

3F4: Data Transmission

Handout 1: Binary Linear Coding and Decoding Fundamentals

Jossy Sayir

Probability, Systems, Information and Inference Lab Ψ^2
Department of Engineering
js851@cam.ac.uk

Lent Term 2026

1 / 18

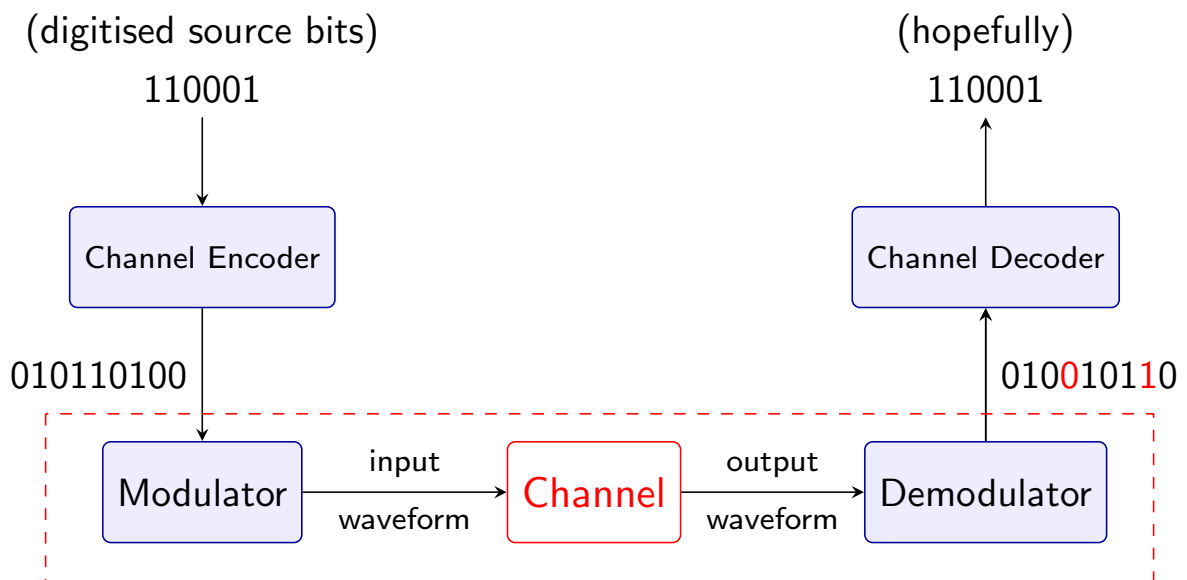
Outline of this part of the course

1. Binary Linear Coding and Decoding Fundamentals
2. Convolutional Codes
3. Trellis based decoders and related algorithms
4. Distance properties of convolutional codes

Today's lecture contains parts that are revision for the 48 among you who have taken 3F7, and new for the 7 who have not taken 3F7.

2 / 18

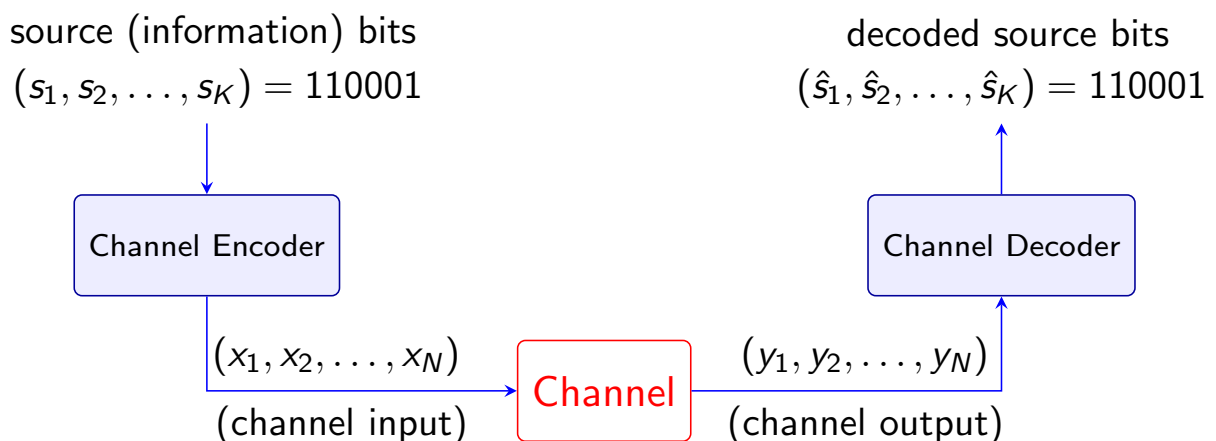
The communication channel as seen in 2P6 Comms...



