Lecture 1
Basic Notation

1.1. What This Class is About

Traditionally, coding theory has been about methods for reliable transmission of information through unreliable media. The transmission media are called *communication channels*. Infidelities of the communication channels leads to corruption of the transmitted data. The role of coding theory is to pre-process the data in such a way as to provide reliable recovery after corruption. This pre-processing is often called "encoding." The pre-processing has obviously to take into account the nature of infidelities of the communication channel: for example, if a channel has the property that it occasionally flips a bit, then one might want to employ a different coding technique, than in the case of a very unreliable channel which flips bits more often than not.

One of the immediate questions that comes to mind is that of fundamental limits on reliable communication when the channel is unreliable. This is answered by Shannon's coding theorems. Although this theorem governs pretty much all the digital communication today, its proof is somewhat unsatisfactory, in that it does not present a constructive way for coding data so as to achieve these limits. Moreover, the classical proof is far from any attempt to provide efficient algorithmic solutions for the coding, and the corresponding decoding tasks. In essence, this is what this course is about: design and analysis of codes and corresponding encoding/decoding algorithms that are extremely efficient in terms of (a) the way they can cope with channel infidelities (so called error-correction capability), and (b) and in terms of their encoding/decoding algorithms.

Coding theory very naturally borrows techniques from various mathematical disciplines, among them probability and statistics, algebra, and combinatorics. We will not be able to develop all the basics in this course, and we will assume that the audience is somewhat familiar with these notions.

Before we dive into the design and analysis of codes, we have to start with the basics. This lecture will provide some of the basic definitions and notions that will be used throughout this class.

1.2. Channels

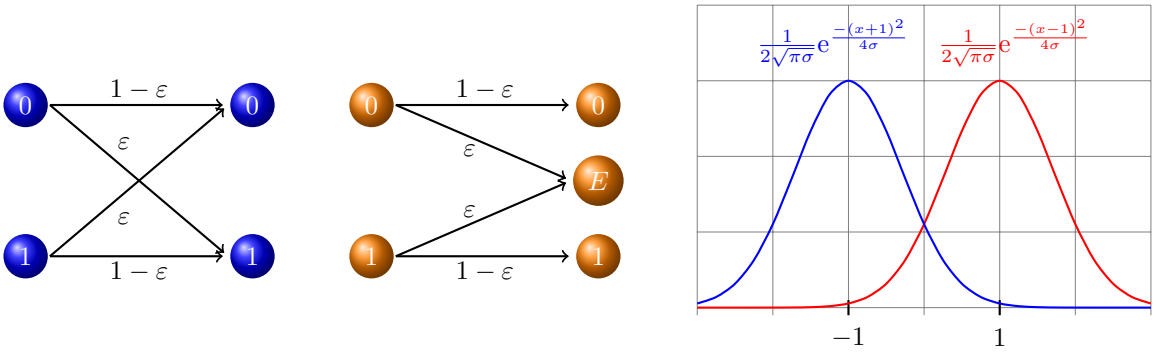
1.2.1. Definition

A (finite input) memoryless communication channel $\mathcal{C}$ is a triple (Σ, I, p), where

1. Σ is a finite set
2. I is a measurable set with measure μ
3. $p\colon I \times \Sigma \to \mathbb{R}_{\geq 0}$ is a function such that $\int_I p(y,x)\mathrm{d}\mu = 1$.

Often $p(y,x)$ is interpreted as the "probability of receiving y given that x was sent." Often times, $p(y,x)$ is written as $p(y|x)$ for this reason. We will always deal with memoryless channels, so that we extend p to a function on $\Sigma^n \times I^n$ as $p((y_1,\ldots,y_n),(x_1,\ldots,x_n)) := \prod_{i=1}^n p(y_i,x_i)$.

