsymbol{P}_2$ are the probability row-vectors of Y and Z, respectively.
- Let $\boldsymbol{P}_{k,i}$ be the ith row of the matrix $\boldsymbol{P}_k, k = 1, 2$. We have:

$$I(X;Y) = H(\boldsymbol{p}\boldsymbol{P}_1) - \sum_{i=1}^{|\mathcal{X}|} p_i H(\boldsymbol{P}_{1,i})$$

$$I(X;Z) = H(\boldsymbol{p}\boldsymbol{P}_2) - \sum_{i=1}^{|\mathcal{X}|} p_i H(\boldsymbol{P}_{2,i})$$

Laboratory: Wiretap Channel Secrecy Capacity (cont.)

- Then, the secrecy capacity can be written as

$$C_S = \max_{\boldsymbol{p}} \left\{ H(\boldsymbol{p}\boldsymbol{P}_1) - H(\boldsymbol{p}\boldsymbol{P}_2) - \sum_{i=1}^{|\mathcal{X}|} p_i [H(\boldsymbol{P}_{1,i}) - H(\boldsymbol{P}_{2,i})] \right\}$$

- It can be shown that the function

$$\varphi(\boldsymbol{p}) \triangleq -\left\{ H(\boldsymbol{p}\boldsymbol{P}_1) - H(\boldsymbol{p}\boldsymbol{P}_2) - \sum_{i=1}^{|\mathcal{X}|} p_i [H(\boldsymbol{P}_{1,i}) - H(\boldsymbol{P}_{2,i})] \right\}$$

  is convex in $\boldsymbol{p}$.

- Write a Matlab program to find the secrecy capacity for given matrices $\boldsymbol{P}_1, \boldsymbol{P}_E$ by using the Optimization Toolbox (the function to be used is `fmincon`).