Introduction to information theory and coding

Louis WEHENKEL

Channel coding (data transmission)

1. Intuitive introduction
2. Discrete channel capacity
3. Differential entropy and continuous channel capacity
4. Error detecting and correcting codes
5. Turbo-codes

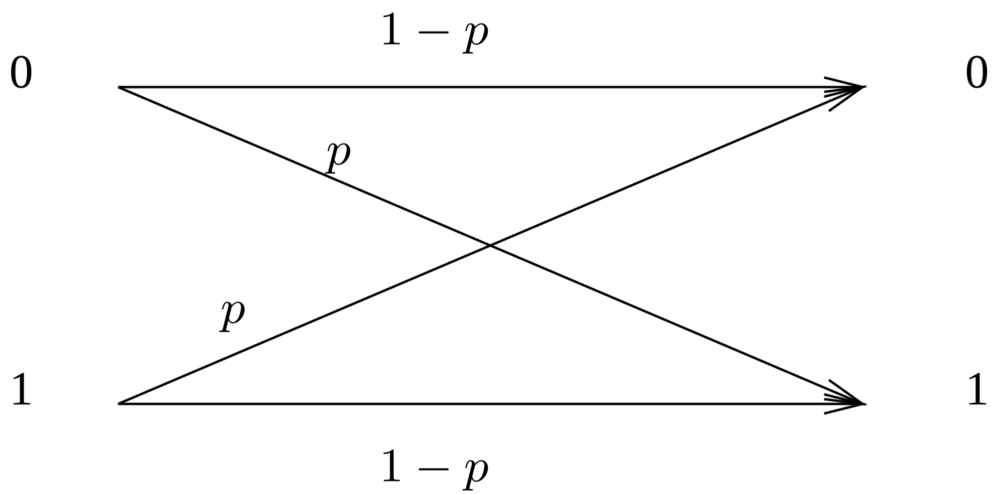
1. Intuitive introduction to channel coding

How can we communicate in reliable fashion over noisy channels ?

Examples of noisy communication channels :

1. Phone line (twisted pair : thermal noise, distortions, cross-talk…)
2. Satellite (cosmic rays…)
3. Hard-disk (read or write errors, imperfect material)

Simplistic model : binary symmetric channel (p = probability of error)



Let us suppose that : $p = 0.1$ (one error every 10 bits, on the average)

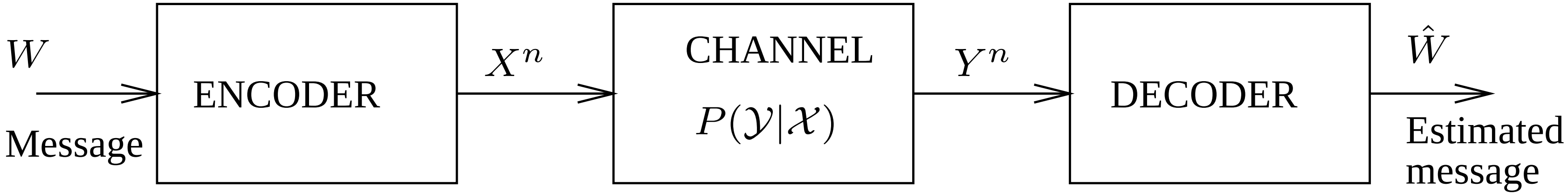
But in order to use the channel (say, a hard-disk) : we want to make sure that during the whole lifecycle of 100 disks there is no more than one error.

E.g. : life period = 10 years. And let us suppose that we transfer 1GB every day on the disk :

$\Rightarrow P_e < 10^{-15}$ (requested)

Two engineering approaches :

1. Physical aproach : better circuits, lower density, better cooling, increase signal power…
2. System approach : compensate for the bad performances by using the disk in an “intelligent” way



Information theory (and coding) ⇒ system approach (and solutions)

Add redundancy to the input message and exploit this redundancy when decoding the received message

Information theory :

Which are the possibilities (and limitations) terms of performance tradeoffs ?

⇒ analysis problem

Coding theory :

How to build practical error-compensation systems ?

⇒ design problem

(Cf. analogy with “Systems theory vs Control theory”)

Some preliminary remarks

Analysis problem : - more or less solved for most channels (but not for networks)

Design problam : - Very difficult in general
- Solved in a satisfactory way for a subset of problems only.

Cost of the system approach : - performance tradeoff, computational complexity,
- loss of real-time characteristics (in some cases).

Cost of the physical approach : - investment, power (energy).

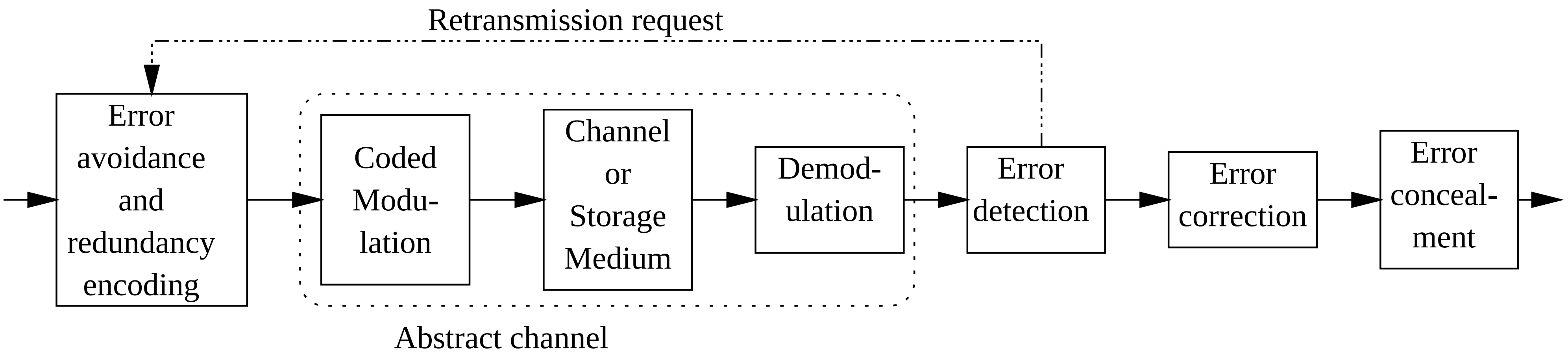
$\Rightarrow$ Tradeoff depends on application contexts

Example : (deep space communication)

Each dB of coding gain (which allows to reduce power, size . . . of antennas) leads to decrease in cost of $80.000.000 for the Deep Space Network!

