Chapter 7

Channel Capacity

In Section 5.2 the block diagram of a communication system in Figure 5.1 was discussed. It was stated that the source coding and the channel coding can be separated. In Chapters 5 and 6 the source coding was treated. Here we will concentrate on the analysis of the channel coding system.

In Figure 7.1 a block diagram is shown for the channel encoder, the channel and the channel decoder. The channel in this figure is a mathematical model that represents everything that can occur in the actual transmission. This includes e.g. background noise, scratches on the surface of a CD or erasures due to overflow in router buffers.

Figure 7.1: A model of the channel coding system.

A little more formal description of the encoding scheme is as follows.

- The symbols U are assumed to be taken from an index set $\mathcal{I} = \{1, 2, \ldots, M\}$.
- The encoding function $x : \mathcal{I} \to \mathcal{X}$ is a mapping from the index set $\mathcal{I}$ to the set of codewords $\mathcal{X}$. We can denote the codewords as $x(1), x(2), \ldots, x(M)$. In the most cases in this text, we will assume that the codewords are binary vectors of length k, $\mathcal{X} \in \{0, 1\}^n$.
- As the codeword is transmitted over the channel, errors occur and the received vectors are $y \in \mathcal{Y}$. In many situations the received symbols are taken from a larger alphabet than the code symbols. It can be that $\mathcal{Y}$ are real values detected by the receiver, while the code symbols $\mathcal{X}$ are binary.
- The decoding function is then a (typically non-linear) mapping from the received word to an estimate of the transmitted codeword or the actual index set, $g : \mathcal{Y} \to \mathcal{I}$.

The code above is an (M, n) code. In this environment the code rate is defined as $R =$

$\frac{\log_{|\mathcal{X}|} M}{n} = \frac{k}{n}$. If we are using binary vectors as codewords the logarithm function here is over base 2.

An important measure of the efficiency of code is error probability, i.e. the probability that the decoder makes an erroneous decision,

$$P_e = P\big(g(Y) \neq i | X = x(i)\big)$$

This probability will never be zero, as long as the probability for the channel to change one codeword into another is nonzero. When receiving a codeword the most likely event is that the transmission was error free, and the estimate of the index should correspond to this codeword. However, the error probability can be arbitrarily small, and we say that a code rate is *achievable* if it tends to zero for large n,

$$P_e \to 0, \quad n \to \infty$$

An important result for bounding the error probability is given in Fano's Lemma.

Lemma 35 (Fano's Lemma) *Let U and $\widehat{U}$ be two random variables with the same alphabet of size L, and $P_e = P(\widehat{U} \neq U)$ the error probability. Then*

$$h(P_e) + P_e \log(L-1) \geq H(U|\widehat{U})$$

□

To see this we first introduce a random variable Z that describes the error,

$$Z = \begin{cases} 0, & U = \widehat{U} \\ 1, & U \neq \widehat{U} \end{cases}$$

With this, we first notice that

$$H(UZ|\widehat{U}) = H(U|\widehat{U}) + H(Z|U\widehat{U}) = H(U|\widehat{U})$$

since given both U and $\widehat{U}$, there is no uncertainty about Z. Then

$$\begin{aligned} H(U|\widehat{U}) &= H(UZ|\widehat{U}) = H(Z|\widehat{U}) + H(U|\widehat{U}Z) \\ &\leq H(Z) + H(U|\widehat{U}Z) \\ &= \underbrace{H(Z)}_{h(P_e)} + \underbrace{H(U|\widehat{U}Z=0)}_{=0} P(Z=0) + \underbrace{H(U|\widehat{U}Z=1)}_{\leq \log(L-1)} \underbrace{P(Z=1)}_{P_e} \\ &\leq h(P_e) + P_e \log(L-1) \end{aligned}$$

where we used that conditioned on $\widehat{U}$ and $Z = 1$, U can take only $L-1$ values and the entropy is upper bounded by $\log(L-1)$.

