Űrkommunikáció
Space Communication
2023/3.

Ergodicity of stochastic processes

A **stochastic process** is said to be **ergodic** if its statistical properties can be deduced from a single realization (sufficiently long, random sample) of the process.

Ergodicity: If **ensemble average always equals time average**, then the system is ergodic

Example for WWS i.e. $m_\xi\ (t) = m_\xi\ (t_0) = m_\xi$ and $K_\xi\ (t_0, t_0 + \tau) = K_\xi\ (\tau)$ processes

- **Mean-ergodic** process:

$$m_\xi\ = \lim_{T\to\infty} \frac{1}{T} \int_{t_0}^{t_0+T} \xi_t\ dt$$

- **Auto-covariance-ergodic** process:

$$K_\xi\ (\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{t_0}^{t_0+T} (\xi_t - m_\xi\ ) \cdot (\xi_{t+\tau} - m_\xi\ ) dt$$

- **Autocorrelation-ergodic** process:

$$R_\xi\ (\tau) = \lim_{T\to\infty} \frac{1}{T} \int_{t_0}^{t_0+T} \xi_t \cdot \xi_{t+\tau} dt$$

A process which is ergodic in the mean and auto-covariance is sometimes called **ergodic in the wide sense**.

Ergodicity of stochastic processes

- **Example for Ergodic process**

Each resistor has an associated **thermal noise** that depends on the temperature.

Experiment: Take N resistors (N should be very large) and plot the voltage across those resistors for a long period. For each resistor you will have a waveform, which is a **realization of the thermal noise process.**

Time average: Calculate the average value of that waveform;

Ensemble average: There are N waveforms as there are N resistors. Take a particular instant of time t_i in all those plots and find the average value;

Mean-ergodic: Time average = Ensemble average

- **Example for Non-ergodic process**

Suppose that we have two coins: one coin is fair and the other has two heads.

Fair coin 0  1  False coin: 1  1 

We choose (at random) one of the coins first, and then perform a sequence of independent tosses of our selected coin.

Ensemble average is 1⁄2 (1⁄2 + 1) = 3⁄4

Time average: the long-term average is 1⁄2 for the fair coin and 1 for the two-headed coin.

So the long term time-average is either 1/2 or 1.

The process is **not ergodic in mean**.

Entropy of stochastic processes

Goal of communication: Transmit or store not just one random variable but a series of random variables.

Let us diel with **discrete stochastic processes** which are the series (ordered in space or time) of random variables. Most of our findings will be also valid for continuous processes.

Recap *probability theory*: **Joint and conditional probability (Bayes's theorem)**

- Two discrete random variables X and Y

$$p_{X,Y}(x,y) = Prob(X = x \; and \; Y = y) =$$
$$= Prob(Y = y|X = x) \cdot Prob(X = x) = Prob(X = x|Y = y) \cdot Prob(Y = y)$$

Short notations:

$$p(x,y) = p(y|x) \cdot p(x) = p(x|y) \cdot p(y)$$
$$p(y|x) = \frac{p(x,y)}{p(x)} \quad and \quad p(x|y) = \frac{p(x,y)}{p(y)}$$

- ***n* discrete random variables** $X_1, X_2, \dots, X_n$ (or short $\bar{X}$ and $\bar{x} = [x_1, x_2, \dots, x_n]$)

$$p(\bar{x}) = p(x_1, x_2, \dots, x_n) = p(x_1) \cdot p(x_2|x_1) \cdot p(x_3|x_1, x_2) \cdots p(x_n|x_1, x_2, \dots, x_{n-1})$$

This identity is known as the **chain rule** of probability.

$$p(\bar{x}) = p(x_1, x_2, \dots, x_n) = \prod_{i=1}^{n} p(x_i|x_1, x_2, \dots, x_{i-1})$$

Entropy of stochastic processes

Let us start with some definitions:

Vector of n **discrete random variables** $\bar{X} = [X_1, X_2, \dots, X_n]$ and outcome $\bar{x} = [x_1, x_2, \dots, x_n]$

Def.: **Conditional Information**: The self-information of an event with the knowledge of the previous events:

$$I(x_n|x_1, x_2, \dots, x_{n-1}) = ld\,\frac{1}{p(x_n|x_1, x_2, \dots, x_{n-1})}\ [bit, Shannon]$$