$\Rightarrow$ coding can make “infeasible” projects “feasible”

Error handling systems



(Error concealment : exploits natural redundancy to interpolate “missing values”)

NB:

Error detection with retransmission request is an alternative to error correction, but is not always possible.

In some protocols, error detection merely leads to dropping packets.

⇒ We will come back later to the discussion of “error detection and retransmission” versus “forward error correction”.

Codes for detecting and/or correcting errors on the binary symmetric channel

1. Repetition codes :

Source	Code
0	000
1	111

Decoder : majority vote.

Example of transmission : $T = 0010110$.

s	0	0	1	0	1	1	0
x	000	000	111	000	111	111	000
b	000	001	000	000	101	000	000
y	000	001	111	000	010	111	000

(b: noise vector)

Decoding : $\hat{T} = 0010\underline{0}10$

P_e (per source bit) : $p^3 + 3p^2(1-p) = 0.028$ and code rate : $R = 1/3$

NB: to reach $P_e \leq 10^{-15}$ we need $R \leq 1/60 \ldots$

Other properties : correction of single errors, detection of double errors.

2. Linear block codes (Hamming $(7,4)$)

We would like to maximize the code rate under the reliability constraint $P_e \leq 10^{-15}$

Block codes : to a block of K source bits we associate a codeword of length $N \geq K$.

Example : Hamming $(7,4)$

s	x	s	x	s	x	s	x
0000	0000000	0100	0100110	1000	1000101	1100	1100011
0001	0001011	0101	0101101	1001	1001110	1101	1101000
0010	0010111	0110	0110001	1010	1010010	1110	1110100
0011	0011100	0111	0111010	1011	1011001	1111	1111111

This code may be written in compact way by (s and x denote line vectors)

$$x = s\boldsymbol{G} \quad \text{with} \quad \boldsymbol{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix} = [\boldsymbol{I}_4 \boldsymbol{P}]$$

⇒ **linear** block code (additions and multiplications modulo 2)

Definition (linear code) We say that a code is linear if all linear combinations of codewords are also codewords.

Binary codewords of length n form an $n-$dimensional linear space. A linear code consists of a linear subspace of this space.

In our example ⇒ first 4 bits = source word, last 3 bits = parity control bits.
E.g. : 5th bit = parity (sum mod. 2) of the first 3 bits.

Some more definitions

Hamming distance $d(x, y)$ of two vectors : nb of bits different.

Hamming weight of a bit-vector : number of bits equal to 1.

Minimum distance of a code : minimum distance of any two codewords

For a linear code : minimum distance = minimum weight of non-zero codewords.

⇒ Minimum distance of Hamming $(7, 4)$ code : = 3

Decoding :

Let $r = x + b$ denote the received word ($b =$ error vector)

Maximum likelihood decoding :

Guess that codeword $\hat{x}$ was sent which maximizes the probability $p(\hat{x}|r)$.

Assuming that all codewords are equiprobable, this is equivalent to maximizing $p(r|\hat{x})$

If $d(\hat{x}, r) = k$ then $p(r|\hat{x}) = p^k(1-p)^{n-k}$ (here $n = 7$)

Hence : $p(r|\hat{x})$ maximal $\Leftrightarrow$ $d(\hat{x}, r)$ minimal (supposing that $p < 0.5$).

For the Hamming $(7, 4)$ code :

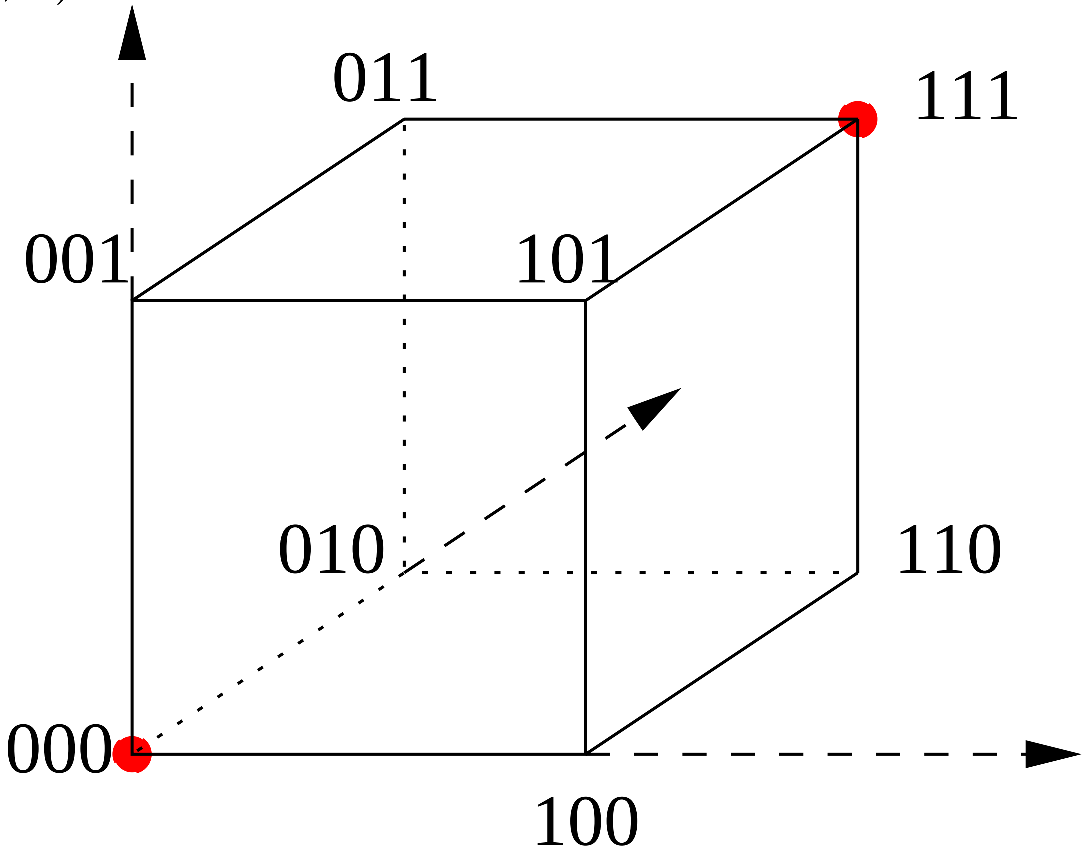
If Hamming weight of $b \leq 1$: correct decoding.