To get an understanding of the interpretation of the lemma we first need to plot the function. To do that we introduce the left hand side as a function

$$F(p) = h(p) + p \log(L-1)$$

and take the derivative to find the optima,

$$\frac{\partial}{\partial p}F(p) = \log\frac{1-p}{p} + \log(L-1) = 0$$

which gives an optima for $p = \frac{L-1}{L}$ where $F(\frac{L-1}{L}) = \log L$. At the end points we have $F(0) = 0$ and $F(1) = \log(L-1)$. To see that this is really a maximum we have found we take the derivative ones again to get

$$\frac{\partial^2}{\partial p^2}F(p) = \frac{1}{p(p-1)\ln 2} \leq 0, \quad 0 \leq p \leq 1$$

Since the second derivative is negative in the whole interval we conclude that the function is concave, hence it must be a global maximum. In Figure 7.2 the function is plotted. We notice now that the function is non-negative and that it spans the same interval as the right hand side of the lemma,

$$0 \leq H(U|\widehat{U}) \leq \log L$$

which we interpret as the uncertainty we have about the transmitted symbol U when we receive $\widehat{U}$. Therefore, whatever value we get on this uncertainty we can map it to a positive value or interval for the error probability. The only time we can actually get a zero error probability is when the uncertainty is zero, i.e. when we have either no disturbances on the channel or completely known disturbances. And this is never the case in a real situation.

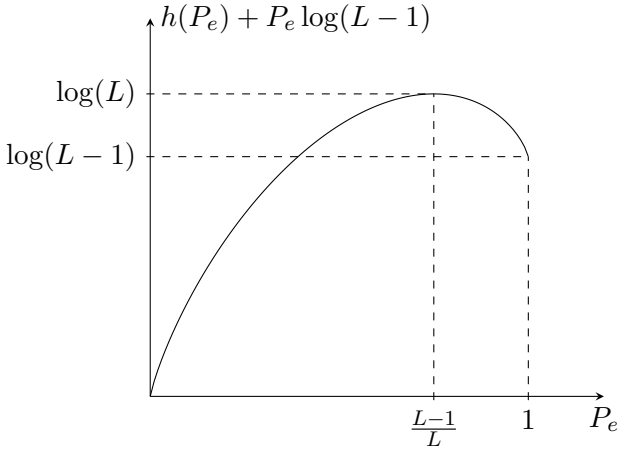


Figure 7.2: A plot of the left hand side of Fano's Lemma.

In the next section we will introduce the channel coding theorem, which gives the necessary and sufficient condition for when it is possible to get arbitrarily small P_e.

In the next section we will see that it is possible for a given channel to give a bound on how much information is possible to transmit.

7.1 Channel Coding Theorem

Shannon defined communication as transmitting information from one place and time to another place and time. This describes a lot of scenarios, for example a telephone call,

recording and playing a CD, DVD or USB stick, but also a normal face to face conversation or this text. In all of those scenarios there is a probability of errors along the actual transmission. In the case of telephone call it can be disturbances along the line, for CD and DVD it can dust on the lens or scratches on the surface, for a normal conversation there are typically background noise that is sorted out by our brains and in this text there are surly typos. The *channel* of the transmission is a statistical representation of all these disturbances. There are of course many different types of channels and they can be made arbitrarily complicated depending on the level of modelling the reality. In this chapter we will concentrate on memory-less discrete channels.

Definition 20 *A **discrete channel** is a system $(\mathcal{X}, P(y|x), \mathcal{Y})$, with*

- *input alphabet $\mathcal{X}$*
- *output alphabet $\mathcal{Y}$*
- *transition probability distribution $P(y|x)$*

*The channel is **memory-less** if the probability distribution is independent of previous input symbols.* □

In Figure 7.3 a discrete memory-less channel is shown.

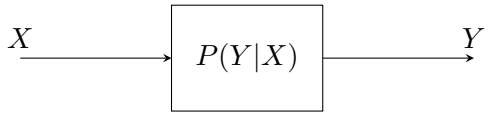


Figure 7.3: A discrete memory-less channel.

One of the most known, an used, channels is described in the next example.