Lecture by Amin Shokrollahi



(a) (b) (c)

Figure 1.1: (a) The binary symmetric channel, (b) The binary erasure channel, (c) The binary input AWGN channel

1.2.2. Symmetry

A *permutation* of a set S is a bijective map on S. Suppose that μ is a permutation on S such that $\mu^t = \mathrm{id}_S$ for some t. This means that if we apply μ t times to any element of S, we obtain that element back. The *orbit* of an element a in S under μ is the set $\{a, \mu(a), \ldots, \mu^{t-1}(a)\}$. The size of the orbit is at most t, but can be smaller.

A channel is called *input (output) symmetric* if there exist permutations σ and τ on Σ and I, respectively, with $\sigma^q = \mathrm{id}_\Sigma$ and $\tau^q = \mathrm{id}_I$, where $q = |\Sigma|$, such that all orbits of σ (all orbits of τ) have q elements and such that for all $x \in \Sigma$ and $y \in I$ we have $p(y, x) = p(\tau(y), \sigma(x))$. A channel is called *symmetric* if it is input and output symmetric.

A channel is called *binary* if Σ has only two elements. Combining the notation introduced so far, we can talk of binary input symmetric or binary output symmetric, or binary symmetric channels. A *binary symmetric* channel is a binary channel that is symmetric. In all these cases, the permutations σ and τ giving rise to the definitions are involutions, i.e., $\sigma^2 = \mathrm{id}_\Sigma$ and $\tau^2 = \mathrm{id}_I$.

A channel is called *discrete* if the output alphabet I is a discrete set.

We define the *error probability* of an input symmetric binary channel as

$$\frac{1}{2}\int_I \min(p(\tau(y), 1), p(y, 1))\mathrm{d}y,$$

where we assume that $1 \in \Sigma$.

Example 1.1.

1. Let $\Sigma = I = \{0, 1\}$, and $p(0, 0) = p(1, 1) = 1$, $p(0, 1) = p(1, 0) = 0$. This channel is symmetric and it is called the trivial channel.

2. Let $\Sigma = \{0, 1\}$ and $I = \{0, 1, E\}$, $\varepsilon \in [0, 1]$, and $p(0, 0) = p(1, 1) = 1 - \varepsilon$, $p(0, E) = p(1, E) = \varepsilon$, and $p(0, 1) = p(1, 0) = 0$. This channel is called the *binary erasure channel* with probability ε, denote $\mathrm{BEC}(\varepsilon)$. It is input symmetric via the maps $\sigma(0) = 1, \sigma(1) = 0$, and $\tau(0) = 1, \tau(1) = 0$, and $\tau(E) = E$. Its error probability is equal to $\varepsilon/2$.

3. Let $\Sigma = I = \{0, 1\}$, $\varepsilon \in [0, 1]$, and $p(0, 0) = p(1, 1) = 1 - \varepsilon$, and $p(0, 1) = p(1, 0) = \varepsilon$. The channel is symmetric via $\sigma = \tau$, and $\sigma(0) = 1, \sigma(1) = 0$. This channel is called the *binary symmetric channel* with crossover probability ε, and is denoted by $\mathrm{BSC}(\epsilon)$. Its error probability is ε.

4. Let $\Sigma = \{-1, +1\}$, $I = \mathbb{R}$, $\sigma \in \mathbb{R}_{>0}$, and $p(y, a) = \frac{1}{2\sqrt{\pi\sigma}}\mathrm{e}^{-(x-a)^2/4\sigma}$ for $a \in \{-1, +1\}$. This channel is called the *binary input additive white Gaussian noise* channel with variance σ, and is denoted by $\mathrm{AWGN}(\sigma)$. The error probability of this channel is $\frac{1}{2\sqrt{\pi\sigma}}\int_{-\infty}^{0}\mathrm{e}^{-(x-1)^2/4\sigma}\mathrm{d}x$. Using the Q-function, the error probability can be expressed as $Q(1/\sqrt{2\sigma})$. This channel is symmetric via $\sigma(a) = -a$, $a \in \{-1, 1\}$, and $\tau(y) = -y$, $y \in \mathbb{R}$.