Def.: **Conditional Entropy**: The average of the conditional information:

$$H(X_n|X_1, X_2, \dots, X_{n-1}) = E\{I(x_n|x_1, x_2, \dots, x_{n-1})\} =$$

$$= \sum_{\bar{x}} p(\bar{x}) \cdot ld\,\frac{1}{p(x_n|x_1, x_2, \dots, x_{n-1})}\ \left[\frac{bit}{symbol}\right]$$

Def.: **Joint Information**: Amount of Information conveyed by a block of random variables:

$$I(x_1, x_2, \dots, x_n) = ld\,\frac{1}{p(x_1, x_2, \dots, x_n)}\ [bit, Shannon]$$

Def.: **Joint Entropy** or **Block Entropy**: The average of the joint information:

$$H(X_1, X_2, \dots, X_n) = H(\bar{X}) = E\{I(x_1, x_2, \dots, x_n)\} =$$

$$= \sum_{\bar{x}} p(\bar{x}) \cdot ld\,\frac{1}{p(x_1, x_2, \dots, x_n)}\ \left[\frac{bit}{symbol}\right] =$$

Entropy of stochastic processes

Cont. **Joint Entropy** or **Block Entropy**:

$$H(\bar{X}) = \sum_{\bar{x}} p(\bar{x}) \cdot ld \frac{1}{p(\bar{x})} = -\sum_{\bar{x}} p(\bar{x}) \cdot ld\, p(\bar{x}) \xrightarrow{\text{chain rule}}$$

$$= -\sum_{\bar{x}} p(\bar{x})\, ld \prod_{i=1}^{n} p(x_i | x_1, x_2, \ldots, x_{i-1}) \xrightarrow{\text{Log of product}}$$

$$= -\sum_{\bar{x}} p(\bar{x}) \cdot [ld\, p(x_1) + ld\, p(x_2|x_1) + ld\, p(x_3|x_1, x_2) + \cdots + ld\, p(x_n|x_1, x_2, \ldots, x_{n-1})] =$$

$$= \sum_{\bar{x}} p(\bar{x}) \cdot \left[ld \frac{1}{p(x_1)} + ld \frac{1}{p(x_2|x_1)} + ld \frac{1}{p(x_3|x_1, x_2)} + \cdots + ld \frac{1}{p(x_n|x_1, x_2, \ldots, x_{n-1})} \right] =$$

$$= \underbrace{\sum_{\bar{x}} p(\bar{x}) \cdot ld \frac{1}{p(x_1)}}_{H(X_1)} + \underbrace{\sum_{\bar{x}} p(\bar{x}) \cdot ld \frac{1}{p(x_2|x_1)}}_{H(X_2|X_1)} + \cdots + \underbrace{\sum_{\bar{x}} p(\bar{x}) \cdot ld \frac{1}{p(x_n|x_1, x_2, \ldots, x_{n-1})}}_{H(X_n|X_1, X_2, \ldots, X_{n-1})} =$$

$$= \sum_{i=1}^{n} H(x_i | x_1, x_2, \ldots, x_{i-1})$$

Entropy of stochastic processes

Cont. **Joint Entropy** or **Block Entropy**:

$$H(\bar{X}) = \sum_{i=1}^{n} H(x_i | x_1, x_2, \ldots, x_{i-1})$$

Let us consider a ***source without memory***, i.e. the outcomes in the series (time or space) are independent from each other and stationary at least in first order.

$$H(X_1) = H(X_i) = H(X)$$

Def.: **Discrete Memoryless Source (DMS)**

$$H_{DMS}(\bar{X}) = \sum_{i=1}^{n} H(X_i) \underset{\textbf{Stationarity}}{\Longleftrightarrow} n \cdot H(X)$$

Def.: **Entropy per symbol** from Block Entropy of n symbols

$$H_n(X) = \frac{1}{n} H(\bar{X}) = \frac{1}{n} H(X_1, X_2, \ldots, X_n) \underset{DM}{\Longleftrightarrow} H(X)$$

Now let us consider the case $\boldsymbol{n} \to \infty$ i.e. **Entropy per symbol of stochastic processes $\boldsymbol{H_\infty(X)}$**.

But how to define and by what conditions exists?