Example 44 [BSC] One of the most well used discrete memory-less channels (DMC) is the binary symmetric channel (BSC). It is characterised by that both the input and the output alphabets are binary, $\mathcal{X} = \{0, 1\}$ and $\mathcal{Y} = \{0, 1\}$, and that the transition probabilities are symmetric, i.e. they can be described by the table

		Y	
$p(y\|x)$		0	1
X	0	$1-p$	p
	1	p	$1-p$

Hence, the probability for having an error, i.e. transmitting 0 and receiving 1 or transmitting 1 and receiving 0, is p and the probability for no error, i.e. receiving the same as receiving, is $1 - p$. The symmetry means that the probability of error is independent of transmitted symbol. The channel can be viewed graphically as in Figure 7.4

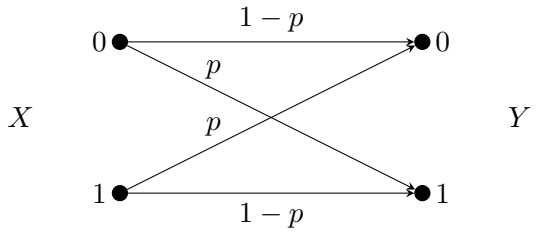


Figure 7.4: A graphical interpretation of the BSC.

A measure of the amount of information that is possible to transmit over a channel can obtained by the mutual information between the transmitted variable X and the received variable Y, $I(X;Y)$. In (almost) all communication situations we are interested in transmitting as much information as possible. This maximum we define as the *channel capacity*.

Definition 21 *The information **channel capacity** of a discrete memory-less channel is*

$$C = \max_{p(x)} I(X;Y)$$

where the maximum is taken over all input distributions. □

Example 45 For the binary symmetric channel the mutual information between transmitter and receiver side is

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) = H(Y) - \sum_{i=0}^{1} P(X=i)H(Y|X=i) \\ &= H(Y) - \sum_{i=0}^{1} P(X=i)h(p) = H(Y) - h(p) \leq 1 - h(p) \end{aligned}$$

where we have equality if and only if $P(Y=0) = P(Y=1) = \frac{1}{2}$. The probabilities of Y can be derived from the probabilities of X as

$$\begin{aligned} P(Y=0) &= (1-p)P(X=0) + pP(X=1) \\ P(Y=0) &= pP(X=0) + (1-p)P(X=1) \end{aligned}$$

From symmetry we see that $P(Y=0) = P(Y=1) = \frac{1}{2}$ is equivalent to $P(X=0) = P(X=1) = \frac{1}{2}$. That means the capacity for this channel can be derived as

$$C = \max_{p(x)} I(X;Y) = 1 - h(p)$$

where the maximising distribution of X is when $P(X=0) = P(X=1) = \frac{1}{2}$. In Figure 7.5 the capacity is plotted as a function of the error probability p.

We see in the figure that the capacity is 0 for $p = 1/2$. It means that with equal probabilities on the channel we have no information about the transmitted symbol from the received symbol. That is, when trying to estimate the transmitted symbol, we are not helped by the received Y. On the other hand, if $p = 0$ or $p = 1$ there are no uncertainty about the transmitted symbol if given the received Y. Then we can transfer 1 bit per transmission, or channel use.

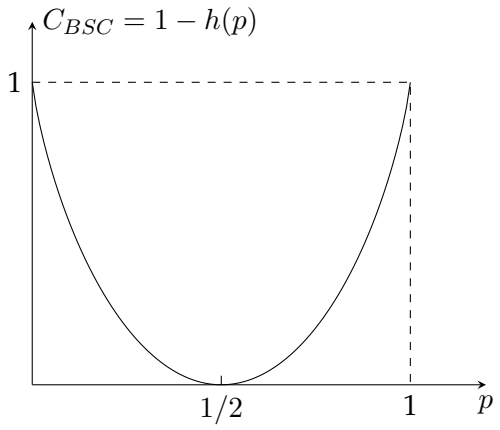


Figure 7.5: Capacity of a BSC as a function of the error probability p.

