3. elements of classical information theory

3.1 measure of information, Shannon entropy and classical data compression

▶ What is information?

<u>example:</u> 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

$\boxed{\text{information is a measure of (a priori) ignorance}}$

Let A be a random event with probability $p(A)$.
$\bar{A}$ is called the <u>complementary event of A</u>, if $p(\bar{A}) = 1 - p(A)$

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

<u>Def:</u> $\{A_1, \dots, A_n\}$ complete set of events
A_i are called <u>elementary events</u> if

(1) $$A_i \Rightarrow \bar{A}_j \quad \forall j \neq i$$

The set of all elementary events is called <u>space of events</u>: Ω

principle of maximal ignorance

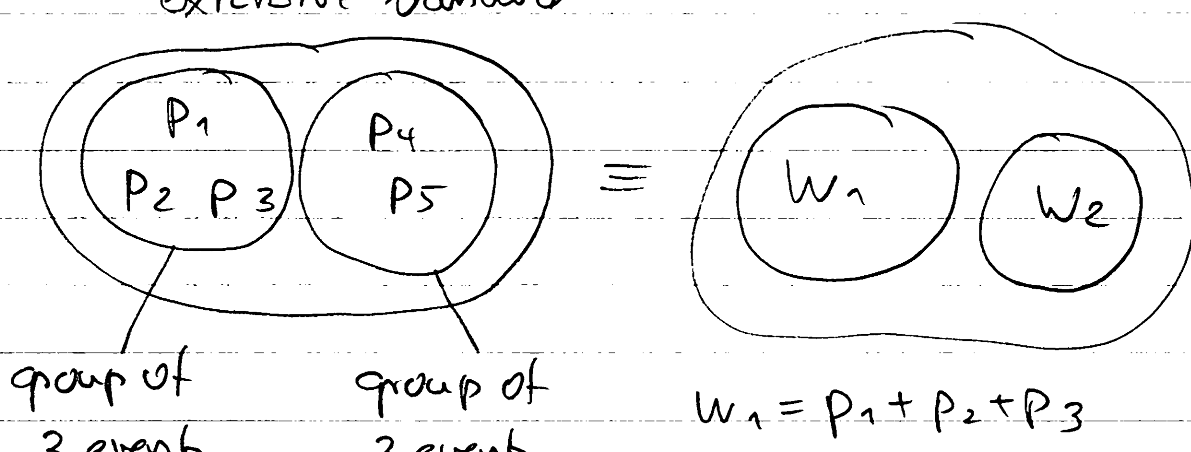
If nothing is known about the possible out-come of a random event, the a priori probability of every elementary event A_i is the same

(2) $$p(A_i) = \frac{1}{n} \qquad \forall A_i \quad i=1,..,n$$

quantitative measure of ignorance (information) S

n discrete events

(i) S is a continuous function of the probabilities p_i
$$S = S(p_1, .., p_n)$$
(ii) $S=0$ for a certain event (i.e. $p_j = 1$ for one 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"

$\Rightarrow$ Shannon Entropie

(4) $$\boxed{S(p_1, \dots, p_n) = -a \sum_{j=1}^{n} p_j \ln p_j} \qquad a > 0$$

rem: the prefactor a can be chosen arbitrarily

often $a = 1/\ln 2$ $\quad a \ln p_j \to \log_2 p_j = \log p_j$

proof of (i)–(iv):

(i) ✓

(ii) ✓ since $p_j = 1 \Rightarrow p_i = 0 \quad i \neq j$

and $0 \ln 0 = 0$

(iii) for this we first proof if $(p_1, .. p_n)$ and $(\bar{p}_1, .., \bar{p}_n)$ are two (different) probabilities of elementary events it holds

(5) $$\sum_{i=1}^{n} \bar{p}_i \ln \frac{\bar{p}_i}{p_i} \geq 0$$

equality holds for $p_i = \bar{p}_i$

proof of (5): $\ln x \leq x-1 \quad \forall x \geq 0$

$$\Rightarrow \ln 1/x = -\ln x \geq 1-x$$

$$\Rightarrow \ln \frac{\bar{p}_i}{p_i} \geq 1 - \frac{p_i}{\bar{p}_i}$$

$$\Rightarrow \sum_i \bar{p}_i \ln \frac{\bar{p}_i}{p_i} \geq \sum_i \bar{p}_i - \sum_i p_i = 0 \quad \square$$