- From now on we consider the part of the system enclosed by dashed lines as **the channel**
- the aim of coding is to correct errors (or is it?)

3 / 18

A better channel model

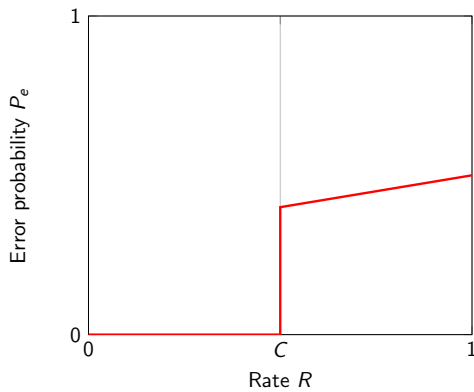


- **Channel:** A collection of conditional probability distributions $P(y|x) = \Pr(Y = y|X = x)$ (not necessarily “errors” but noisy observations!)
- **Capacity:** Fastest transmission rate, in bits per channel use, that can be achieved with arbitrarily small error probability

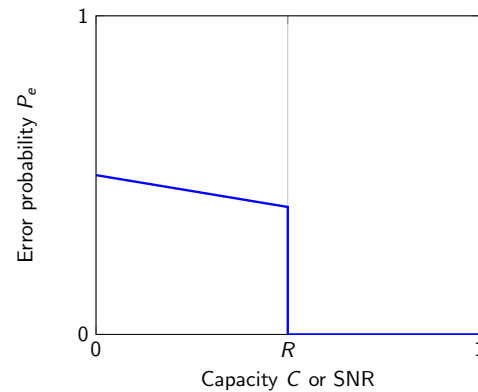
4 / 18

Channel Capacity

- $C = \max_{P_X} I(X; Y)$ in bits per channel use (or divide by channel use interval T to obtain in bits per second)
- $I(X; Y)$ the mutual information is a function of P_{XY}
- To approach capacity, the code symbols X_i transmitted over the channel must be distributed according to the probability distribution P_X that maximises the expression