5. Let $\Sigma = \{0, 1, \ldots, q-1\} = I$, $\varepsilon \in [0, 1]$, and $p(a, b) = \varepsilon/(q-1)$ if $b \neq a$, and $p(a, a) = 1 - \varepsilon$. This channel is symmetric via the functions $\sigma = \tau$ with $\sigma(x) = (x + 1) \bmod q$. This channel is called the q-ary symmetric channel with probability ε, and is denoted by q-$\mathrm{SC}(\varepsilon)$

The first three communication channels given above are binary.

Suppose that $\mathcal{C}$ is a discrete memoryless channel. The "transition matrix" of the channel is the matrix $P(\mathcal{C}) = (p_{xy})_{x\in\Sigma, y\in I}$, where $p_{xy} = p(y, x)$. For example,

$$P(\mathrm{BSC}(\varepsilon)) = \begin{pmatrix} 1-\varepsilon & \varepsilon \\ \varepsilon & 1-\varepsilon \end{pmatrix}, \qquad P(\mathrm{BEC}(\varepsilon)) = \begin{pmatrix} 1-\varepsilon & \varepsilon & 0 \\ 0 & \varepsilon & 1-\varepsilon \end{pmatrix}.$$

The following remark is a little mental exercise.

Remark 1.2. *If the discrete channel $\mathcal{C}$ is input symmetric then all rows of $P(\mathcal{C})$ are permutations of the first row, and if it is output symmetric then all columns of $P(\mathcal{C})$ are permutations of the first column.*

1.2.3. Operational Meaning of a Channel

Operationally, a communication channel models an ideal transmission medium, over which we wish to transmit the elements of the alphabet Σ. At every channel use, we transmit one of the elements of $x \in \Sigma$, and we receive an element $y \in I$. The transmitted element induces a probability distribution on I. For any $y \in I$ the value $p(y, x)$ describes the probability of receiving y if x is sent. The task is to estimate x from the received value y. If the channel has a nonzero average error probability π, then on average (over the choice of x) any estimate for x from y will have an error probability of at least π. To obtain error-free communication, we proceed as follows: Instead of sending x, we send a vector $(x_1, \ldots, x_n) \in \Sigma^n$ using the channel n times consecutively. If the x_i are chosen independently, then we still have the nonzero probability of estimation. However, if there is a dependency among the x_i, then the error probability of the estimation may be made arbitrarily small.

1.3. Codes and Maximum-Likelihood Decoding

A code C over the alphabet Σ is a subset of Σ^n. The (binary) *rate* of the code is the quantity $\log_2(|C|)/n$, and the *symbol rate* of the code is the quantity $\log_{|\Sigma|}(|C|)/n$. The binary rate measures the average number of bits sent over the communication channel when elements of C are used for n consecutive channel uses. The symbol rate measures the average number of alphabet symbols sent over the channel in n consecutive channel uses.

For a code, we define the Maximum-Likelihood (ML)-decoding as follows: for a received word $y \in I^n$ the $x \in C$ maximizing $p(y, x)$ is called the ML-decoding of y.

Theorem 1.3. *If $\mathcal{C} = \mathrm{BSC}(\varepsilon)$, $\varepsilon < 1/2$, then ML-decoding is equivalent to finding a codeword that has the smallest Hamming distance to the received word.*

Proof. Suppose that y is received. Then, for all $z \in C$ we have

$$p(y, z) = \prod_{i=1}^{n} p(y_i, z_i) = (1-\varepsilon)^n \left(\frac{\varepsilon}{1-\varepsilon}\right)^{d(y,z)},$$

where $d(y, z)$ is the Hamming distance between y and z. Since $\varepsilon < 1/2$, the quantity $\varepsilon/(1-\varepsilon) < 1$, and the quantity $p(y, z)$ is largest when $d(y, z)$ is smallest. □

1.4. Channel Capacity

Given a communication channel, what is the maximum rate at which we can send symbols through the channel, and expect the error probability of the ML-decoder to go to zero?

If we assume that the error probability of the ML-decoder is indeed zero, then we call this quantity the *zero-error capacity* of the channel.