If Hamming weight of $b \leq 2$: correct detection.

Otherwise, minimum distance decoding will make errors.

Pictorial illustration

Repetition code $(3, 1)$

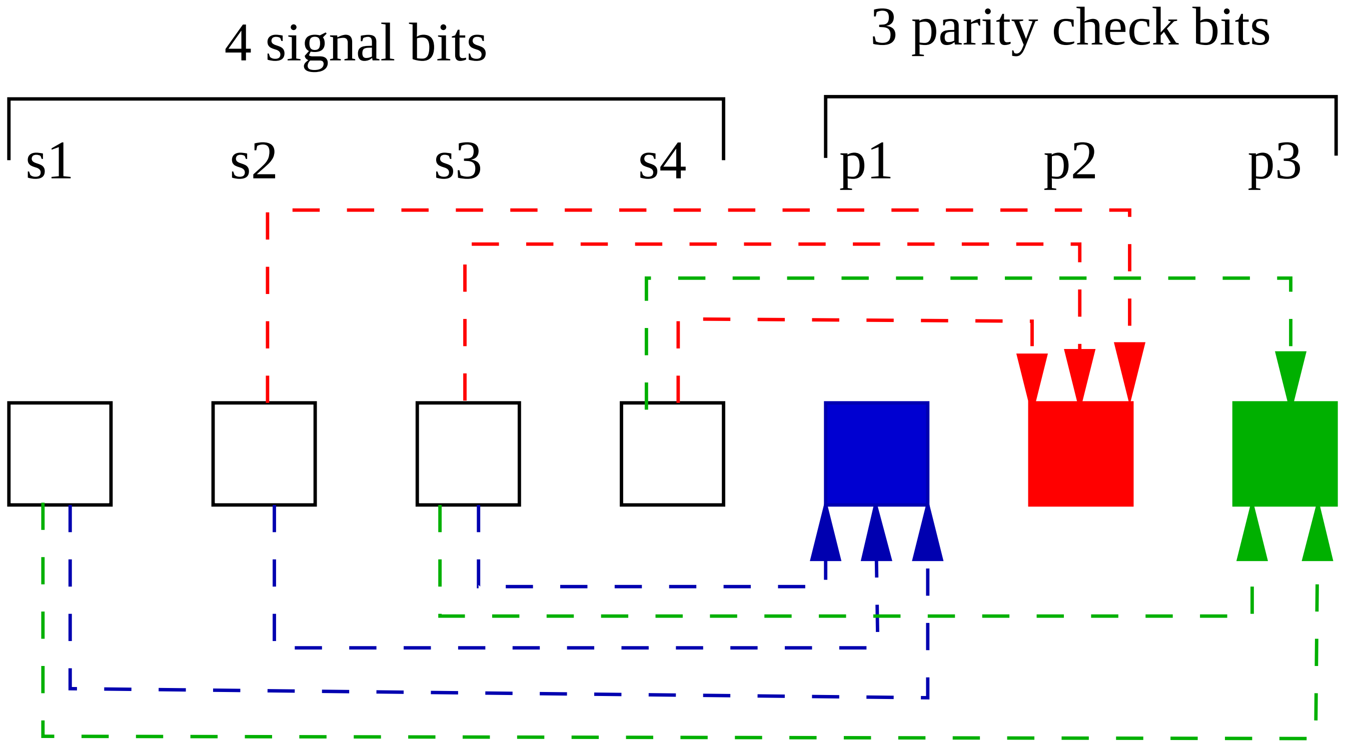


$\Rightarrow$ maximum likelihood decoding $\equiv$ nearest neighbor decoding

But : there is a more efficient way to do it than searching explicitly for the nearest neighbor (syndrom decoding)

