Chapter 4

Channel Coding

4.1 The discrete memoryless channel

A communication channel is a pipe which accepts strings of symbols from its input alphabet $\mathcal{A}$ and emits strings of symbols from an output alphabet $\mathcal{B}$. A telephone line, for example, is a communication channel.

Note that a channel must not necessarily extend in space, it can also be viewed in time. A disc drive, for example, may be viewed a channel: writing and read-out takes place at the same location, but at different instances in time. And even reproducing cells, in which the daughter cells' DNA carries information from the parent cells, may be viewed a channel. In fact any structure "input – something may happen – output" may be viewed a communication channel.

A discrete memoryless channel is characterized by an input alphabet $\mathcal{A} = \{a_1, \dots, a_i, \dots, a_A\}$, an output alphabet $\mathcal{B} = \{b_1, \dots, b_\nu, \dots, b_B\}$, and a *channel matrix* $p_{\nu i} \geq 0, 1 \leq i \leq$

$A, 1 \leq \nu \leq B$ with

$$\sum_{\nu=1}^{B} p_{\nu i} = 1\,, \qquad \text{for all } i = 1, \ldots, A\,. \tag{4.1}$$

Such matrix a is called *stochastic*. The transition matrix in a Markov chain, for example, is a stochastic matrix. Note that we do not require $p_{\nu i}$ to be double stochastic, i. e. we do not require $\sum_i p_{\nu i} = 1$. Examples of channels are displayed in Figs. 4.1–4.3. The channel matrix of the binary symmetric channel is double stochastic. The channel matrices of the binary erasure channel and binary decay channel (also called Z-channel) are *not* double stochastic.

Connecting the input source X and output sink Y to the channel, it may be viewed a *joint ensemble* denoted $Z = XY$. Elementary events, denoted z, are from the set $\mathcal{Z} = \{(x, y) | x \in \mathcal{A}, y \in \mathcal{B}\}$, where $z = (x, y)$ denotes the joint event "x sent and y received". Obviously, in order to make communication possible, there must be *correlations* between x and y. Technically: if $p(x)$ $(q(y))$ are the marginal probabilities for the sub-events "x sent" ("y received"), the probability for the joint event "x sent and y received" should not in general factorize, $\exists (x, y) : p(x, y) \neq p(x)q(y)$.

4.2 Mutual Information

Upon injection of a sequence of input symbols $x_1 x_2 \ldots x_t \ldots$, the channel produces a sequence of output symbols $y_1 y_2 \ldots y_t \ldots$. The channel being memoryless now means that at each instance of use, t, the *conditional* probability that given $x_t = a_i$ is injected, the probability that $y_t = b_\nu$ is received does not depend on t, and is given by

$$P(y_t = b_\nu | x_t = a_i) = p_{\nu i}\,. \tag{4.5}$$

Given a memoryless source (Alice) which emits symbols $a_i \in \mathcal{A}$ with relative frequency p_i, the joint probability for the event "a_i send and b_μ received" is given by

$$P(b_\nu, a_i) = P(b_\nu|a_i)p_i \,. \tag{4.6}$$

The probability for the event "b_μ received", irrespective of what symbol was actually send, is then given by

$$q_\nu = \sum_i P(b_\nu|a_i)p_i \,. \tag{4.7}$$

Finally, we may define

$$Q(x_t = a_i|y_t = b_\nu) \equiv P(b_\nu, a_i)/q_\nu \,, \tag{4.8}$$

which according to Baye[1] is the likelihood that given b_ν is received, a_i was actually send.

The four probabilities give rise to various types of entropies. First, there is the source Shannon entropy,

$$H(X) = -\sum_{i=1}^{A} p_i \mathrm{ld}(p_i) \,, \tag{4.9}$$

which measures the average information carried by each symbol injected into the channel.

Second, there is the entropy of the channel output, which – after all – is just a source

[1] Rev. Thomas Baye, the founding father of statistical inference, addressed (and solved) in 1763 the following problem: given an *observed* sequence of the results of tossing a coin N times – what is the most likely bias of the coin, and what is the probability that the next toss will display a head, say?

Y from the receiver's point of view,

$$H(Y) = -\sum_{\nu=1}^{B} q_\nu \mathrm{ld} q_\nu \,. \tag{4.10}$$

It may not be viewed the average information pouring out at the output, as the output may contain unwanted contributions from channel noise.

Third, we have

$$H(Y|a_i) = -\sum_{\nu=1}^{B} P(b_\nu|a_i)\mathrm{ld}P(b_\nu|a_i)\,, \tag{4.11}$$

which measures the uncertainty about the value of a received symbol y_t before it is measured, given one knows that a_i was actually send. For an ideal channel $H(Y|a_i) = 0$ as the uncertainty removed by a symbol whose value is known ($y_t = x_t$!) is zero.