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

$$\sum_{i=1}^{n} \bar{p}_i \left(\ln \bar{p}_i - \ln p_i \right) \geq 0$$

$$S(\bar{p}_1, .., \bar{p}_n) = -a \sum_{i=1}^{n} \bar{p}_i \ln \bar{p}_i \leq -a \sum_{i=1}^{n} \bar{p}_i \ln \frac{1}{n}$$

$$= a \ln n \quad \square$$

(iv) $W_1 = p_1 + \cdots + p_m$
$W_2 = p_{m+1} + \cdots + p_n$

$$S(W_1, W_2) = -a W_1 \ln W_1 - a W_2 \ln W_2$$

$$W_1 S\left(\frac{p_1}{W_1}, .., \frac{p_m}{W_1}\right) = -a \cancel{W_1} \sum_{i=1}^{m} \frac{p_i}{\cancel{W_1}} \ln \left(\frac{p_i}{W_1} \right)$$

$$= -a \sum_{i=1}^{m} p_i \ln p_i + a W_1 \ln W_1$$

$$\Rightarrow S(W_1, W_2) + W_1 S\left(\frac{p_1}{W_1}, .., \frac{p_m}{W_1}\right) + W_2 S\left(\frac{p_{m+1}}{W_2}, .., \frac{p_n}{W_2}\right)$$

$$= \cancel{-a W_1 \ln W_1} \cancel{- a W_2 \ln W_2}$$
$$- a \sum_{i=1}^{m} p_i \ln p_i \cancel{+ a W_1 \ln W_1}$$
$$- a \sum_{i=m+1}^{n} p_i \ln p_i \cancel{+ a W_2 \ln W_2} = \underline{\underline{S(p_1, .., p_n)}} \quad \square$$

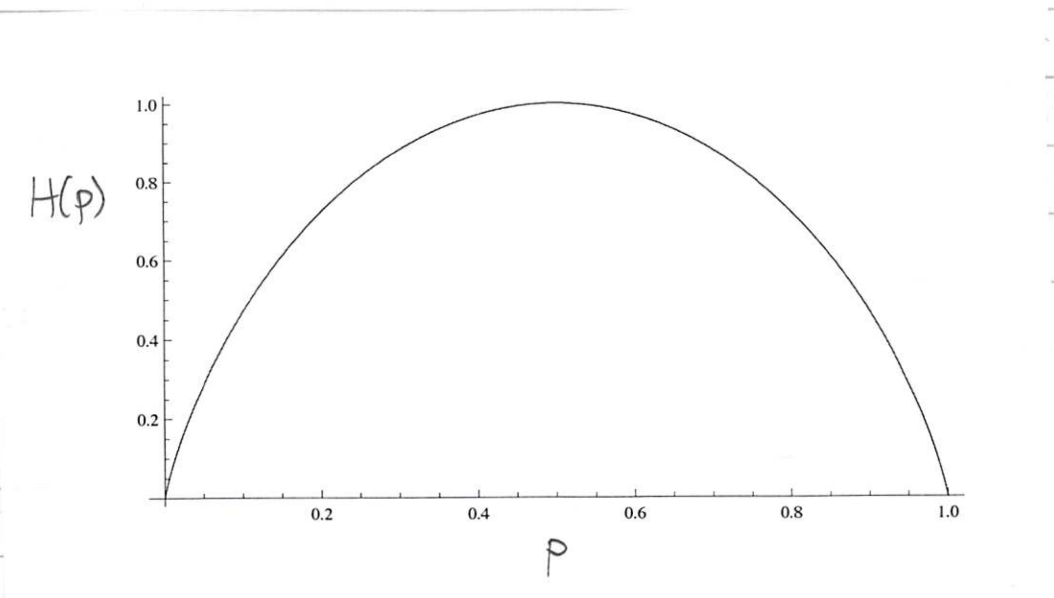
the smallest non-trivial eventspace has two alternatives

$$\{A, \bar{A}\} \hat{=} \{0, 1\}$$

$$\boxed{\text{this space is called a } \underline{\text{classical bit}} \text{ or } \underline{\text{cbit}}}$$

for a c-bit holds ($a = 1/\ln 2$)

$$(6) \quad \boxed{S(p, 1-p) = H(p) = -p \log p - (1-p) \log (1-p) = H(1-p)}$$



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

$$(7) \qquad x_1, x_2, \ldots, x_n \qquad x_i \in \{0, 1\}$$

The relevance of the Shannon entropy can be seen most easily from the following problem: Given a bit word of length n with $n \to \infty$. How many cbits are needed to transfer the bitword von A to B?

