3. elements of classical information theory

3.1 measure of information: Shannon entropy and classical data compression

What is information?

Example: throwing dice

- if a dice player tells you that he has thrown a "5" this is only information for you if you did not watch him throwing the dice
- if he says he has thrown a number between 1 and 6 this has also no information for you

Information is a measure of (a priori) ignorance

Let $A$ be a random event with probability $p(A)$.
$\overline{A}$ is called the complementary event of $A$ if $p(\overline{A}) = 1 - p(A)$

A set of events $\{A_1, \ldots, A_n\}$ is called complete, if
$$\sum_{i=1}^{n} p(A_i) = 1$$

Def: $\{A_1, \ldots, A_n\}$ complete set of events $A_i$ are called elementary events if

(1) $A_i \Rightarrow A_j \quad \forall j \neq i$

The set of all elementary events is called space of events $\Omega$
principle of maximal ignorance

If nothing is known about the possible outcome of a random event, the a priori probability of every elementary event $A_i$ is the same

$$P(A_i) = \frac{1}{n} \quad \forall A_i; \quad i=1,\ldots,n$$

quantitative measure of ignorance (information) $S$

$n$ discrete events

(i) $S$ is a continuous function of the probabilities $p_i$:

$$S = S(p_1, \ldots, p_n)$$

(ii) $S=0$ for a certain event (i.e. $p_j=1$ for $j$)

(iii) $S=\max$ for maximal ignorance (i.e. all $p_i=\frac{1}{n}$) and monotonously increasing with $n$

(iv) $S$ should fulfill the composition rule which guarantees that it behaves as an extensive variable

\[ W_1 = p_1 + p_2 + p_3 \]

\[ W_2 = p_4 + p_5 \]
(3) \( S(p_1, p_5) = S(w_1, w_2) + w_1 S\left(\frac{p_1}{w_1}, \frac{p_2}{w_1}, \frac{p_3}{w_1}\right) + w_2 S\left(\frac{p_4}{w_2}, \frac{p_5}{w_2}\right) \)

i.e. total information =

(i) information if "w_1 or w_2"

+ (ii) weighted information

"if w_1" then "p_1 or p_2 or p_3"
"if w_2" then "p_4 or p_5"

\[ \Rightarrow \quad \text{Shannon Entropy} \]

(4) \[ S(p_1, \ldots, p_n) = -\alpha \sum_{j=1}^{n} p_j \ln p_j \quad \alpha > 0 \]

**Remark:** the prefactor \( \alpha \) can be chosen arbitrarily.

Often \( \alpha = 1 / \ln 2 \) \( \alpha \ln p_j \rightarrow \log_2 p_j = \log p_j \).

**proof of (i)-(iv):**

(i) \( \checkmark \)

(ii) \( \checkmark \) since \( p_i = 1 \) \( \Rightarrow \) \( p_i = 0 \) \( i \neq j \)

and \( 0 \ln 0 = 0 \)

(iii) for this we first prove if \( (p_1, \ldots, p_n) \) and \( (\overline{p_1}, \ldots, \overline{p_n}) \) are two (different) probabilities of elementary events it holds

(5) \[ \sum_{j=1}^{n} p_j \ln \frac{\overline{p_j}}{p_j} \geq 0 \]

equality holds for \( p_i = \overline{p_i} \).
proof of (5): \( \ln x \leq x - 1 \quad \forall x \geq 0 \)

\[ \Rightarrow \ln \frac{1}{x} = -\ln x \geq 1 - x \]

\[ \Rightarrow \ln \frac{p_i}{p_i} \geq 1 - \frac{p_i}{p_i} \]

\[ \Rightarrow \sum_i p_i \ln \frac{p_i}{p_i} \geq \sum_i p_i - \sum p_i = 0 \quad \square \]

From (5) with \( p_i = \frac{1}{n} \) \( \Rightarrow \)

\[ \sum_{i=1}^{n} p_i \left( \ln p_i - \lambda \ln p_i \right) \geq 0 \]

\[ S(p_1, \ldots, p_n) = -\alpha \sum_{i=1}^{n} \ln p_i \leq -\alpha \sum_{i=1}^{n} \frac{p_i}{\ln p_i} \]

\[ = \alpha \ln n \quad \square \]