If we assume that the error probability of the ML-decoder approaches zero as the length of the underlying code increases, then we call this quantity the channel capacity (or the Shannon capacity of the channel).

As for a rigorous definition, let X be a random variable on Σ, and Y be a random variable on I induced by the channel law p. This means that $\Pr[Y = y|X = x] = p(x, y)$, and hence $\Pr[Y = y] = \sum_{x\in\Sigma} \Pr[Y = y \mid X = x] \Pr[X = x]$. The entropy of X, denoted $H(X)$ is given by

$$H(X) = -\sum_{x\in X} \Pr[X = x] \log_2(\Pr[X = x]),$$

and the conditional entropy of X given Y is

$$H(X|Y) = -\sum_{x\in X, y\in Y} \Pr[X = x, Y = y] \log_2(\Pr[X = x|Y = y]).$$

The mutual information $I(X;Y)$ is defined as $H(X) - H(X|Y)$. Since $H(X) \geq H(X|Y)$, the mutual information is nonnegative. In addition, the mutual information is zero iff $H(X) = H(X|Y)$, and the latter holds iff X and Y are independent. Moreover, a simple manipulation using Bayes rule reveals that $I(X;Y)$ is symmetric in its arguments.

The capacity of the channel $\mathcal{C}$ is defined as $\max_{p(X)} I(X;Y)$, where the maximum is taken over all probability distributions on Σ.

If $\mathcal{C}$ is an input symmetric channel, then it can be shown that the maximum is achieved for the uniform distribution on the input alphabet. In this case, we will have two interesting sets of probability distributions on I associated with $\mathcal{C}$, which can be visualized using the transition matrix $P(\mathcal{C})$: each row of the transition matrix is a probability distribution on I, and since the channel is input symmetric, the rows are permutations of one another, by Remark 1.2. In particular, these probability distributions have the same entropy. Denoting by Π the probability distribution on I induced by the first row of $P(\mathcal{C})$, the common entropy of all the rows of the transition matrix is $H(\Pi)$.

The second distribution is obtained by calculating the marginal distribution on the output symbols, given that we have the uniform distribution on the input. In this case, we take the average of all the rows of the transition matrix, which gives us another distribution on I, and we call this distribution Δ.

Theorem 1.4. *If $\mathcal{C}$ is a discrete memoryless input symmetric channel, then the capacity of $\mathcal{C}$ is $H(\Delta) - H(\Pi)$, where Π and Δ are defined above.*

Proof. Let X be the uniform distribution on Σ and Y be the distribution on I induced by X, i.e., for every $y \in I$, we have $\Pr[Y = y] = \frac{1}{|\Sigma|}\sum_{x\in\Sigma} p(y,x)$. Then, Δ is exactly this probability distribution, and $H(\Delta) = H(Y)$.

How about $H(Y|X)$? Using the uniformity of the distribution of X, we obtain

$$\begin{aligned} H(X|Y) &= -\sum_{x\in\Sigma}\sum_{y\in I} \frac{1}{|\Sigma|} p(y,x)\log_2 p(y,x) \\ &= \frac{1}{|\Sigma|}\sum_{x\in\sigma} H(\Pi) \\ &= H(\Pi), \end{aligned}$$

since for every x, the probability distribution on I given by $p(y,x)$ has the same entropy $H(\Pi)$. □

Example 1.5.