$x_1, x_2, \dots, x_n$ $\quad\quad$ $m \leq n$? $\quad\quad$ $x_1, x_2 \dots, x_n$

$$\text{Alice} \xrightarrow{x_1, \dots, x_m} \text{Bob}$$

the answer is given by

Shannon's noiseless coding theorem

to transfer an n c-bit word it is sufficient in the limit $n \to \infty$ to send $nH(p)$ cbits, where p is the probability of a "0" (or a "1", note $H(p) = H(1-p)$) in the n bit word

also: there is a unique code containing $1 + nH(p)$ bits! However Shannon's theorem does not provide an algorithm to construct this code

example (spaces for better reading)

```
0000 0001 0100 0101 0000 0110 0100 1000 0000 0011
1000 0101 1100 0001 0100 0010 0000 0101 0001 0110
0110 0001 0100 1100 0000 0001 0100 0101 0000 0101
1001 0100 0010 0000 0101 0001 0000 0011 0001 0100
0100 1100 0001 0000 0011 0001 1100 0100 0100 0010
0100 0101 0000 0101 0100 0010 0000 0101 0001 0100
1000 0101 1100 1101 0100 0010 0000 0101 0001 0100
0000 0101 0001 0100 1000 0101 1100 0001 1100 1000
```

$p(0) = \frac{225}{320} = 0{,}703$

$p(1) = \frac{95}{320} = 0{,}297$

$p(0) > p(1)$

code:

```
10 000 010 1001 10 1010 010 011 10 1000
011 1001 1101 000 010 001 10 1001 000 1010
1010 000 010 1101 10 000 010 1001 10 1001
1011 010 001 10 1001 000 10 1000 000 010
010 1101 000 10 1000 000 1101 010 010 001
010 1001 10 1001 010 001 10 1001 000 010
011 1001 1101 11110 010 001 10 1001 000 010
10 1001 000 010 011 1001 1101 000 1101 011
```

obviously shorter!

original	code
0000	10
0001	000
0010	001
0100	010
1000	011
0011	1000
0101	1001
1001	1011
0110	1010
1010	1100
1100	1101
0111	11000
1011	11111
1101	11110
1110	11101
1111	111001

code prefers
zeroth

The code used in the example was abit cheating as spaces were used. A code that can be used without spaces is the **Huffmann code** (here for $p(0) = 3/4$ and $p(1) = 1/4$)

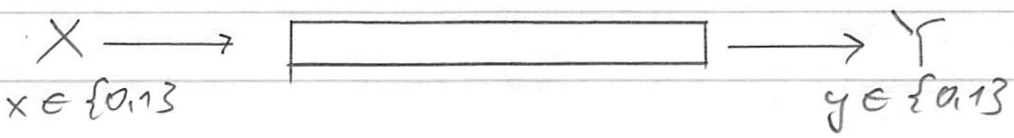
original	code
0000	10
0001	000
0010	001
0100	010
1000	011
0011	11000
0101	11001
0110	11010
1001	11011
1010	11100
1100	11101
0111	111000
1011	111111
1101	111110
1110	111101
1111	1111001

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.

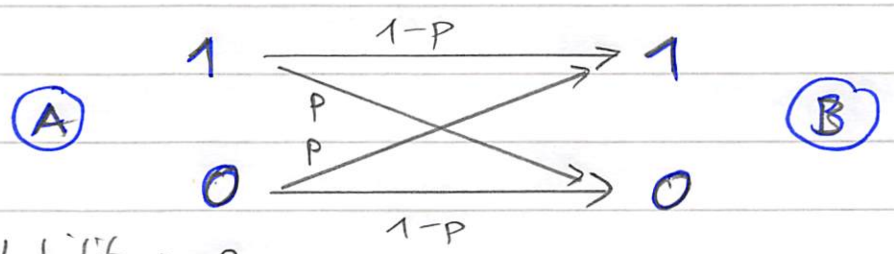
<u>binary channel</u>

Alice Bob



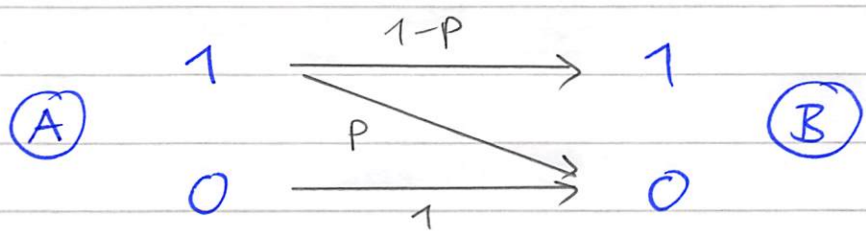
an important class of channels are <u>memoryless channels</u> (Markovian channels)