Entropy of stochastic processes

How to define the **Entropy per symbol of stochastic processes** $\boldsymbol{H_\infty(X)}$

We can observe $\boldsymbol{H_\infty(X)}$ in two ways:

- As **the limit of Entropy per symbol** from Block Entropy if the block size increasing:

$$H_\infty(X) \overset{?}{\Leftrightarrow} \lim_{n\to\infty} H_n(X) = \lim_{n\to\infty} \frac{1}{n} H(X_1, X_2, \dots, X_n)$$

- OR as **the limit of conditional Entropy** of a symbol if the set size of condition symbols increasing.

$$H_\infty(X) \overset{?}{\Leftrightarrow} \lim_{n\to\infty} H(x_n | x_1, x_2, \dots, x_{n-1})$$

Proof of Gallager (1968) with 3 lemmas:

***Lemma A**:* The **conditional Entropy monotone decreasing** if the set size of condition symbols increasing: $H(x_n|x_1, x_2, \dots, x_{n-1}) \le H(x_{n-1}|x_1, x_2, \dots, x_{n-2})$

- less condition -> higher uncertainty: $H(x_n|x_1, x_2, \dots, x_{n-1}) \le H(x_n|x_2, \dots, x_{n-1})$
- n-th order stationarity: $H(x_n|x_2, \dots, x_{n-1}) = H(x_{n-1}|x_1, x_2, \dots, x_{n-2})$.

Entropy of stochastic processes

Lemma B*:* The Entropy per symbol is higher or equal to the conditional Entropy:

$$H_n(X) \geq H(x_n|x_1, x_2, \dots, x_{n-1})$$

$$H_n(X) = \frac{1}{n} H(X_1, X_2, \dots, X_n) = \frac{1}{n} \sum_{i=1}^{n} H(x_i|x_1, x_2, \dots, x_{i-1}) \geq$$

- Because A: The last term in the sum is a lower bound on each of the other term.

$$\geq \frac{1}{n} \sum_{i=1}^{n} H(x_n|x_1, x_2, \dots, x_{n-1}) = \frac{1}{n} \cdot n \cdot H(x_n|x_1, x_2, \dots, x_{n-1}) = H(x_n|x_1, x_2, \dots, x_{n-1})$$

Lemma C*:* The Entropy per symbol **monotone decreasing:** $H_n(X) \leq H_{n-1}(X)$

$$H_n(X) = \frac{1}{n} H(\bar{X}) = \frac{1}{n} H(X_1, X_2, \dots, X_n) = \frac{1}{n} [H(X_1, X_2, \dots, X_{n-1}) + H(x_n|x_1, \dots, x_{n-1})]$$

- Because B: the entropy per symbol is higher as the last term:

$$H_n(X) \leq \frac{1}{n} [(n-1) \cdot H_{n-1}(X) + H_n(X)]$$

$$n \cdot H_n(X) \leq (n-1) \cdot H_{n-1}(X) + H_n(X)$$
$$(n-1) \cdot H_n(X) \leq (n-1) \cdot H_{n-1}(X)$$

Entropy of stochastic processes

Since the **Entropy per symbol** $H_n(X)$ and the **conditional Entropy** $H(x_n|x_1,x_2,\dots,x_{n-1})$ are both nonnegative and nonincreasing with *n (Lemmas A and C)*, **both limits must exist**.

- **The limit of Entropy per symbol**: $\lim_{n\to\infty} H_n(X) = \lim_{j\to\infty} H_{n+j}(X)\ for\ any\ fixed\ n$

$$\lim_{j\to\infty} H_{n+j}(X) = \lim_{j\to\infty} \frac{1}{n+j}\left[H(X_1,X_2,\dots,X_{n-1}) + \sum_{i=n}^{n+j} H(x_i|x_1,\dots,x_{i-1})\right] \le$$

Because A: The first term in the sum is an upper bound on each of the other term.