Let us focus on correcting single errors with the Hamming $(7, 4)$ code

Meaning of the 7 bits of a codeword

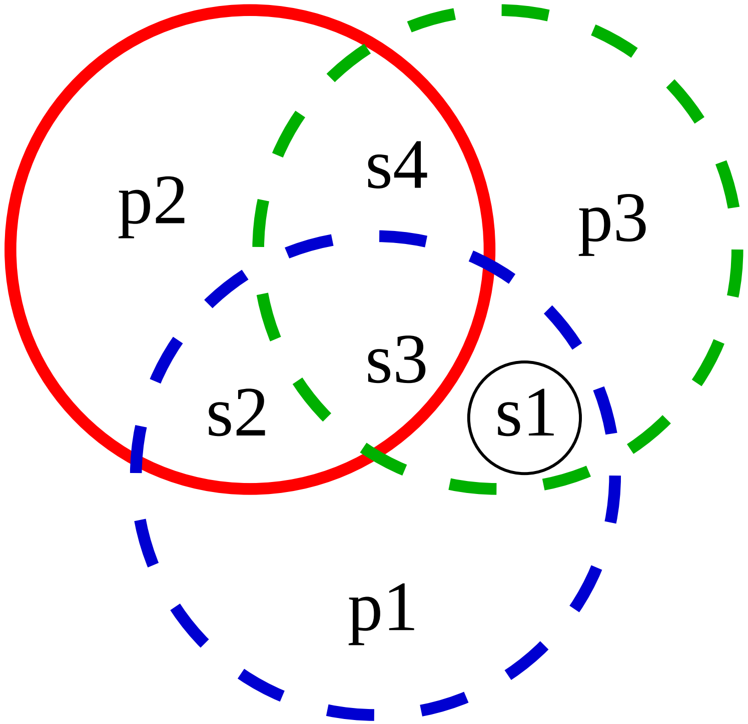


If error on any of the 4 first bits (signal bits) → two or three parity check violations.

If single error on one of the parity check bits : only this parity check is violated.

In both cases, we can identify erroneous bit.

Alternative representation : parity circles



Parity within each circle must be even if no error.

If this is not the case, we should flip one single received bit so as to realize this condition (always possible).

For instance, if parity in the green (upper right) and blue (lower) circles are not even $\Rightarrow$ flip bit s_1

Syndrom decoding

Syndrom : difference between the received parity bit vector and those which are recomputed from the received signal bits.

$\Rightarrow$ Syndrom is a vector of three bits $\Rightarrow 2^3 = 8$ possible syndroms

The syndrom contains all the information needed for optimal decoding :

8 possible syndroms $\rightarrow$ 8 most likely error patterns (can be precomputed).

E.g. : suppose that $r = 0101111$:
- signal bits $0101 \rightarrow$ codeword 0101101 (parity 101)
- syndrom : $101 + 111 = 010$ (bit per bit)
- more probable error pattern : 0000010
- decoded word : $01011\underline{0}1$

E.g. : suppose that $r = 0101110$:
- signal bits $0101 \rightarrow$ codeword 0101101 (parity 101)
- syndrom : $101 + 110 = 011$ (bit per bit)
- more probable error pattern : 0001000
- decoded signal bits : 0100 (code $010\underline{0}110$).

Summary

Like the repetition code, the Hamming code $(7,4)$ corrects single errors and detects double errors, but uses longer words (7 bits instead of 3) and has a higher rate.

If $p = 0.1$: probability of error per code word : 0.14

→ (signal) bit error rate (BER) of : 0.07

Less good in terms of BER but better in terms of code-rate : $R = 4/7$.

There seems to be a compromize between BER and Code-rate.

Intuitively : $\lim_{P_e \rightarrow 0} R(P_e) = 0$

(this is what most people still believed not so long ago...)

And then ?

...Shannon came...

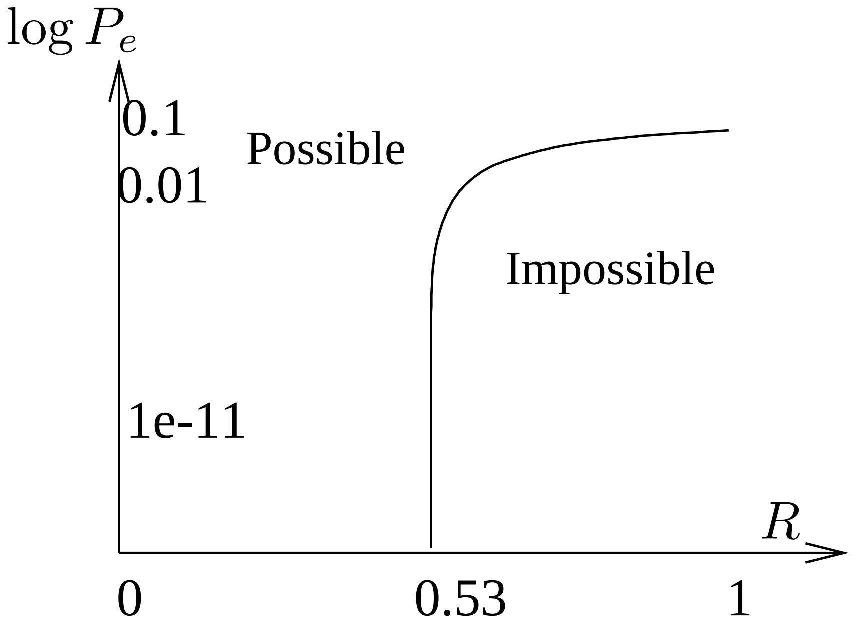
Second Shannon theorem

States that if $R < C(p) = 1 - H_2(p)$ then $P_e = 0$ may be attained.

Third Shannon theorem (rate distorsion : $P_e > 0$ tolerated)

Using irreversible compression we can further increase the code rate by a factor $\frac{1}{1-H_2(P_e)}$ if we accept to reduce realiability (i.e. $P_e \nearrow$).