To upper bound the capacity we upper bound the mutual information,

$$I(X;Y) = H(Y) - H(Y|X) \leq H(Y) \leq \log|\mathcal{Y}|$$

Similarly we get $I(X;Y) \leq \log|\mathcal{X}|$. To summarize the capacity can be bounded as in the next theorem.

Theorem 36 *For a DMC $(\mathcal{X}, P(y|x), \mathcal{Y})$, the channel capacity is bounded by*

$$0 \leq C \leq \min\{\log|\mathcal{X}|, \log|\mathcal{Y}|\}$$

□

To be able to continue on to Shannon's channel coding theorem we first need to extend the ideas from AEP and typical sequences to *jointly typical sequences*. The idea is to consider a sequence of pairs of X and Y and say that each of the sequences should be typical and that the sequence of pairs, viewed as a random variable, should also be typical. Then we get a set of sequences of pairs (X, Y) that are the most likely to actually happen. These pairs can be used for channel coding and it can be derived what code rates are achievable.

Definition 22 *The set $A_\varepsilon^{(n)}(X,Y)$ of **jointly typical** sequences $(\boldsymbol{x}, \boldsymbol{y})$ of length n with respect to the distribution $p(x,y)$ is the set length n sequences*

$$\boldsymbol{x} = (x_1, x_2, \ldots, x_n) \quad \textit{and} \quad \boldsymbol{y} = (y_1, y_2, \ldots, y_n)$$

such that each of $\boldsymbol{x}$, $\boldsymbol{y}$ and $(\boldsymbol{x}, \boldsymbol{y})$ are ε-typical, i.e.

$$\left|-\frac{1}{n}\log p(\boldsymbol{x}) - H(X)\right| \leq \varepsilon,$$
$$\left|-\frac{1}{n}\log p(\boldsymbol{y}) - H(Y)\right| \leq \varepsilon,$$
$$\left|-\frac{1}{n}\log p(\boldsymbol{x}, \boldsymbol{y}) - H(X,Y)\right| \leq \varepsilon$$

where $p(\boldsymbol{x}, \boldsymbol{y}) = \prod_i p(x_i, y_i)$. □

In Figure 7.6 the big rectangle represents the set of all sequences $(\boldsymbol{x}, \boldsymbol{y})$. Then we gather all sequences where $\boldsymbol{x}$ is typical by it self to the left in the rectangle, and all sequences where $\boldsymbol{y}$ is typical by it self to the upper part of the rectangle. In this way we get the light gray shaded rectangle containing only sequences where either $\boldsymbol{x}$ or $\boldsymbol{y}$ are typical. Among those we find the sequences where also the sequence of the pair is typical, which is represented by the dark gray area.

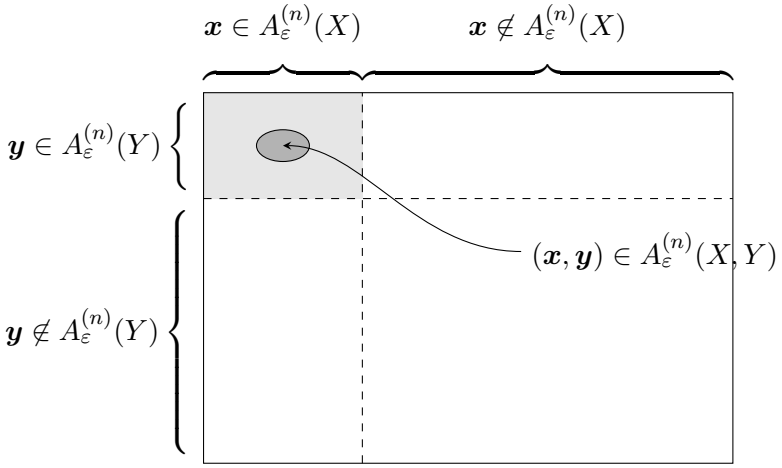


Figure 7.6: The sets of jointly typical sequences.