$$\le \underbrace{\lim_{j\to\infty} \frac{1}{n+j}\cdot H(X_1,X_2,\dots,X_{n-1})}_{0} + \underbrace{\lim_{j\to\infty} \frac{j+1}{n+j}\cdot H(x_n|x_1,\dots,x_{n-1})}_{H(x_n|x_1,\dots,x_{n-1})} = H(x_n|x_1,\dots,x_{n-1})\ \forall n$$

$$\lim_{n\to\infty} H_n(X) \le \lim_{n\to\infty} H(x_n|x_1,\dots,x_{n-1})$$

- ***From Lemma B:*** $H_n(X) \ge H(x_n|x_1,x_2,\dots,x_{n-1})\ \ \forall n$

$$\lim_{n\to\infty} H_n(X) \ge \lim_{n\to\infty} H(x_n|x_1,x_2,\dots,x_{n-1})$$

The Entropy $H_\infty(X)$ of strict stationary stochastic process:

$$\boldsymbol{H_\infty(X) = \lim_{n\to\infty} H_n(X) = \lim_{n\to\infty} H(x_n|x_1,x_2,\dots,x_{n-1})}$$

Example: German text, 26 possible symbols

First order PDF in %

Symbol	%	Symbol	%
A	6.51	N	9.78
B	1.89	O	2.51
C	3.06	P	0.79
D	5.08	Q	0.02
E	17.40	R	7.00
F	1.66	S	7.27
G	3.01	T	6.15
H	4.76	U	4.35
I	7.55	V	0.67
J	0.27	W	1.89
K	1.21	X	0.03
L	3.44	Y	0.04
M	2.53	Z	1.13

Discrete random variable **X={A,B,C,….,X,Y,Z}**
Size of the event set: **n=26**
Stochastic process: series of X
Can be regarded **stationary and ergodic.**
Realization of the process: **Text**

Statistics published by **Karl Küpfmüller**

Forrás: TU Darmstadt

- Without knowledge of 1st order PDF
  The Entropy has its maximum

$$H_0(X) = ld\, n \cong 4,7 \left[\frac{bit}{symbol}\right]$$

- From the 1st order PDF

$$H(X) = H_1(X) \cong 4,1 \left[\frac{bit}{symbol}\right]$$

- Entropy per symbol from Block Entropy of 2 symbols

$$H_2(X) \cong 3,0 \left[\frac{bit}{symbol}\right]$$

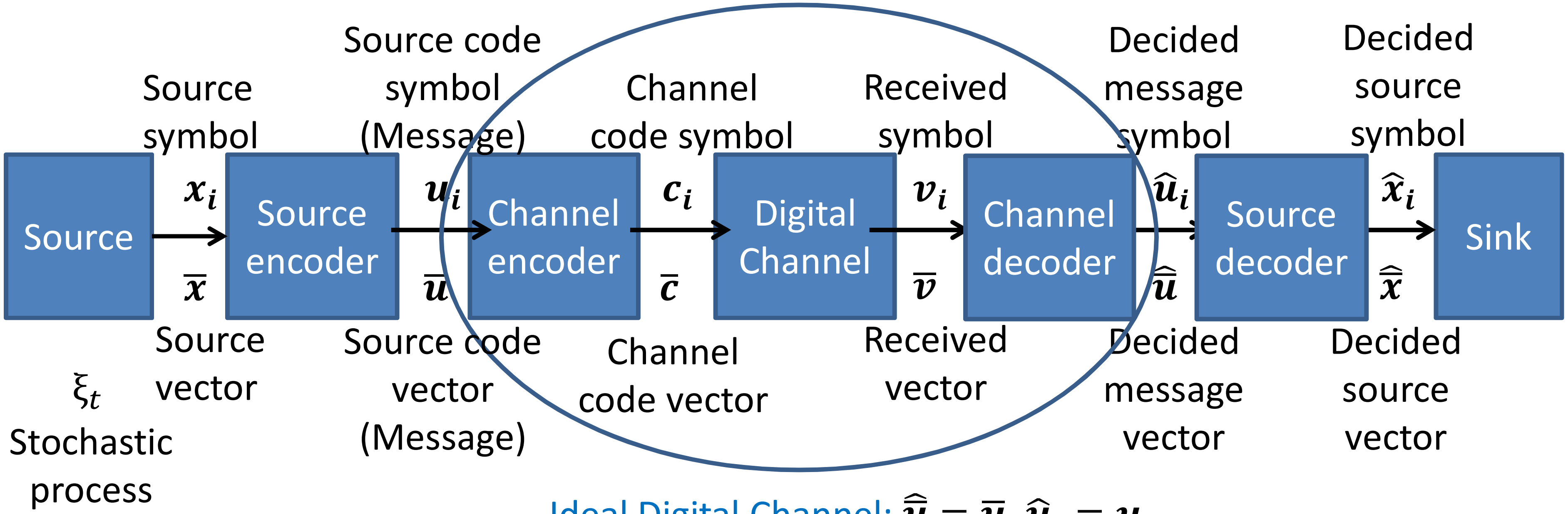
- Entropy per symbol of the process

$$H_\infty(X) \cong 1,6 \left[\frac{bit}{symbol}\right]$$

Redundancy $R(X) = H_0(X) - H_\infty(X) \cong 3,1 \left[\frac{bit}{symbol}\right]$

Source Coding

The goal of source encoding is to **reduce the redundancy**.