- <u>symmetric binary channel</u>



equal error probability $p = P_{error}$
$0 \to 1$ and $1 \to 0$

- <u>binary decay channel</u>



transfermatrix: p_{ij} $(p = P_{error})$

$$(8) \quad \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix} \qquad \begin{pmatrix} 1 & p \\ 0 & 1-p \end{pmatrix}$$

<u>symmetric</u> <u>decay</u>

An important quantity to characterize channels is the <u>mutual information</u>

$$(9) \qquad I(X:Y) = S(X) - S(X|Y) = S(Y) - S(Y|X)$$

where

$$S(Y|X) = -\sum_j p(x_j) \sum_e p(y_e|x_j) \log\left(p(y_e|x_j)\right)$$

$$(10) \qquad = \sum_{ej} p(x_j, y_e) \log\left(p(y_e|x_j)\right)$$

is the <u>conditional Shannon entropy</u>.

$p(x_j, y_e)$ joint probability of $x = x_j$ and $y = y_e$

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

it holds

$$(11) \qquad p(x,y) = p(y,x) = p(x|y)\,p(y) = p(y|x)\,p(x)$$

if x and y are <u>independent</u>, then

$$(12) \qquad p(x|y) = p(x) \quad \Rightarrow \quad p(x,y) = p(x)\,p(y)$$

One can write the mutual information in the form

$$(13) \qquad I(X:Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)}$$

i.e. if x and y are independent $\Rightarrow$

(14) $I(X:Y)=0$ and $S(X|Y)=S(X)$

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

for a binary symmetric channel

(15) $$\boxed{I(X:Y) = H(p(y)) - H(p_{error})}$$

<u>Def:</u> <u>channel capacity</u>

(16) $$C \equiv \max_{p(x)} I(X:Y)$$

Y - output
X - input

$\Rightarrow$ binary symmetric channel

(17) $$\boxed{C = 1 - H(p_{error})}$$

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

<u>transfer rate R</u>

(18) $$R = \frac{\#\text{ bits in message}}{\#\text{ bits in channel}} \qquad 0 \le R \le 1$$

assume error probability p_b of channel

if $R=1$ total error probabity $e = p_b < 1$

to reduce error probabity we code

(19) $\quad 0 \to 000 \qquad 1 \to 111 \qquad$ (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) $$e = p_b^3 + 3p_b^2(1-p_b) < p_b$$

m-fold repition
$$e \xrightarrow[m \to \infty]{} 0$$
$$R \xrightarrow[m \to \infty]{} 0$$

can one do better? Yes!

Shannon's noisy coding theorem (1948)

For any channel with capacity C there is a code that allows a datatransfer 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 cbits encoded in $n > k$ cbits

\# 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: $C = \{(0,1,1),(1,0,1),(1,1,0),(0,0,0)\}$

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

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

basis of C

$\{v_1, v_2, \ldots, v_k\}$ v_i = codeword consisting of $n \geq k$ cbits every codeword can be written as superposition of $\underline{v}_i$

$$(21) \qquad \underline{v} = \bigoplus_{i=1}^{k} \alpha_i \underline{v}_i \qquad \alpha_i \in \{0,1\}$$

(bitwise addition modulo 2)
$\underline{v}$ encodes the k-bit message $(\alpha_1, \alpha_2, \ldots, \alpha_k)$
The k basis vectors form the $k \times n$ generator matrix

$$(22) \qquad \underline{\underline{G}} = \begin{bmatrix} \underline{v}_1 \\ \vdots \\ \underline{v}_k \end{bmatrix}$$

in our example above

$$(23) \qquad \underline{\underline{G}} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

then eq. (21) can be written as

$$(24) \qquad \underline{v} = \underline{\alpha} \cdot \underline{\underline{G}} \qquad \underline{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_k)$$

Alternatively the code can be characterized by the n-k constraints

$$(25) \qquad \boxed{\underline{\underline{H}}\, \underline{v}^T = 0} \qquad \text{(we leave "T" out for simplicity)}$$

$\underline{\underline{H}}$ is an $(n-k) \times n$ matrix called parity-check matrix
The rows of $\underline{\underline{H}}$ are $(n-k)$ linear independent vectors which are orthogonal to all codewords.