Fourth, averaging (4.11) with respect to the source,

$$\begin{aligned} H(Y|X) &= -\sum_{i=1}^{A} H(Y|a_i)p_i & (4.12)\\ &= \sum_{i,\nu}^{A,B} P(a_i, b_\nu)\mathrm{ld}\frac{1}{P(b_\nu|a_i)}\,, & (4.13) \end{aligned}$$

we obtain the *conditional Shannon information*: How much uncertainty about y_t would remain on average, if we would first learn the value of x_t, before we actually look at y_t.

Finally

$$I(Y:X) = H(Y) - H(Y|X) \tag{4.14}$$

is a good measure of the information successfully transmitted from input to output. It is called *mutual information*, *synentropy*, or *transinformation* and plays an important role in information theory.

Theorem The mutual information is symmetric

$$I(Y:X) = H(Y) - H(Y|X) = H(X) - H(X|Y) = I(X:Y)\,, \tag{4.15}$$

i.e. it is quite rightfully called “mutual”.

The proof is elementary. Recall $P(x,y) = P(y|x)p(x) = Q(x|y)q(y)$, and wit

$$\begin{aligned} I(Y:X) &= H(Y) - H(Y|X) && (4.16)\\ &= \sum_y q(y)\mathrm{ld}\frac{1}{q(y)} - \sum_{xy} P(x,y)\mathrm{ld}\frac{1}{P(y|x)} && (4.17)\\ &= \sum_{xy} P(x,y)\mathrm{ld}\frac{P(y|x)}{q(y)} && (4.18)\\ &= \sum_{xy} P(x,y)\mathrm{ld}\frac{P(x,y)}{p(x)q(y)}\,. && (4.19) \end{aligned}$$

This expression is symmetric in x and y. *End-of-proof*

Theorem The mutual information obeys the inequalities

$$0 \le I(Y:X) \le H(X)\,, \tag{4.20}$$

with $I(Y:X) = 0$ iff Alice and Bob are statistically independent (maximally noisy channel), and $I(Y:X) = H(X)$ iff Alice and Bob are maximally correlated (ideal channel).

The proof is simple. By virtue of Eq. (4.19), the mutual information is in fact a relative entropy, hence it is non-negative – see Eq. (2.17). To prove the second inequality recall that for ideal channels the channel matrix is the identity, and therefore the conditional Shannon entropy is zero. *End-of-proof*

Information can disappear, but it can not spontaneously be born. This is the morale of the

Theorem The mutual information obeys the *data processing inequality*

$$\text{for serial } W \to X \to Y : \quad I(Y:W) \leq I(X:W)\,. \tag{4.21}$$

The proof is simple. The serial process $W \to X \to Y$ defines a Markov chain with joint probabilities $P(w,x,y) = P(y|x)P(x|w)p(w)$. *End-of-proof*

Nothing is forever. Even if you engrave your work in stone – it will turn into dust, leaving no trace, neither of you nor of your existence.[2]

4.3 Channel Capacity, Rates and Errors

The capacity of a channel is defined

$$C := \max_{\mathbf{p}} I(Y:X)\,, \tag{4.22}$$

where the maximum is taken over all input distributions **p**. It is measured in data bits transmitted per use of channel.

[2] These days information fades much more quickly than in the good old days. Some time ago, when I actually found a proof of the Goldbach conjecture, I stored it on a weird-format floppy disc (which came with a SchneiderComputer in the '80s), waiting for a better occasion to reveal my findings. But alas – no device exists anymore which can read this thing! So I threw it away ...

The capacity of the binary symmetric channel, for example, in which there is probability p_b of a bit-flip, is given by

$$C(p_\mathrm{b}) = 1 - H_2(p_\mathrm{b})\,, \tag{4.23}$$

where $H_2(\cdot)$ is the binary entropy function.

If you transmit a signal bit $s \in \{\mathtt{0}, \mathtt{1}\}$ over a binary symmetric channel, the transmission rate

$$R = \frac{\text{data bits transmitted}}{\text{use of channel}} \tag{4.24}$$

is maximal, $R = 1$, but as the channel flips a cebit with probability p_b, you will make, with probability $e = p_\mathrm{b}$, an error in decoding the received cebit.

The chances of decoding error can be reduced, of course, by just repeating the transmission. Use the encoding $f(\mathtt{0}) = \mathtt{000}$, $f(\mathtt{1}) = \mathtt{111}$ (called repetition code R_3), and transmit not the signal bit s, but the codeword $\mathbf{t}^{(s)} = f(s)$ instead. At the other end of the channel, decode the received block of three bits, which will in general differ from the transmitted codeword, using majority voting. The probability to decode incorrectly is now given by $e = p_\mathrm{b}^3 + 3p_\mathrm{b}^2(1 - p_\mathrm{b})$. SInce $e < p_\mathrm{b}$, the repetition code R_3 gives rise to less decoding error probability than the simple code – which is good news. The bad news is, that the rate R has been reduced, $R = 1/3 < 1$.

Extending the argument to longer blocks, one is tempted to believe that error free communication over a noisy channel is only possible at zero rate – i.e. error free communication is actually *impossible*. In fact, that was the widespread belief until Shannon in his 1948 landmark paper proved that this is not the case.