(iv) \( W_1 = p_1 + \ldots + p_m \)

\[ W_2 = p_{m+1} + \ldots + p_n \]

\[ S(w_1, w_2) = -\alpha w_1 \ln w_1 - \alpha w_2 \ln w_2 \]

\[ w_1 S\left( \frac{p_1}{w_1}, \ldots, \frac{p_m}{w_1} \right) = -\alpha w_1 \sum_{i=2}^{m} \frac{p_i}{w_2} \ln \left( \frac{p_i}{w_1} \right) \]

\[ = -\alpha \sum_{i=2}^{m} p_i \ln p_i + \alpha w_1 \ln w_1 \]

\[ \Rightarrow S(w_1, w_2) + w_1 S\left( \frac{p_1}{w_1}, \ldots, \frac{p_m}{w_1} \right) + w_2 S\left( \frac{p_{m+1}}{w_2}, \ldots, \frac{p_n}{w_2} \right) \]

\[ = -\alpha w_1 \ln w_1 - \alpha w_2 \ln w_2 \]

\[ -\alpha \sum_{i=2}^{m} p_i \ln p_i + \alpha w_1 \ln w_1 \]

\[ -\alpha \sum_{i=m+1}^{n} p_i \ln p_i + \alpha w_2 \ln w_2 = S(p_1, \ldots, p_n) \quad \square \]
the smallest non-trivial eventspace has two alternatives

\[ \{A, \bar{A}\} \subseteq \{0, 1\} \]

this space is called a classical bit or c-bit

for a c-bit holds \( a = \sqrt{2} \)

\[ \sum (p, 1-p) = H(p) = -p \log p - (1-p) \log (1-p) = H(1-p) \]

classical information is transferred as strings of c-bits so called bit-words

\[ x_1, x_2, \ldots, x_n \quad x_i \in \{0, 1\} \]

The relevance of the Shannon entropy can be seen most easily from the following problem: Given a bitword of length \( n \) with \( n \to \infty \), how many c-bits are needed to transfer the bitword from \( A \) to \( B \)?
Shannon's noiseless coding theorem states that to transfer an n-bit word if it is sufficient in the limit $n \to \infty$ to send $nH(p)$ bits, where $p$ is the probability of a "1". However, Shannon's theorem does not provide an algorithm to construct this code.

Example (speak for better reading):

<table>
<thead>
<tr>
<th>Code</th>
<th>0000</th>
<th>0010</th>
<th>0100</th>
<th>0110</th>
<th>1000</th>
<th>1010</th>
<th>1001</th>
<th>1011</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0000</td>
<td>0010</td>
<td>0100</td>
<td>0110</td>
<td>1000</td>
<td>1010</td>
<td>1001</td>
<td>1011</td>
</tr>
<tr>
<td></td>
<td>0100</td>
<td>0110</td>
<td>1000</td>
<td>1010</td>
<td>0000</td>
<td>0010</td>
<td>0100</td>
<td>0110</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>1010</td>
<td>0000</td>
<td>0010</td>
<td>0100</td>
<td>0110</td>
<td>1000</td>
<td>1010</td>
</tr>
</tbody>
</table>

The answer is given by $p(\omega)$.
<table>
<thead>
<tr>
<th>Original</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>000</td>
</tr>
<tr>
<td>100</td>
<td>0100</td>
</tr>
<tr>
<td>010</td>
<td>0010</td>
</tr>
<tr>
<td>110</td>
<td>1000</td>
</tr>
<tr>
<td>0001</td>
<td>0101</td>
</tr>
<tr>
<td>101</td>
<td>1100</td>
</tr>
<tr>
<td>111</td>
<td>1010</td>
</tr>
<tr>
<td>011</td>
<td>0001</td>
</tr>
<tr>
<td>1001</td>
<td>0110</td>
</tr>
<tr>
<td>1101</td>
<td>1001</td>
</tr>
<tr>
<td>0011</td>
<td>0111</td>
</tr>
<tr>
<td>1011</td>
<td>1101</td>
</tr>
<tr>
<td>1111</td>
<td>1011</td>
</tr>
<tr>
<td>0111</td>
<td>0011</td>
</tr>
<tr>
<td>1000</td>
<td>0100</td>
</tr>
<tr>
<td>1100</td>
<td>1000</td>
</tr>
</tbody>
</table>

Huffman code (here for $p(0) = 3/4$ and $p(1) = 1/4$)

Concatenate a code that can be used without space in the text used. A code that was cut is changing an space.
3.2 Communication channels

Shannon's noiseless coding theorem considers information transfer through ideal channels. In reality, communication channels have losses and are subject to noise.

\[ \begin{array}{ccc}
\text{Alice} & \text{binary channel} & \text{Bob} \\
X & \rightarrow & Y \\
x \in \text{fa}\_\text{i}s & \rightarrow & y \in \text{fa}\_\text{i}s \\
\end{array} \]

An important class of channels are memoryless channels (Markovian channels).