In our example

(26) $$\underline{\underline{H}} = (1,1,1)$$

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

$$(0,1,1)\cdot(1,1,1) = 0 \oplus 1 \oplus 1 = 0 \qquad (\oplus \text{ mod } 2)$$
$$(1,0,1)\cdot(1,1,1) = 1 \oplus 0 \oplus 1 = 0$$

code: (e.g.)
$$\vec{\alpha} = (0,0) \longrightarrow (0,0,0)$$
$$\vec{\alpha} = (1,0) \longrightarrow (0,1,1)$$
$$\vec{\alpha} = (0,1) \longrightarrow (1,0,1)$$
$$\vec{\alpha} = (1,1) \longrightarrow (1,1,0)$$

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 <u>error vector</u> (mod 2)

(27) $$\underline{v} \longrightarrow \underline{v} \oplus \underline{e}$$

where $\underline{e}$ has entries "1" at the places where an error happens, e.g.

(28) $$e = (1,0,1,0,0,0,\ldots)$$

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

detection of an error

$$(29)\quad \boxed{\underline{\underline{H}}(\underline{v}\oplus\underline{e}) = \underbrace{\underline{\underline{H}}\,\underline{v}}_{=0} \oplus \underline{\underline{H}}\,\underline{e} = \underline{\underline{H}}\,\underline{e}}\quad \text{error detection}$$

$\underline{\underline{H}}\,\underline{e}$ is the error syndrom.

The set $\mathcal{E}$ of all errors $\{e_i\}$ which have different error syndroms can be corrected. Once the error has been unambiguously detected it is corrected by adding the same error for a second time

$$(30)\quad \boxed{(\underline{v}\oplus\underline{e}_i)\oplus\underline{e}_i = \underline{v}}\quad \text{error correction}$$

If however $\underline{\underline{H}}\,e_1 = \underline{\underline{H}}\,e_2$ we could misinterpret the error syndrom and correct wrongly!

$$(31)\quad \underline{v}\oplus\underline{e}_1 \longrightarrow (\underline{v}\oplus\underline{e}_1)\oplus\underline{e}_2 \neq \underline{v}$$

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 into 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 an error

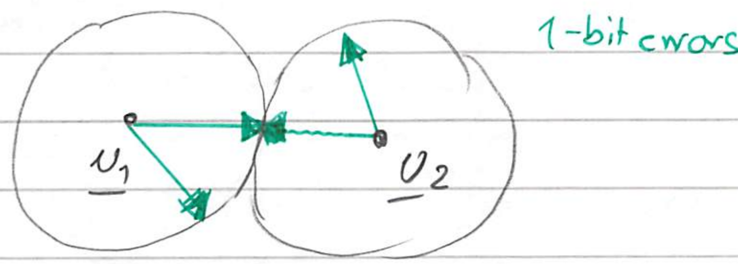
$$e_1 = (1,0,0) \qquad e_2 = (0,1,0)$$

$$\underline{v}_1 \oplus \underline{e}_1 = (0,1,1) \oplus (1,0,0) = (1,1,1)$$

(32)

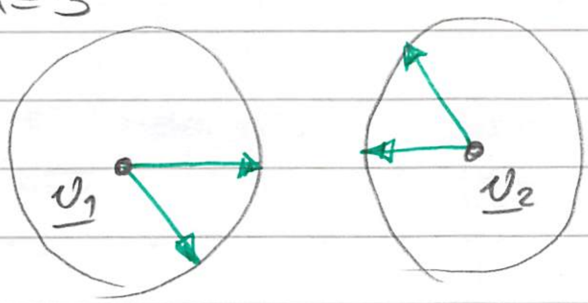
$$= (1,0,1) \oplus (0,1,0) = \underline{v}_2 \oplus \underline{e}_2$$

here



$d=2$

for $d=3$



$d=3$

example of an $[n=7, k=4, d=3]$ code

original	Hamming [n=7, k=4, d=3] code
0000	0000000
0001	1010101
0010	0110011
0011	1100110
0100	0001111
0101	1011010
0110	0111100
0111	1101001
1000	1111111
1001	0101010
1010	1001100
1011	1110000
1101	0100101
1110	1000011
1111	0010110

the $[n=7, k=4, d=3]$ code allows to correct all 1-bit errors in an $n=7$ bit code