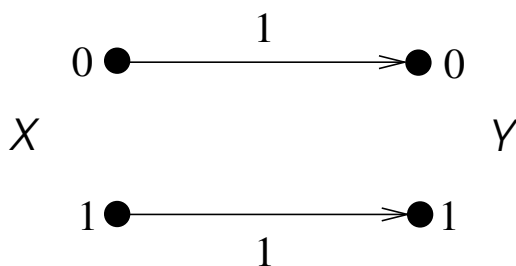


(a) Varying the rate for a given capacity



(b) Varying the capacity or Signal-to-Noise Ratio (SNR) for a given rate

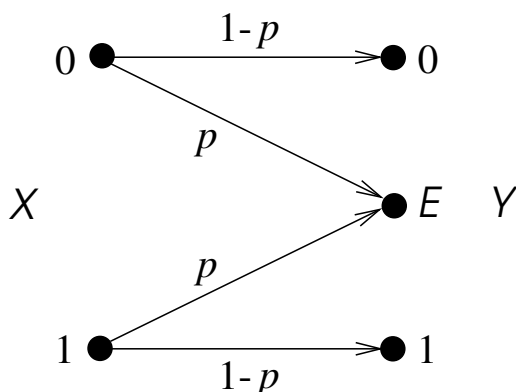
Simple channel examples



Binary noiseless channel

Capacity: Obviously

$C = 1$ bit/transmission

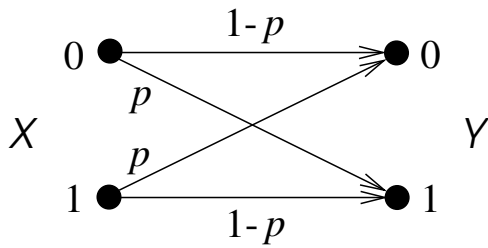


Erasure channel

Capacity: A simple (and correct, as it turns out) guess is

$C = 1 - p$ bits/transmission

Not-so-simple channel examples



Binary symmetric channel

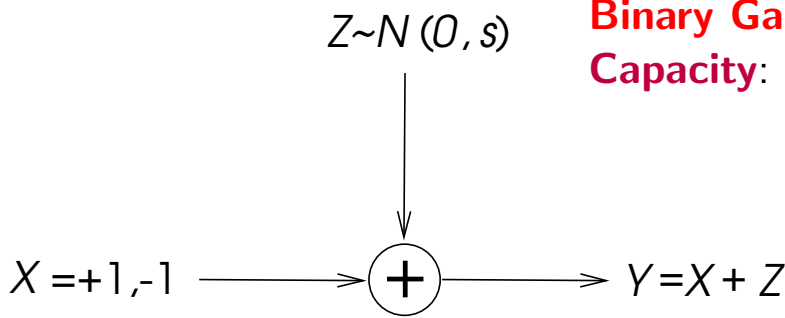
with crossover probability p

BSC(p)

Capacity:

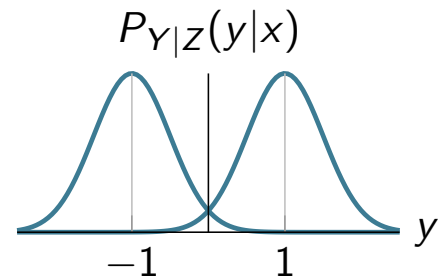
$$C = 1 - H_2(p) \text{ bits/transmission}$$

$$[H_2(p) = -p \log_2 p - (1-p) \log_2 (1-p)]$$



Binary Gaussian noise channel

Capacity: ???



7 / 18

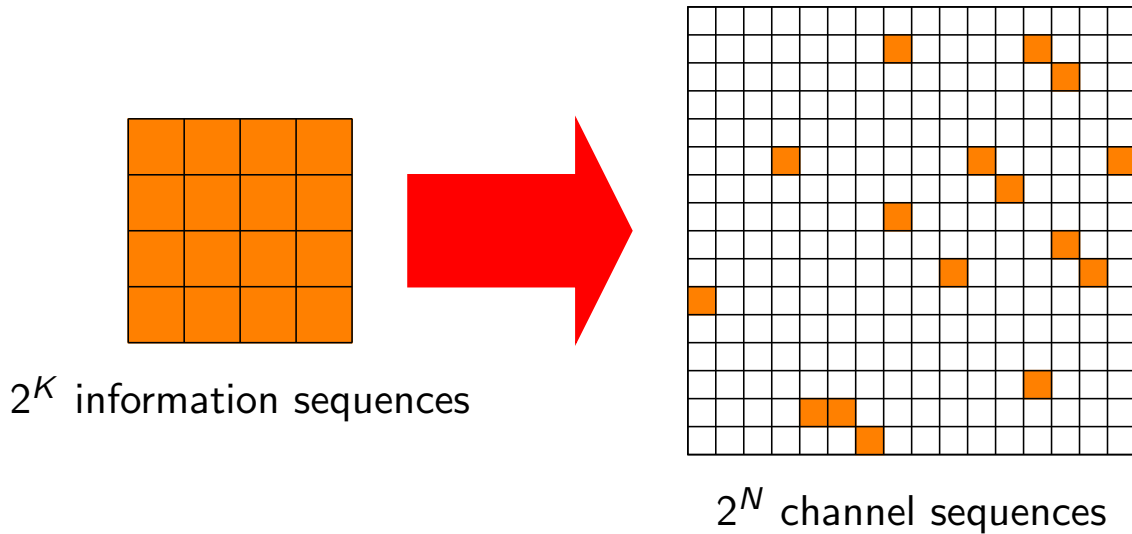
Binary-input symmetric memoryless channels (BISMC)

- Binary input alphabet $\{0, 1\}$, output alphabet arbitrary (possibly continuous)
- The output alphabet can be decomposed into pairs $\{y_0, y_1\}$ and at most one singleton y^* such that
 - the sub-channels $\{0, 1\} \rightarrow \{y_0, y_1\}$ are BSCs
 - $P_{Y|X}(y^*|0) = P_{Y|X}(y^*|1)$
 - think of it as: 1) “how strong is the noise” and 2) given the noise strength, the two possible observations are symmetric
- All the channels we saw in previous slides are BISMCs

8 / 18

Block coding

- maps K information bits to N code binary digits, $N > K$
- add redundancy



9 / 18

Binary Linear Block Codes

- K binary basis row vectors of length N , $\underline{g}_1, \dots, \underline{g}_K$
- the code forms a **vector space** \mathcal{C} over binary numbers
- The data bits are mapped to a codeword as follows:

$$\underline{x} = s_1 \underline{g}_1 + s_2 \underline{g}_2 + \dots + s_K \underline{g}_K$$