As in Chapter 5, the definition for ε typical sequences can be given in an alternative form.

Definition 23 (Equivalent definition) *The set $A_\varepsilon^{(n)}(X, Y)$ of **jointly typical** sequences $(\boldsymbol{x}, \boldsymbol{y})$ of length n with respect to the distribution $p(x, y)$ is the set length n sequences such that*

$$\begin{aligned} 2^{-n(H(X)+\varepsilon)} &\leq p(\boldsymbol{x}) \leq 2^{-n(H(X)-\varepsilon)} \\ 2^{-n(H(Y)+\varepsilon)} &\leq p(\boldsymbol{y}) \leq 2^{-n(H(Y)-\varepsilon)} \\ 2^{-n(H(X,Y)+\varepsilon)} &\leq p(\boldsymbol{x}, \boldsymbol{y}) \leq 2^{-n(H(X,Y)-\varepsilon)} \end{aligned}$$

□

As for the case with typical sequences in Chapter 5 we want to show that the set of typical sequences is a small fraction of all sequences but their probability is close to one. The following theorem corresponds to Theorem 20.

Theorem 37 *Let $(\boldsymbol{X}, \boldsymbol{Y})$ be sequences of length n drawn iid according to $p(\boldsymbol{x}, \boldsymbol{y}) = \prod_i p(x_i, y_i)$. Then, for sufficiently large n,*

1. $P\Big((\boldsymbol{x}, \boldsymbol{y}) \in A_\varepsilon(X, Y)\Big) \geq 1 - \varepsilon$

2. $(1 - \varepsilon)2^{n(H(X,Y)-\varepsilon)} \leq |A_\varepsilon(X, Y))| \leq 2^{n(H(X,Y)+\varepsilon)}$

3. *Let $(\tilde{\boldsymbol{X}}, \tilde{\boldsymbol{Y}})$ be distributed according to $p(\boldsymbol{x})p(\boldsymbol{y})$, i.e. $\tilde{\boldsymbol{X}}$ and $\tilde{\boldsymbol{Y}}$ are independent with the same marginals as $p(\boldsymbol{x}, \boldsymbol{y})$. Then*

$$(1-\varepsilon)2^{-n(I(X;Y)+3\varepsilon)} \leq P\Big((\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \in A_\varepsilon(X,Y)\Big) \leq 2^{-n(I(X;Y)-3\varepsilon)}$$

□

To show the first part of the theorem we use the weak law of large number and state that there exists an n_1 such that for $n \geq n_1$

$$P_1 = P\Big(\big|-\tfrac{1}{n}\log p(\boldsymbol{x}) - H(X)\big| > \varepsilon\Big) < \frac{\varepsilon}{3}$$

Similarly, there exists an n_2 such that for $n \geq n_2$

$$P_2 = P\Big(\big|-\tfrac{1}{n}\log p(\boldsymbol{y}) - H(Y)\big| > \varepsilon\Big) < \frac{\varepsilon}{3}$$

and there exists an n_3 such that for $n \geq n_3$

$$P_3 = P\Big(\big|-\tfrac{1}{n}\log p(\boldsymbol{x}, \boldsymbol{y}) - H(X,Y)\big| > \varepsilon\Big) < \frac{\varepsilon}{3}$$

Then, for $n \geq \max\{n_1, n_2, n_3\}$ we get

$$\begin{aligned} P\big((\boldsymbol{x}, \boldsymbol{y}) \in A_\varepsilon^{(n)}(X,Y)\big) &= P\Big(\big|-\tfrac{1}{n}\log p(\boldsymbol{x}) - H(X)\big| > \varepsilon \\ &\qquad \cup \big|-\tfrac{1}{n}\log p(\boldsymbol{y}) - H(Y)\big| > \varepsilon \\ &\qquad \cup \big|-\tfrac{1}{n}\log p(\boldsymbol{x}, \boldsymbol{y}) - H(X,Y)\big| > \varepsilon\Big) \\ &\leq P_1 + P_2 + P_3 < \varepsilon \end{aligned}$$

where we used the union bound[1] in the second last inequality, which shows the first part.