- Symmetric binary channel

\[ \begin{array}{ccc}
\text{A} & \text{B} \\
1 & 1 \\
0 & 0 \\
\end{array} \]

Equal error probability \( p = P_{\text{error}} \)
\( 0 \rightarrow 1 \) and \( 1 \rightarrow 0 \)

- Binary decay channel

\[ \begin{array}{ccc}
\text{A} & \text{B} \\
1 & 1 \\
0 & 0 \\
\end{array} \]

Transformation : \( P_{ij} \) (\( p = P_{\text{error}} \))

\[ \begin{pmatrix}
1-p & p \\
p & 1-p \\
\end{pmatrix} \] (Symmetric)

\[ \begin{pmatrix}
1 & p \\
0 & 1-p \\
\end{pmatrix} \] (Decay)
An important quantity to characterize channels is the **mutual information**

\[ I(X;Y) = S(X) - S(X|Y) = S(Y) - S(Y|X) \]

Where

\[ S(Y|X) = - \sum_{j} p(x_j) \sum_{e} p(y | x_j) \log \left( \frac{p(y | x_j)}{e} \right) \]

\[ = \sum_{e} p(x_j, y_e) \log \left( \frac{p(y | x_j)}{e} \right) \]

is the conditional Shannon entropy.

\[ p(x_j, y_e) \] joint probability of \( x = x_j \) and \( y = y_e \)

\[ p(y | x_j) \] conditional probability of \( y = y_e \) given that \( x = x_j \) is true

If holds

\[ p(x | y) = p(y | x) = p(x | y) p(y) = p(y | x) p(x) \]

if \( x \) and \( y \) are independent, then

\[ p(x | y) = p(x) \Rightarrow p(x, y) = p(x) p(y) \]

One can write the mutual information in the form

\[ I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \]

J.E. if \( x \) and \( y \) are independent
(14) \[ I(X;Y) = 0 \quad \text{and} \quad S(X | Y) = S(X) \]

Thus for independent events there is no information in the knowledge of one about the other event.

For a binary symmetric channel

(15) \[ I(X;Y) = H(p_{xy}) - H(p_{raw}) \]

Def: channel capacity

(16) \[ C = \max_{p(x)} I(X;Y) \quad \text{Y-output} \]

\[ X-\text{input} \]

\( \Rightarrow \) binary symmetric channel

(17) \[ C = 1 - H(p_{raw}) \]

Now let us ask the question how many bits are needed to transfer an \( n \)-bit word through a noisy channel.

Transfer rate \( R \)

\[ R = \frac{\# \text{bits in message}}{\# \text{bits in channel}} \quad 0 \leq R \leq 1 \]

Assume error probability \( p_b \) of channel

If \( R = 1 \), total error probability \( e = p_b < 1 \)
to reduce error probability use code

\[(g) \quad 0 \rightarrow 000 \quad 1 \rightarrow 111 \quad \text{(repetition code)}\]

now \( R = \frac{1}{3} \). If only one of the code bits has an error, the message can still be reproduced exactly, i.e., the probability of error (two or more code bits have an error)

\[(20) \quad e = p_b^3 + 3p_b^2(1-p_b) < p_b\]

\[\text{m-fold repetition} \quad e \rightarrow 0 \quad \text{🙂} \quad R \rightarrow 0 \quad \text{🙁}
\]

Can one do better? Yes!

**Shannon's noisy coding theorem (1948)**

For any channel with capacity \( C \), there is a code that allows a data transfer with arbitrary small error probability and \( 0 < R < C \)

Now we will discuss how data transfer can be achieved through a noisy channel with non-vanishing transfer rate
3.3 Classical error correction

Within the last 60 years an extensive theory of error correcting codes has been developed. We will now shortly discuss an important special class: binary, linear codes

binary code \( k \) codewords encoded in \( n > k \) codewords

- \# of encodable words \( 2^k \)
- \# of words \( 2^n \)

linear binary code

Set \( C \) of codewords which form a Galois field (endlicher Körper) with respect to bitwise addition modulo 2

Example: \( C = \left\{ (0,1,1), (1,0,1), (1,1,0), (0,0,0) \right\} \)

\((0,1,1) \oplus (1,0,1) = (1,1,0) \in C\)
\((0,1,1) \oplus (1,1,0) = (1,0,1) \in C\)
\((1,0,1) \oplus (1,1,0) = (0,1,1) \in C\)

\( C \) has \( n = 3 \) and \( k = 2 \) basis e.g. \( (0,1,1) \)

basis of \( C \)

\( \{ u_1, u_2, \ldots, u_k \} \)

\( u_i = \) codeword consisting of \( n \geq k \) codewords
Every codeword can be written as a superposition of \( \xi_i \): 

\[
\nu = \bigoplus_{j=1}^{k} \xi_j \nu_j, \quad \xi_j \in \{0,1\}
\]

(bitwise addition modulo 2)

\( \nu \) encodes the \( k \)-bit message \((\alpha_1, \alpha_2, \ldots, \alpha_k)\).