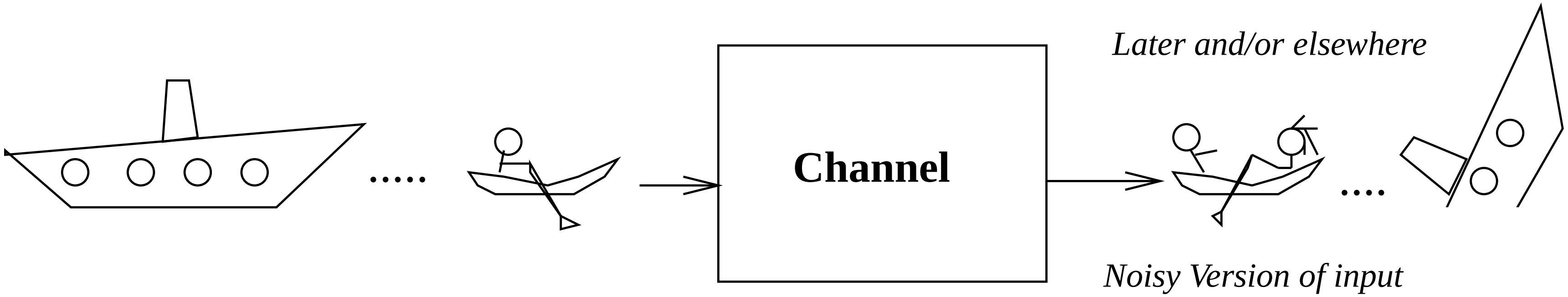
Conclusion : we can operate in a region satisfying $R \leq \frac{C(p)}{1-H_2(P_e)}$



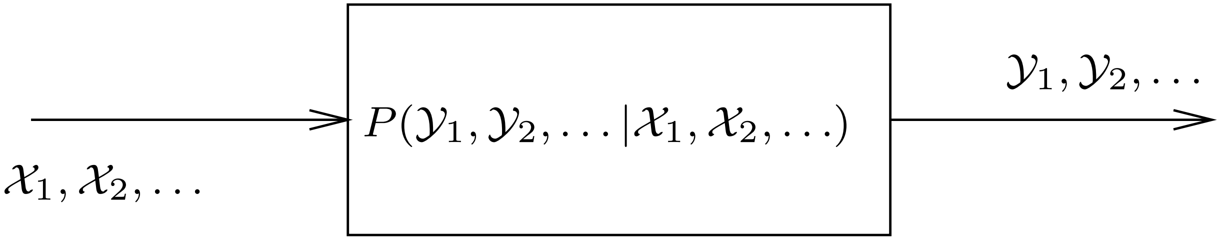
Conclusion : we only need two disks and a very good code to reach $P_e \leq 10^{-15}$.

2nd Shannon theorem (channel coding)

What is a channel ?



Abstract model :


$\mathcal{X}_1, \mathcal{X}_2, \ldots$
$P(\mathcal{Y}_1, \mathcal{Y}_2, \ldots | \mathcal{X}_1, \mathcal{X}_2, \ldots)$
$\mathcal{Y}_1, \mathcal{Y}_2, \ldots$

in other words, the specification of (all) the conditional probability distributions

$$P(\mathcal{Y}_1, \ldots, \mathcal{Y}_m | \mathcal{X}_1, \ldots, \mathcal{X}_n),$$

defined $\forall m, n = 1, 2, \ldots$

Simplifications

Causal channel : if $\forall m \leq n$

$$P(\mathcal{Y}_1,\ldots,\mathcal{Y}_m|\mathcal{X}_1,\ldots,\mathcal{X}_n) = P(\mathcal{Y}_1,\ldots,\mathcal{Y}_m|\mathcal{X}_1,\ldots,\mathcal{X}_m). \tag{1}$$

Causal and memoryless channel : if $\forall k \geq 2$

$$P(\mathcal{Y}_k|\mathcal{X}_1,\ldots,\mathcal{X}_k,\mathcal{Y}_1,\ldots,\mathcal{Y}_{k-1}) = P(\mathcal{Y}_k|\mathcal{X}_k), \tag{2}$$

Causal, memoryless and stationary channel : if $\forall k \geq 1$ we have

$$P(\mathcal{Y}_k|\mathcal{X}_k) = P(\mathcal{Y}|\mathcal{X}), \tag{3}$$

$\Rightarrow$ this will be our working model

1 symbol enters at time k → 1 symbol comes out at time k.

If stationary process at the input → stationary process out

If ergodic process at the input → ergodic process at the output

(NB: one can generalize to stationary channels of finite memory...)

Information capacity of a (stationary memoryless) channel

By definition :

$$C = \max_{P(\mathcal{X})} I(\mathcal{X};\mathcal{Y}). \tag{4}$$

Remarks

This quantity relates to one single use of the channel (one symbol)

$I(\mathcal{X};\mathcal{Y})$ depends both on source and channel properties.

C solely depends on channel properties.

We will see later that this quantity coincides with the notion of *operational capacity*.

NB: How would you generalize this notion to more general classes of channels ?

Examples of discrete channels and values of capacity

Channel transition matrix :

$$[P(Y_j|X_i)] = \begin{bmatrix} P(Y_1|X_1) & \cdots & P(Y_{|\mathcal{Y}|}|X_1) \\ \vdots & \ddots & \vdots \\ P(Y_1|X_{|\mathcal{X}|}) & \cdots & P(Y_{|\mathcal{Y}|}|X_{|\mathcal{X}|}) \end{bmatrix}$$

1. Binary channel without noise

Input and output alphabets are binary : $[P(Y_j|X_i)] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$

$I(\mathcal{X};\mathcal{Y}) = H(\mathcal{X})$, maximal when $H(\mathcal{X})$ is maximal (=1 Shannon).

Achievable rate (without errors) : 1 source symbol/channel use.

Can we do better ?

No, unless we admit $P_e > 0$.