Ideal Digital Channel: $\hat{\overline{u}} = \overline{u}$, $\hat{u}_i = u_i$

Source encoding

- ✓ Encoding rule: $\Omega(\overline{x}) = \overline{u}$
- ✓ Explicit: $\Omega(\overline{x}_i) = \overline{u}_i \neq \Omega(\overline{x}_j) = \overline{u}_j$
- ✓ Code vectors (code words) should be **separable** from each other in a sequence of code symbols.

Decoding of source codes

- ✓ Knows the encoding rule, therefore all possible $\overline{u}$ and corresponding $\overline{x}$
- ✓ Separate the code words in a sequence of code symbols
- ✓ Decoding rule: $\Omega^{-1}(\overline{u}) = \overline{x}$

Source coding, separability

The sequence of source code symbols should be separable to code words (vectors).

We have basically 3 methods to achieve that:

- Using **fixed length code** words; each code should have the same length.
- Using a **specific symbol, a separator** to find the limits of the code words.

  Space (as separator) at different positions:

  Thisisanexampleforseparability.

  This I sane x ample for separ ability.

  This is an example for separability.

- Applying a so called **instantaneously decodable** encoding, i.e. the code word set should fulfill the **prefix** condition. Note that no code word in this case is a prefix of any other code word. Or with other formulation: not any code word is a continuation of another code word.

  ✓ **Kraft inequality**: A **necessary and sufficient** condition for the lengths of valid code words of a source code **to fulfill the prefix** condition.

Source coding, separability

Example: Consider a discrete random variable X whit n=4 possible values, PDF of X, and a binary Code symbol set U={0,1}:

$$RV: X = \{x_1, x_2, x_3, x_4\}, \qquad PDF: p(X) = \left\{p(x_1) = \tfrac{1}{2}, p(x_2) = \tfrac{1}{4}, p(x_3) = p(x_4) = \tfrac{1}{8}\right\}$$

Case	A	B	C	D	E
x_1	00	0	0	0	0
x_2	01	1	10	01	10
x_3	10	00	110	011	110
x_4	11	11	111	0111	1110

A: Fixed source code length
B: Non-separable
C: Prefix condition
D: 0 symbol separate, non prefix
E: Prefix + 0 symbol separate

Separation of code words

in a sequence of source code symbols:

0 1 0 1 1 1 0 1 0 0 1 1 0 …

A: x_2 x_2 x_4 x_2 x_1 x_4 …

B: … x_2? x_4 …

C: x_1 x_2 x_4 x_1 x_2 x_1 x_3 …

D: x_2 x_4 x_2 x_1 x_3 …

E: x_1 x_2 x_4 x_2 x_1 x_3 …

Entropy H(X)=1.75 [Shannon/symbol]

Average of code length **L**

i.e. number of binary digits in average

L=2 [bit/symbol]

L=1.75 [bit/symbol]

L=1.875 [bit/symbol]

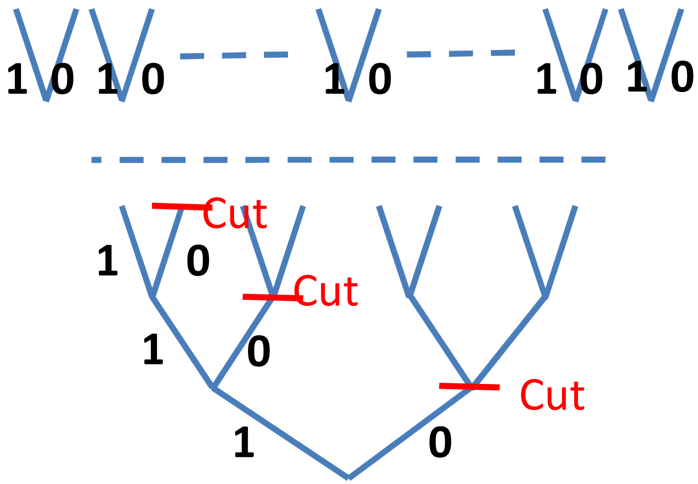
L=1.875 [bit/symbol]

Source coding, **Kraft inequality**

Kraft inequality: A **necessary and sufficient** condition for the lengths of valid code words of a source code **to fulfill the prefix** condition.