(all additions **modulo 2**)

This can be compactly written as

$$\underline{x} = \underline{s} \mathbf{G}, \quad \text{where } \mathbf{G} \text{ is the } K \times N \text{ matrix } \mathbf{G} = \begin{bmatrix} \underline{g}_1 \\ \underline{g}_2 \\ \vdots \\ \underline{g}_K \end{bmatrix}$$

- \mathbf{G} is a **generator matrix** of the code
- K is the code **dimension**
- N is the code **blocklength**
- $R = K/N$ is the **rate** of the code

10 / 18

Binary linear block codes (continued)

- The generator matrix is not unique (any basis of the vector space will do!)
- One can always find a basis of the form

$$\mathbf{G} = \left[\begin{array}{cc} I_{K \times K} & P_{K \times (N-K)} \end{array} \right]$$

This is called the **systematic encoder matrix**. When used for encoding, it has the property that

$$(x_1, \dots, x_N) = (s_1, \dots, s_K, p_1, \dots, p_{N-K}),$$

i.e., the codeword consists of the information word followed by so-called *parity bits*

- Another description of the code is via a basis of its null space (recall IB P7 Linear Algebra)

$$\underline{x}\mathbf{H}^T = \underline{0}$$

where the rows of \mathbf{H} are $(N - K)$ basis vectors of the null space. \mathbf{H} is called the **parity-check matrix** of the code

11 / 18

Marginal probability distribution for a linear code

- If the source has been appropriately **compressed** (source encoded), the source digits s_1, \dots, s_K are independent and uniformly distributed $P_S(0) = P_S(1) = 1/2$
- Code digit x_j is obtained by summing modulo 2

$$x_j = \sum_i s_i g_{ij} = \sum_{i:g_{ij}=1} s_i$$

- **Adding uniformly independent random variables modulo 2 yields a uniform binary random variable!**
- Unless all elements in the i -th column of the generator matrix are zero¹, $P_{X_i}(0) = P_{X_i}(1) = 1/2$
- **Binary linear codes always have a uniform marginal distribution. They are only suited for channel with a binary uniform capacity-achieving distribution, i.e., BiSMCs!**

¹Why is this a pathological case?

Weight/Distance properties of linear codes

- Hamming weight $w(\underline{x})$ of a vector \underline{x} is the **number of non-zero elements** of \underline{x}
- Hamming distance $d(\underline{x}, \underline{y}) = w(\underline{x} - \underline{y}) = w(\underline{x} + \underline{y})$ where the last equality holds for binary vectors because $+$ and $-$ are the same in binary, e.g., $1 - 1 = 1 + 1 = 0$.
- For a binary linear code, if $\underline{x}_1 = \underline{s}_1 \mathbf{G}$ and $\underline{x}_2 = \underline{s}_2 \mathbf{G}$ are codewords, then

$$\underline{y} = \underline{x}_1 + \underline{x}_2 = (\underline{s}_1 + \underline{s}_2) \mathbf{G}$$

is also a codeword. Note that if $d(\underline{x}_1, \underline{x}_2) = d$ then $w(\underline{y}) = d$.

- **If two codewords are at a distance d from each other, then there exists a codeword of weight d .**
- For any codeword \underline{x}_3 , there exists a codeword $\underline{x}_4 = \underline{x}_3 + \underline{y}$ such that $d(\underline{x}_3, \underline{x}_4) = d$.
- **There is a codeword at distance d from every codeword!**

13 / 18

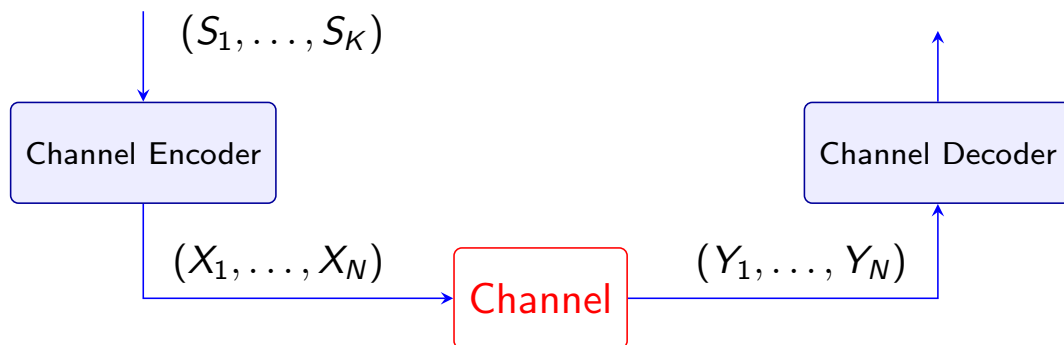
Distance and weight symmetry of linear codes