1. $\mathrm{Cap}(\mathrm{BEC}(\varepsilon)) = 1 - \varepsilon$. To see this, we appeal to the transition matrix of $\mathrm{BEC}(\varepsilon)$. In this case, the distribution Π is given as the vector $(1-\varepsilon, \varepsilon, 0)$, where the first entry is the probability of picking the element 0, the second is the probability of E, and the third is the probability of 1. The entropy of this distribution is $h(\varepsilon)$, where $h(x)$ is the binary entropy function. The average of the rows of the transition matrix is $((1-\varepsilon)/2, \varepsilon, (1-\varepsilon)/2)$, which has an entropy equal to $-(1-\varepsilon)\log_2((1-\varepsilon)/2) - \varepsilon\log_2(\varepsilon)$ which in turn is equal to $(1-\varepsilon) + h(\varepsilon)$. In total, the capacity is $H(\Delta) - H(\Pi) = 1 - \varepsilon + h(\varepsilon) - h(\varepsilon) = 1 - \varepsilon$.
2. $\mathrm{Cap}(\mathrm{BSC}(\varepsilon)) = 1 - h(\varepsilon)$, where $h(x) = -x\log_2(x) - (1-x)\log_2(1-x)$ is the binary entropy function. In this case, Π is given by $(1-\varepsilon, \varepsilon)$, and Δ is given by $(1/2, 1/2)$. The entropy of Δ is 1, and that of Π is $h(\varepsilon)$.
3. $\mathrm{Cap}(\mathrm{AWGN}(\sigma)) = 1 - \frac{1}{2\sqrt{\pi m}}\int_{-\infty}^{\infty} \log_2(1+\mathrm{e}^{-x})\mathrm{e}^{-\frac{(x-m)^2}{4m}}\,\mathrm{d}x$, where $m = 2/\sigma^2$. (Note that this is not a consequence of the previous theorem, since this channel is not discrete.)

Example 1.6. We compare the capacity of $\mathrm{AWGN}(\sigma)$ with $\mathrm{BSC}(\varepsilon)$, where ε is the error probability of $\mathrm{AWGN}(\sigma)$. The result is given in Figure 1.2. As can be seen, the capacity of $\mathrm{AWGN}(\sigma)$ is bigger. We often phrase this as saying that "using soft information is better than hard-decision decoding."

Shannon's Channel Coding Theorem shows that the capacity is an achievable upper bound for reliable communication. Its proof requires a simple lemma, the proof of which we leave as an exercise.

Lemma 1.7. *Let n be an integer, and $e \leq n$. Then*

$$\sum_{i=0}^{e}\binom{n}{e} = 2^{nh(e/n)+o(n)},$$

where $h(x)$ is the binary entropy function.

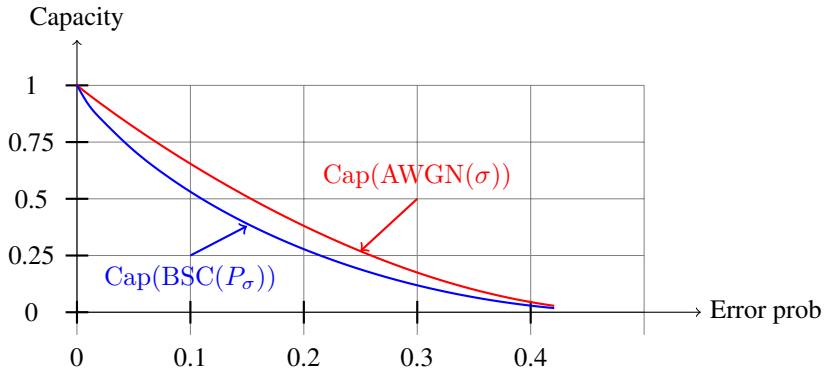


Figure 1.2: Comparison of the capacities of $\mathrm{AWGN}(\sigma)$ and a $\mathrm{BSC}(\varepsilon)$ with equal error probability.

Theorem 1.8 (Channel Coding Theorem for the BSC)**.** *Given a discrete memoryless communication channel $\mathcal{C}$ with input alphabet Σ, then for any $R < \mathrm{Cap}(\mathcal{C})$ there exists a series of codes $C_m \subset \Sigma^{n_m}$ of rate R such that the error probability of the ML-decoder for C_m is upper bounded by $\mathrm{e}^{-\gamma n_m R}$ for some $\gamma > 0$ (depending on the channel) called the error exponent.*

Proof. We will sketch the proof of this theorem for the case $\mathcal{C} = \mathrm{BSC}(\varepsilon)$, $\varepsilon < 1/2$, only. The proof of the general case can be found in any textbook on information theory.