2. Noisy channel without overlapping outputs

E.g. : transition matrix

$$\left[\begin{array}{cccc} p & (1-p) & 0 & 0 \\ 0 & 0 & q & (1-q) \end{array}\right].$$

$H(\mathcal{X}|\mathcal{Y}) = 0 \Rightarrow I(\mathcal{X};\mathcal{Y}) = H(\mathcal{X}) \Rightarrow C = 1$. (Achievable...)

3. Noisy type-writer

Input alphabet : a, b, c,..., z   Output alphabet : a, b, c,..., z

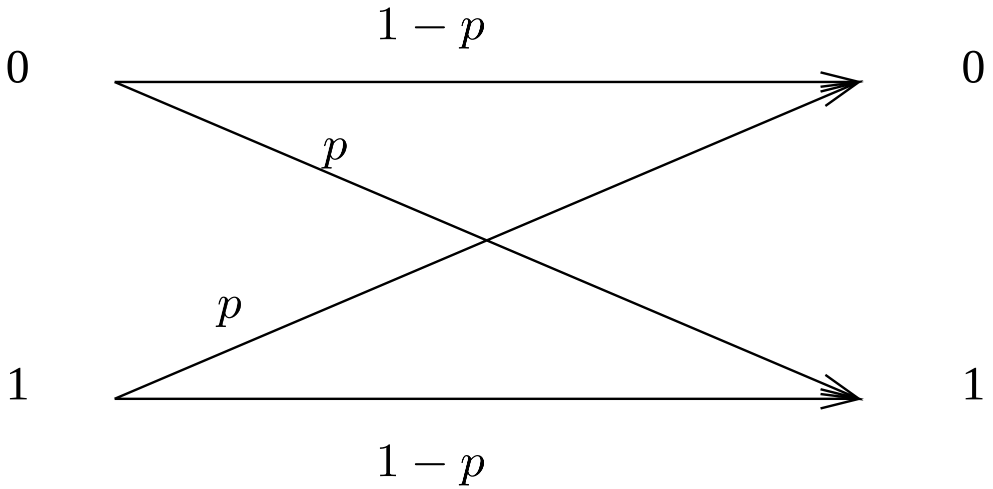
$P(a|a) = 0.5, P(b|a) = 0.5, P(b|b) = 0.5, P(c|b) = 0.5, \ldots P(z|z) = 0.5, P(a|z) = 0.5$

$I(\mathcal{X};\mathcal{Y}) = H(\mathcal{Y}) - H(\mathcal{Y}|\mathcal{X})$ with $H(\mathcal{Y}|\mathcal{X}) = 1 \Rightarrow$ max if outputs are equiprobable
E.g. if inputs are equiprobable : $H(\mathcal{Y}) = \log_2 26 \Rightarrow C = \log_2 13$

Achievable : just use the right subset of input alphabet...

(NB: this is **THE** idea of channel coding)

4. Binary symmetric channel



$$[P(Y_j|X_i)] = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}.$$

Information capacity of this channel :

$$\begin{aligned} I(\mathcal{X};\mathcal{Y}) &= H(\mathcal{Y}) - H(\mathcal{Y}|\mathcal{X}) = H(\mathcal{Y}) - \sum_{X\in\mathcal{X}} P(X)H(\mathcal{Y}|X) \\ &= H(\mathcal{Y}) - \sum_{X\in\mathcal{X}} P(X)H_2(p) = H(\mathcal{Y}) - H_2(p) \leq 1 - H_2(p), \end{aligned}$$

Equal to 0, if $p = 0.5$ and equal to 1 if $p = 0.0$. Symmetric : $C(p) = C(1-p)$.

Achievable ? : less trivial...

Main properties of the information capacity

1. $C \geq 0$.
2. $C \leq \min\{\log|\mathcal{X}|, \log|\mathcal{Y}|\}$.

Moreover, one can show that $I(\mathcal{X};\mathcal{Y})$ is continuous and concave with respect to $P(\mathcal{X})$.

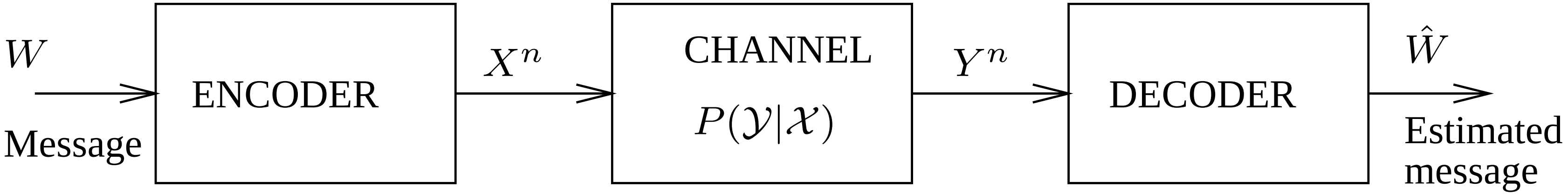
Thus every local maximum must be a global maximum (on the convex set of input probability distributions $P(\mathcal{X})$).

Since the function $I(\mathcal{X};\mathcal{Y})$ is upper bounded capacity must be finite.

One can use powerful optimisation techniques to compute information capacity for a large class of channels with the desired accuracy.

In general, the solution can not be obtained analytically.

Communication system



A message W (finite set of possible messages $\mathcal{W} = \{1, 2, \ldots, M\}$) is encoded by the *encoder* into a sequence of n channel input symbols, denoted by $X^n(W)$.

At the other end of the channel another (random) sequence of channel output symbols Y^n is received (distributed according to $P(\mathcal{Y}^n|X^n(W))$.)

The sequence Y^n is then decoded by the *decoder*, who chooses an element $\hat{W}(Y^n) \in \mathcal{W} \rightarrow$ the receiver makes an *error* if $\hat{W}(Y^n) \neq W$.