The “distance spectrum” (number of codewords at distance 1,2,3,4,...) from any codeword is the same and equal to its “weight spectrum” (number of weight 1,2,3,4,..., or, equivalently, at distance 1,2,3,4,... from the all zero codeword $\underline{0} = \underline{0G}$)

All decoder operations, and probabilities of success/error are *codeword independent* for a linear block code transmitted over a BSMC! We might as well just design decoders and evaluate the probability of success or error assuming that the all zero codeword $\underline{0}$ was transmitted.

14 / 18

Decoder perspective



An unknown codeword $(X_1, \dots, X_N) \in \mathcal{C}$ from a linear block code \mathcal{C} is transmitted and an observation vector $(Y_1, \dots, Y_N) = (y_1, \dots, y_N)$ is received.

15 / 18

Flavours of optimality

- Blockwise optimal decoding (optimal for **Block Error Rate**):

$$(\hat{x}_1, \dots, \hat{x}_N) = \arg \max_{(x_1, \dots, x_N) \in \mathcal{C}} P(x_1, \dots, x_N | y_1, \dots, y_N)$$

and $(\hat{s}_1, \dots, \hat{s}_K)$ is the corresponding information word.

- Bitwise optimal decoding (optimal for **Bit Error Rate**):

$$\hat{x}_\ell = \arg \max_{x \in \{0,1\}} P_{X_\ell | Y_1, \dots, Y_N}(x | y_1, \dots, y_N) \text{ for } \ell = 1, \dots, N$$

or

$$\hat{s}_\ell = \arg \max_{s \in \{0,1\}} P_{S_\ell | Y_1, \dots, Y_N}(s | y_1, \dots, y_N)$$

- Both are “Maximum A-Posteriori” (MAP) rules
- Using Bayes’ rule, this is often re-framed as “Maximum Likelihood (ML) decoding”. Note however that “ML” in statistics refers a sub-optimal decision rule where one assumes non-uniform data to be uniform. In communications, compressed transmitted data is truly uniform, so ML and MAP are equivalent

16 / 18

Decoder implementation

	Computation	Decision
Block optimal	$P(\underline{x} \underline{y})$	$\arg \max_{\underline{x}} P(\underline{x} \underline{y})$
Bit optimal	$P(x_\ell \underline{y})$	$\arg \max_{x \in \{0,1\}} P(X_\ell = x \underline{y})$

- Bayes' rule:

$$P(\underline{x}|\underline{y}) = \frac{P(\underline{x})P(\underline{y}|\underline{x})}{P(\underline{y})} = \frac{P(\underline{x}) \prod_{\ell=1}^n P(y_\ell|x_\ell)}{P(\underline{y})}$$

- Block decision:

$$\arg \max_{\underline{x} \in \mathcal{C}} P(\underline{x}|\underline{y}) = \arg \max_{\underline{x} \in \mathcal{C}} P(\underline{x}) \prod_{\ell=1}^n P(y_\ell|x_\ell)$$

- Bit computation:

$$P(X_\ell = x|\underline{y}) = \sum_{\underline{x} \in \mathcal{C}, x_\ell = x} P(\underline{x}|\underline{y})$$

- We need to compute the sum or max over 2^K codewords!

17 / 18

The coding “problem”

- Information theory teaches us that for a fixed rate, N and hence $K = RN$ must be allowed to grow to achieve smaller error rates
- Information theory also says that codes are good “on average”. We can expect to get good results by just picking a $K \times N$ generator matrix \mathbf{G} at random
- Optimal decoding (to minimise bit or block error probability) requires the computation of a sum or max over 2^K codewords \rightarrow the complexity of the optimal decoder grows exponentially in K
- We need strategies to reduce the complexity of the encoder if we want to achieve good results
- LDPC codes (taught in 3F7) use a sub-optimal decoder that works on codes whose parity-check matrix is sparse
- In 3F4, we will learn about Turbo Codes, an equivalent coding method based on [convolutional codes](#)

18 / 18