4.4 Channel coding theorem

Theorem Given a binary symmetric channel of capacity C and any R, with $0 < R < C$, then, if $(M_L : 1 \leq L < \infty)$ is any sequence if integers satisfying

$$1 \leq M_L \leq 2^{LR}\,, \qquad 1 \leq L < \infty \tag{4.25}$$

and δ any positive quantity, there exists a sequence of codes $(\mathcal{C}_L : 1 \leq L < \infty)$ and an integer $L_0(\delta)$ with $\mathcal{C}_L$ having M_L codewords of length L and with maximum error probability $\hat{e}(\mathcal{C}_L) \leq \delta$ for all $L \geq L_0(\delta)$.

In a nutshell: there exist codes with finite rate R, $R < C$, which allow for communication over a noisy channel of capacity C with arbitrarily small probability of decoding error. How small – well, that will essentially depend on the code length you can afford: the longer, the less probable a decoding error.

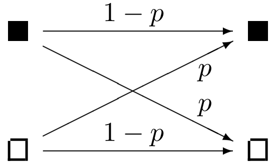
For a sound proof of the theorem see Dominic Welsh *Codes and Cryptography* – or any book on information theory. Here we give only the idea of the proof.

Consider an extended channel Z^L, which has A^L possible input words $\mathbf{x}$ and $B^L = 2^{LH_0(Y)}$ possible output words $\mathbf{y}$. The key observation is that, if L is large, any particular input produces, with high probability, an output in a small subspace of $\mathcal{B}^L$. The strategy is to identify a set of input words whose corresponding output sets display negligeable overlap. This set of non-confusable input words defines a code – a block code of length L. The task is to maximize this set.

For L large, the output will be, with high probability, one of $\approx 2^{LH(Y)}$ words, called typical $\mathbf{y}$. For any particular input $\mathbf{x}$ – which will also be typical with high probability – the output will be one of $2^{LH(Y|X)}$ words. So the typical-$\mathbf{y}$ set contains $2^{LH(X)}$ typical-$\mathbf{y}$-given-typical-$\mathbf{x}$ sets, each of size $2^{LH(Y|X)}$. Since $H(X)+H(Y|X) \geq H(Y)$, the typical-$\mathbf{y}$-given-typical-$\mathbf{x}$ sets will overlap, which will lead to confusion in decoding. But thinning out the set of typical $\mathbf{x}$ we are going to send, the number of typical-$\mathbf{y}$-given-typical-$\mathbf{x}$ sets will decrease, until finally they do not overlap anymore. This will be reached if the set of typical-$\mathbf{y}$ contains of the order of $2^{LH(Y)-LH(Y|X)}$ typical-$\mathbf{y}$-given-typical-$\mathbf{x}$ sets. Recall we learn that the number of non-confusable inputs is $\leq 2^L I(Y:X)$. The maximum of this bound is realized for a $\mathbf{p}$ which maximizes $I(Y:X)$, in which case the number of non-confusable inputs – the codewords – is $\leq 2^{LC}$. In order to fully transmit a code word, the channel must be used L times. Each time, the number of conveyed bits is $\leq C$, i.e. the rate of the code is $R \leq C$.

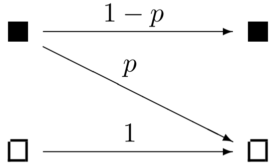
So far, the arguments are based on fixed-length block codes of large size L. The scheme could be implemented as follows. Wait till the source has produced a sequence $\mathbf{s}$ of $LC/\mathrm{ld}(A)$ symbols (the signal), encode this block into an associated codeword $\mathbf{t}^{(\mathbf{s})}$ of length L, transmit $\mathbf{x} = \mathbf{t}^{(\mathbf{s})}$, and at the receivers side decode by using a large look-up table in order to identify the codeword, and thus the message, associated with the received $\mathbf{y}$. In the limit $L \to \infty$, the probability for a decoding error goes to zero, but for every-day communication purposes this scheme is clearly unpractical.

One rather accepts a non-zero probability of decoding error e, for which one tries to identify a reasonably sized code and efficient decoding algorithm, such that the rate R is as large as possible. These kinds of codes are called error-correction codes. The most frequent type of error correction codes are linear binary block codes. The repetition code R_3 is of such type, but also the so called $[L, K]$ Hamming codes – see the Supplement B.4.



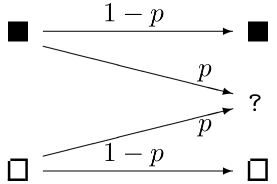
$$(p_{\nu i}) = \begin{pmatrix} 1-p & p \\ p & 1-p \end{pmatrix} . \qquad (4.2)$$

Figure 4.1: The binary symmetric channel



$$(p_{\nu i}) = \begin{pmatrix} 1 & p \\ 0 & 1-p \end{pmatrix} . \qquad (4.3)$$

Figure 4.2: The binary decay channel



$$(p_{\nu i}) = \begin{pmatrix} 1-p & 0 \\ p & p \\ 0 & 1-p \end{pmatrix} . \qquad (4.4)$$

Figure 4.3: The binary erasure channel