In what follows, we will suppose that the encoder and the decoder operate in a deterministic fashion :

- $X^n(W)$ is the coding rule (or function);

- $\hat{W}(Y^n)$ is the decoding rule (or function).

Memoryless channel specification

Input and output alphabets $\mathcal{X}$ and $\mathcal{Y}$, and the $P(\mathcal{Y}_k|\mathcal{X}_k)$ are given

Channel is used without feedback :

In this case : $P(\mathcal{Y}^n|\mathcal{X}^n) = \prod_{i=1}^{n} P(\mathcal{Y}_i|\mathcal{X}_i)$.

Definitions that will follow :

- Channel code (M, n)

- Different kinds of error rates...

- Communication rate.

- Achievable rates.

- Operational capacity.

(M, n) **Code**

An (M, n) code for a channel $(\mathcal{X}, P(\mathcal{Y}|\mathcal{X}), \mathcal{Y})$ is defined by

1. A set of indices $\{1, \ldots, M\}$;
2. A coding function $X^n(\cdot) \quad : \{1, \ldots, M\} \rightarrow \mathcal{X}^n$, which gives the codebook $X^n(1), \ldots, X^n(M)$.
3. A decoding function

   $$g(\cdot) \; : \mathcal{Y}^n \rightarrow \{1, \ldots, M\}, \tag{5}$$

   which is a deterministic mapping from all possible output strings into an input index $g(Y^n)$.

$\Rightarrow$ M code words coded using n input symbols.

Decoding error rates

1. Conditional Probability of error given that index i was sent

$$\lambda_i = P(g(\mathcal{Y}^n) \neq i|\mathcal{X}^n = X^n(i)) = \sum_{Y^n \in \mathcal{Y}^n} P(Y^n|X^n(i))(1 - \delta_{g(Y^n),i})$$

2. Maximal error probability for and (M, n) code :

$$\lambda^{(n)} = \max_{i \in \{1,\ldots,M\}} \lambda_i$$

3. The (arithmetic) average probability of error :

$$P_e^{(n)} = \frac{1}{M} \sum_{i=1}^{M} \lambda_i$$

$\Rightarrow$ expected error rate if i has a uniform distribution.

Optimal decoding rule

By definition : the decoding rule which minimises expected error rate.

For a received word $Y^n \rightarrow$ choose i such that $P(X^n(i)|Y^n)$ is maximal.

$\Rightarrow$ **maximises a posteriori probability (MAP)**

$\Rightarrow$ minimises for each Y^n error probability
$\Rightarrow$ minimises the expected error rate.

$\Rightarrow$ general principle in decision theory : *Bayes rule*

We use information $\mathcal{Y}$ (random variable which is observed).

We want to guess (decide on) a certain variable D (choose among M possibilities).

Correct decision : $\mathcal{D}^*$ a random variable $\Rightarrow P(\mathcal{D}^*|\mathcal{Y})$ known.

Cost of the taken decision : 0 if correct, 1 if incorrect.

Optimal decision based on information $\mathcal{Y}$: $\hat{D}(Y) = \arg_D \max\{P(D|Y)\}$

For our channel :

$$P(X^n(i)|Y^n) = \frac{P(Y^n|X^n(i))P(X^n(i))}{\sum_{i=1}^{M} P(Y^n|X^n(i))P(X^n(i))}$$

Since $\sum_{i=1}^{M} P(Y^n|X^n(i))P(X^n(i))$ does not depend on the decision, this is the same than maximizing $P(Y^n|X^n(i))P(X^n(i))$.

Discussion

$P(Y^n|X^n(i))$: channel specification.

$P(X^n(i))$: source specification.

If non-redundant source : $P(X^n(i))$ independent of $i \Rightarrow$ maximize $P(Y^n|X^n(i))$.

$\Rightarrow$ **Maximum likelihood rule** : minimize $P_e^{(n)}$

Quasi optimal, if source is quasi non-redundant.

E.g. if we code long source messages (cf. AEP)

Communication rate : denoted R

The communication rate (denoted by R) of an (M, n) code is defined by $R = \frac{\log M}{n}$ Shannon/channel use $\equiv$ input entropy per channel use if inputs are uniformly distributed.

Achievable rate (more subtle notion)

R is said to be achievable if $\exists$ a sequence of $(M(n), n), n = 1, 2, \ldots$ codes suchthat

1. $M(n) = \lceil 2^{nR} \rceil$ and 2. $\lim_{n\to\infty} \lambda^{(n)} = 0$

$\Rightarrow$ codes of rate $\approx R$, eventually become “quasi perfect” and remain so (when using very long codewords)

Remark

Definition is independent of the source distribution (cf. maximal probability of error).

Operational capacity : $C_o =$ is the supremum of all achievable rates.

$R = 0$ is achievable, but we need to check that it is possible to have $C_o > 0$.

Second Shannon theorem

Objective : prove that information capacity C is equal to operational capacity C_o.

Hypothesis : the pair $(\mathcal{X}^n, \mathcal{Y}^n)$ satisfy the AEP (stationary and ergodic : OK for stationary finite memory channels and ergodic inputs.)