Design a prefix binary source code with **N possible code words**.

- Consider a **binary tree** with **M levels**.
- Regard the symbols of a code word along the branches of the tree.
- We should cut the tree by each code word ending.



Level	Number of endings	Length l_i of the code word
M	2^M	$l_i = M$
M-1	2^{M-1}	$l_i = M - 1$
…	…	…
3rd	$2^3 = 8$	$l_i = 3$
2nd	$2^2 = 4$	$l_i = 2$
1st	$2^1 = 2$	$l_i = 1$
Root		

Kraft inequality:

$$\sum_{i=1}^{N} 2^{M-l_i} \leq 2^M \rightarrow \sum_{i=1}^{N} \mathbf{2^{-l_i} \leq 1} \text{ for } \boldsymbol{binary} \quad \text{and} \quad \sum_{i=1}^{N} \boldsymbol{r^{-l_i} \leq 1} \text{ for } \boldsymbol{r - ary\ code\ symbols}$$

Source coding, **Kraft inequality**

Kraft's inequality:

$$\sum_{i=1}^{N} \mathbf{2^{-l_i}} \leq \mathbf{1} \text{ for } \boldsymbol{binary} \quad \text{and} \quad \sum_{i=1}^{N} \boldsymbol{r^{-l_i}} \leq \mathbf{1} \text{ for } \boldsymbol{r - ary\ code\ symbols}$$

Using Kraft's inequality, we can also characterize redundancy in prefix codes.

Definitions:

- A prefix code satisfying Kraft's inequality with strict inequality ($\sum_{i=1}^{N} \mathbf{2^{-l_i}} < \mathbf{1}$) is called **redundant**.
- A prefix code satisfying Kraft's inequality with strict equality ($\sum_{i=1}^{N} \mathbf{2^{-l_i}} = \mathbf{1}$) is called **complete**.
- The **prefix redundancy** is $1 - \sum_{i=1}^{N} \mathbf{2^{-l_i}}$

Theorem: For any redundant prefix code with code word lengths l_1, l_2,…, l_σ there exists a complete prefix code with word lengths m_1, m_2,…, m_σ such that $m_i \leq l_i$ for all $i \in [1..\sigma]$

Proof: Assume $\boldsymbol{l_\sigma}$ **is the longest**, then $\mathbf{2^{-l_i} = 2^{z_i} \cdot 2^{-l_\sigma}}$ (e.g. $2^{-3} = 2 \cdot 2^{-4}$) and the redundancy gap

$$\mathbf{1} - \sum_{i=1}^{N} \mathbf{2^{-l_i}} = \mathbf{1} - \mathbf{2^{-l_\sigma}} \cdot \sum_{i=1}^{N} \mathbf{2^{z_i}} = \mathbf{2^{l_\sigma} \cdot 2^{-l_\sigma}} - \mathbf{2^{-l_\sigma}} \cdot \sum_{i=1}^{N} \mathbf{2^{z_i}} = \mathbf{2^{-l_\sigma}} \cdot \left(\mathbf{2^{l_\sigma}} - \sum_{i=1}^{N} \mathbf{2^{z_i}} \right)$$

The gap is a multiple of $\mathbf{2^{-l_\sigma}}$ too. We reduce l_σ by one bit.

Source coding, Classification

- Basically we have **four types** of source codes according to the **length of source word** (vector of source symbols) and the **length of code word** (vector of code symbols) are fixed or variable.

		length of source word k		**Known PDF a-priori**
		Fixed	Variable	
length of code word l	Fixed	Type I: Without considering redundancy ASCII	Type III: Lempel-Ziv code	NO
	Variable	Type II: Shannon-Fano code, Huffman code	Type IV: Arithmetic code (Shannon)	YES

Type I: Not really a compressing code, it is rather a mapping
Types II, III, IV: Achieve compression, Entropy coding