The \( k \) basis vectors form the \( k \times n \) generator matrix.

\[
G = \begin{bmatrix}
\nu_1 \\
\vdots \\
\nu_k
\end{bmatrix}
\]

In our example above:

\[
G = \begin{pmatrix}
0 & 1 & 1 \\
1 & 0 & 1
\end{pmatrix}
\]

Then eq. (21) can be written as:

\[
\nu = \alpha \cdot G, \quad \alpha = (\alpha_1, \alpha_2, \ldots, \alpha_k)
\]

Alternatively, the code can be characterized by the \( n-k \) constraints:

\[
H \nu^T = 0
\]

(we leave "T" out for simplicity)

\( H \) is an \((n-k) \times n\) matrix called the parity-check matrix.

The rows of \( H \) are \((n-k) \) linearly independent vectors which are orthogonal to all codewords.
In our example

(26) \( \mathbf{H} = (1, 1, 1) \)

basis was \( \{(0,1,1), (1,0,1)\} \)

\[
(0,1,1) \cdot (1,1,1) = 0 \oplus 1 \oplus 1 = 0 \quad (\oplus \text{mod } 2)
\]

\[
(1,0,1) \cdot (1,1,1) = 1 \oplus 0 \oplus 1 = 0
\]

code:

\[
\vec{x} = (0,0) \rightarrow (0,0,0)
\]

(e.g.,

\[
\begin{align*}
\vec{x} &= (1,0) \rightarrow (0,1,1) \\
\vec{x} &= (0,1) \rightarrow (1,0,1) \\
\vec{x} &= (1,1) \rightarrow (1,1,0)
\end{align*}
\]

The parity-check matrix can be used to detect a possible error. For a classical channel, the only possible error is a spin flip. A spin-flip error can be represented as addition with an error vector \((\text{mod } 2)\)

(27) \( \mathbf{e} \rightarrow \mathbf{u} \oplus \mathbf{e} \)

Where \( \mathbf{e} \) has entries "1" at the places where an error happens, e.g.,

(28) \( \mathbf{e} = (1,0,1,0,0,0,\ldots) \)

corresponds to a spin-flip error of first and third bit.
Detection of an error

(29) $H(u \oplus e) = H_u \oplus H_e = H_e \Rightarrow u$

$H_e$ is the error syndrome.

The set $E$ of all errors $f \in E$ which have different error syndromes can be corrected. Once the error has been unambiguously detected it is corrected by adding the same error for a second time.

(30) $(u \oplus e_1) \oplus e_1 = u$

If however $H_e_1 = H_e_2$ we could misinterpret the error syndrome and correct wrongly.

(31) $u \oplus e_1 \rightarrow (u \oplus e_1) \oplus e_2 \neq u$

Hamming distance

The Hamming distance of a code is the minimum number of bits that need to be changed to transfer one element of the code to another one.

A linear, binary code of Hamming distance $d = 2t + 1$ can correct $t$ bit flip errors.
the code introduced above has $d=2$ and can thus not correct any error.

$e_1 = (1,0,0) \quad e_2 = (0,1,0)$

\[ u_1 \oplus e_1 = (0,1,1) \oplus (1,0,0) = (1,1,1) \]
\[ = (1,0,1) \oplus (0,1,0) = u_2 \oplus e_2 \]

Here, $d=2$.

For $d=3$:

\[ u_1 \quad u_2 \]
\[ d=3 \]
Example of an \([n=7, k=4, d=3]\) code

<table>
<thead>
<tr>
<th>Original</th>
<th>Hamming ([n=7, k=4, d=3]) code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>0000000</td>
</tr>
<tr>
<td>0001</td>
<td>1010101</td>
</tr>
<tr>
<td>0010</td>
<td>0110011</td>
</tr>
<tr>
<td>0011</td>
<td>1100110</td>
</tr>
<tr>
<td>0100</td>
<td>0001111</td>
</tr>
<tr>
<td>0101</td>
<td>1011010</td>
</tr>
<tr>
<td>0110</td>
<td>0111100</td>
</tr>
<tr>
<td>0111</td>
<td>1101001</td>
</tr>
<tr>
<td>1000</td>
<td>1111111</td>
</tr>
<tr>
<td>1001</td>
<td>0101010</td>
</tr>
<tr>
<td>1010</td>
<td>1001100</td>
</tr>
<tr>
<td>1011</td>
<td>1110000</td>
</tr>
<tr>
<td>1101</td>
<td>0100101</td>
</tr>
<tr>
<td>1110</td>
<td>1000011</td>
</tr>
<tr>
<td>1111</td>
<td>0010110</td>
</tr>
</tbody>
</table>

The \([n=7, k=4, d=3]\) code allows to correct all 1-bit errors in an \(n=7\) bit code.