Information capacity (per channel use) :

$$C = \lim_{n\to\infty} \frac{1}{n} \max_{P(\mathcal{X}^n)} \{I(\mathcal{X}^n; \mathcal{Y}^n)\}$$

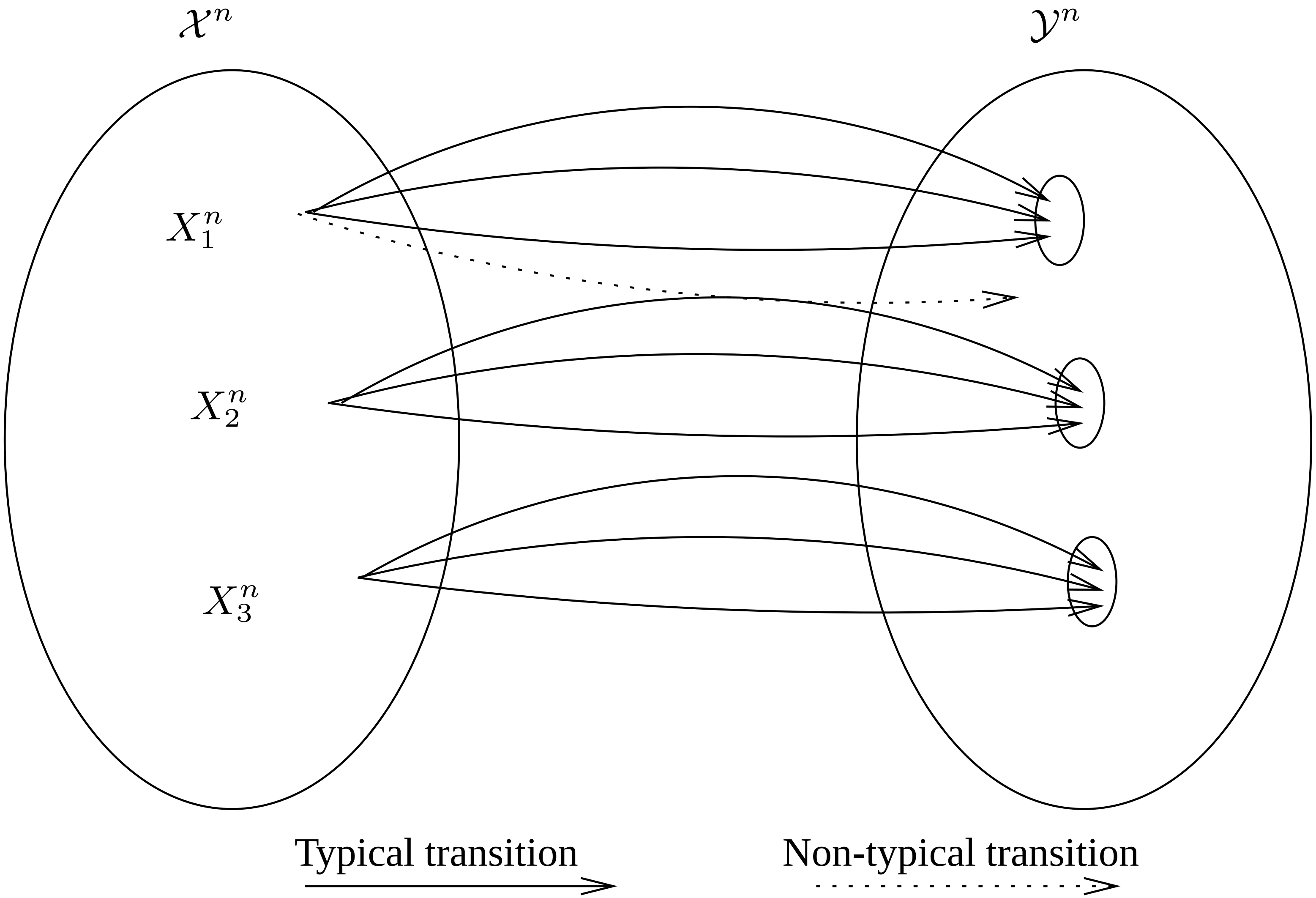
(with $\mathcal{X}^n$ stationary and ergodic)

For memoryless channels the maximum is obtained for independent source symbols and the above definition yields indeed : $C = \max_{P(\mathcal{X})} I(\mathcal{X}; \mathcal{Y})$

Basic idea : for large block lengths, every channel looks like the noisy typewriter :
- it has a subset of inputs that produce essentially disjoint sequences at the output
- the rest of the proof is matter of counting and packing...

Outline of the proof (Shannon's random coding ideas)

Let us fix for a moment $P(\mathcal{X})$, n and M, and construct a codebook by generating random signals according to $P(\mathcal{X})$ ($M \times n$ drawings) ⇒ if n is large the codewords as well as the received sequences must be typical (AEP)



Let us count and pack (approximately and intuitively...)

At the input : $2^{nH(\mathcal{X})}$ possible typical messages (we choose M of them at random to construct our codebook).

At the output : for each X^n (typical) $2^{nH(\mathcal{Y}|\mathcal{X})}$ possible typical output sequences.

The total number of possible (typical outputs) is $2^{nH(\mathcal{Y})}$

If we want that there is no overlap : we must impose that

$$\frac{2^{nH(\mathcal{Y})}}{M} \geq 2^{nH(\mathcal{Y}|\mathcal{X})} \Rightarrow M \leq 2^{nI(\mathcal{Y};\mathcal{X})}$$

Now, let us choose the $P(\mathcal{X})$ which maximizes $I(\mathcal{Y};\mathcal{X}) = C$: it should be possible to find $M \leq 2^{nC}$ input sequences such that the output regions do not overlap.

Conclusion

Using long codewords ($n \to \infty$) it is possible to exploit redundancy (correlation between inputs and outputs) to transmit information in a reliable way.

Second Shannon theorem

Statement in two parts :

Forwards : $R < C \Rightarrow R$ achievable ($\lambda^{(n)} \rightarrow 0$).

Backwards : $R > C \Rightarrow R$ not achievable.

The proof is based on random coding and the joint AEP.

One can show that for a long codewords, a large proportion of the random codes generated by drawing $M = 2^{n(C-\epsilon)}$ words are actually very good.

Caveat

In practice random coding is not a feasible solution because of computational limitations, just like typical message data compression is not a feasible approach to data compression.