Let $\varepsilon > 0$ be a fixed small real number, and suppose that $R < 1 - h(p + \varepsilon(1-p))$, where h is the binary entropy function. We will show that if n is large enough, then there exists a code C of rate R for which the error probability of the ML-decoder on $\mathrm{BSC}(\varepsilon)$ is at most $2^{-n\gamma R}$ for some $\gamma > 0$ (depending on ε).

Let C be any collection of $M = 2^{nR}$ elements of $\{0,1\}^n$, and denote the elements of C by $x_1, \ldots, x_M$. For $1 \le i \le M$ let P_i denote the probability that the transmission of x_i over $\mathrm{BSC}(\varepsilon)$ leads to a word y which has smaller Hamming distance to some x_j, $j \neq i$, than to x_i. The average error probability of the ML-decoder is then $P_C := \frac{1}{M}\sum_{i=1}^{M} P_i$. Moreover, let $P^*(M,n,p)$ be the minimum of P_C over all possible codes C with M elements. The goal is to upper bound this quantity.

First, note that by Chernoff bounds, there are more than $n\rho$ errors with probability $\le \mathrm{e}^{-n(1-p)\varepsilon^2/2}$, where $:= p + \varepsilon(1-p)$. We modify the ML-decoder, and assume that there is a decoding error as soon as we have more than ρn errors, and otherwise, if there are fewer than ρn errors, we apply the ML-decoding rule. Obviously, the error probability of this decoder is an upper bound on the error probability of ML.

For $x, y \in \{0,1\}^n$ let

$$f(u,v) := \begin{cases} 0, & \text{if } d(u,v) > \rho n \\ 1, & \text{if } d(u,v) \le \rho n \end{cases}.$$

Further, set

$$g_i(u) := 1 - f(u,x_i) + \sum_{j \neq i} f(u,x_j).$$

Further, let $p(y,x_i)$ be the probability that y is received after transmission, given that x_i is transmitted. Then we have

$$\begin{aligned} P_i &\le \sum_y p(y,x_i) g_i(y) \\ &= \sum_y p(y,x_i)(1 - f(y,x_i)) + \sum_y \sum_{j \neq i} p(y,x_i) f(y,x_j). \end{aligned}$$

The first term is upper bounded by $\mathrm{e}^{-n(1-p)\varepsilon^2/2}$ by the aforementioned Chernoff bound. Hence

$$P_C \le \mathrm{e}^{-n(1-p)\varepsilon^2/2} + \frac{1}{M}\sum_{i=1}^{M}\sum_y \sum_{j \neq i} p(y,x_i) f(y,x_j).$$

Now, we choose C at random, by choosing the x_i uniformly and independently at random. For fixed y, we have then

$$\mathrm{E}[p(y,x_i)] = \sum_{x \in \{0,1\}^n} p^{d(x,y)}(1-p)^{n-d(x,y)} = 1.$$

Moreover,

$$\mathrm{E}[f(y,x_j)] = \frac{1}{2^n}\sum_{x\in\{0,1\}^n} f(y,x) = \frac{\sum_{e=0}^{\rho n}\binom{n}{e}}{2^n} = 2^{-n(1-h(\rho))+o(n)},$$

where the last equality follows from Lemma 1.7. Moreover, since the x_i and x_j are chosen independently, we have

$$\mathrm{E}[p(y,x_i)f(y,x_j)] = \mathrm{E}[p(y,x_i)]\mathrm{E}[f(y,x_j)].$$

Altogether, this gives

$$\begin{aligned} P^*(M,n,p) &\leq \mathrm{E}[P_C] \\ &\leq \mathrm{e}^{-n(1-p)\varepsilon^2/2} + (M-1)2^{-n(1-h(\rho))+o(n)} \\ &= \mathrm{e}^{-n(1-p)\varepsilon^2/2} + 2^{-n(1-h(\rho))+R+o(n)}. \end{aligned}$$

A simple manipulation yields the result. □

We remark that random codes of rate R approach the Shannon capacity with high probability. We also remark — without proof — the following converse to the Shannon coding theorem.

Remark 1.9. *For any code of rate $R > \mathrm{Cap}(\mathcal{C})$ the error probability of the ML-decoding is bounded from below by a constant. More precisely, for any sequence of codes of length n and rate $R > \mathrm{Cap}(\mathcal{C})$ there is a constant $\gamma > 0$ such that the error probability of the ML-decoder is at least $1 - \mathrm{e}^{-\gamma n}$.*

The fundamental problems associated with Shannon's coding theorem are:

1. Find codes that approach the Shannon capacity using ML-decoding.
2. Find codes that approach the Shannon capacity using a polynomial time decoding algorithm.

As mentioned above, random codes give an affirmative solution to the first problem (though they are not explicit; more on explicit codes later). A much more difficult problem is finding codes that approach the Shannon capacity arbitrarily closely, but have polynomial time decoding complexity. In a way, this class is about solving the second problem.

In the following, we will give a solution to the second problem for the binary erasure channel.

1.5. Codes Achieving the Capacity of the BEC with Polynomial Decoding Complexity

Throughout this section, $\mathcal{C} = \mathrm{BEC}(\varepsilon)$. The capacity of this channel is $1-\varepsilon$. Let R be smaller than $1-\varepsilon$. We will show the existence of codes of rate R with polynomial time (in the length n of the codewords) decoding complexity over $\mathrm{BEC}(\varepsilon)$ such that the error probability of the decoder is at most $\mathrm{e}^{-\gamma n}$ for some $\gamma > 0$ (depending on R).

Let C be a random subspace of $\mathbb{F}_2^n$ obtained in the following way: choose Rn vectors $x_1, \ldots, x_{Rn}$ of $\mathbb{F}_2^n$ uniformly at random. The subspace C is then $\langle x_1, \ldots, x_{Rn}\rangle$.

Exercise 1.1. Show that for any given δ the dimension of C is larger than $(R-\delta)n$ with probability at least $1-\mathrm{e}^{-c_\delta n}$ for some $c_\delta > 0$.

Let c be an element in C, and let y be the vector obtained from c after transmission through the channel $\mathcal{C}$.

Exercise 1.2. Show that for any $\delta > 0$ the vector y coincides with c on at least $(1-\epsilon-\delta)n$ positions, with probability at least $1-\mathrm{e}^{-\gamma n}$ for some $\gamma > 0$ (Chernoff bounds).

The task of the decoder is to recover c from y. This results to solving a system of linear equations. Let G denote the $Rn \times n$-matrix in which the rows are the vectors $x_1, \ldots, x_{Rn}$. The vector c equals $z \cdot G$ for some $z \in \mathbb{F}_2^n$, and the task is to recover z. Let $i_1, \ldots, i_m$ denote the non-erased positions of c, i.e., those positions in which $y_{i_j} \neq E$. Let G_1 denote the matrix obtained from G by considering only columns $i_1, \ldots, i_m$. Then we have $z \cdot G_1 = \hat{y}$, where $\hat{y}$ is the vector in which the coordinates of y are not E. This is a system of linear equations for z, which is uniquely solvable if G_1 is of rank Rn. Note that G_1 is a random binary matrix, i.e., the entries of G_1 are chosen independently and uniformly from $\{0,1\}$. The error probability of the decoder is then upper bounded by the probability that G_1 is of rank smaller than $(1-\varepsilon)Rn$.

Theorem 1.10. *Let G be a random matrix over $\mathbb{F}_2$ with k rows and n columns, $k < n$. The probability that the rank of G is smaller than k is at most 2^{k-n}.*

Proof. This probability is at most equal to the probability that there exists a nonzero vector z such that $z \cdot G = 0$. For every nonzero z, the probability that $z \cdot G = 0$ is $1/2^n$. Therefore, the expected number of nonzero z such that $z \cdot G = 0$ is $(2^k - 1)/2^n \leq 2^{k-n}$. By the union bound, this is an upper bound on the probability in question. □