For the second part we use the same argument as for the case in Chapter 5. The right hand side of the inequality can be shown by

$$1 \geq \sum_{\boldsymbol{x},\boldsymbol{y} \in A_\varepsilon^{(n)}} p(\boldsymbol{x}, \boldsymbol{y}) \geq \sum_{\boldsymbol{x},\boldsymbol{y} \in A_\varepsilon^{(n)}} 2^{-n(H(X,Y)+\varepsilon)} = |A_\varepsilon^{(n)}| 2^{-n(H(X,Y)+\varepsilon)}$$

The left hand side can be shown, for sufficiently large n, by

$$1 - \varepsilon \leq \sum_{\boldsymbol{x},\boldsymbol{y} \in A_\varepsilon^{(n)}} p(\boldsymbol{x}, \boldsymbol{y}) \leq \sum_{\boldsymbol{x},\boldsymbol{y} \in A_\varepsilon^{(n)}} 2^{-n(H(X,Y)-\varepsilon)} = |A_\varepsilon^{(n)}| 2^{-n(H(X,Y)-\varepsilon)}$$

[1] The union bound states that for the events $\mathcal{A}_1, \ldots, \mathcal{A}_n$ the probability that at least one is true is

$$P\Big(\bigcup_{i=1}^{n} \mathcal{A}_i\Big) \leq \sum_{i=1}^{n} P(\mathcal{A}_i)$$

For $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}})$ distributed according to $p(\boldsymbol{x})p(\boldsymbol{y})$, where $p(\boldsymbol{x}) = \sum_{\boldsymbol{y}} p(\boldsymbol{x}, \boldsymbol{y})$ and $p(\boldsymbol{y}) = \sum_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{y})$, respectively, we get the probability of $(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{x}})$ being a jointly typical sequence

$$\begin{aligned} P\big((\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \in A_\varepsilon^{(n)}(X,Y)\big) &= \sum_{\boldsymbol{x},\boldsymbol{y} \in A_\varepsilon^{(n)}} p(\boldsymbol{x})p(\boldsymbol{y}) \leq \sum_{\boldsymbol{x},\boldsymbol{y} \in A_\varepsilon^{(n)}} 2^{-n(H(X)-\varepsilon)} 2^{-n(H(Y)-\varepsilon)} \\ &\leq 2^{-n(H(X)-\varepsilon)} 2^{-n(H(Y)-\varepsilon)} 2^{n(H(X,y)+\varepsilon)} \\ &= 2^{-n(H(X)-H(Y)-H(X,Y)+-3\varepsilon)} \\ &= 2^{-n(I(X;Y)-3\varepsilon)} \end{aligned}$$

to get the right hand side. In the same manner, the left hand side can be obtained by

$$\begin{aligned} P\big((\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{y}}) \in A_\varepsilon^{(n)}(X,Y)\big) &= \sum_{\boldsymbol{x},\boldsymbol{y} \in A_\varepsilon^{(n)}} p(\boldsymbol{x})p(\boldsymbol{y}) \geq \sum_{\boldsymbol{x},\boldsymbol{y} \in A_\varepsilon^{(n)}} 2^{-n(H(X)+\varepsilon)} 2^{-n(H(Y)+\varepsilon)} \\ &\geq 2^{-n(H(X)+\varepsilon)} 2^{-n(H(Y)+\varepsilon)} (1-\varepsilon) 2^{n(H(X,y)-\varepsilon)} \\ &= (1-\varepsilon) 2^{-n(H(X)-H(Y)-H(X,Y)+3\varepsilon)} \\ &= (1-\varepsilon) 2^{-n(I(X;Y)+3\varepsilon)} \end{aligned}$$

We are now ready to state the channel coding theorem.

Theorem 38 (Channel coding theorem) *All rates below capacity are achievable. Specifically, for every rate*

$$R < C = \max_{p(x)} I(X;Y)$$

there exists a code with probability of error approaching zero. Conversely, any code with probability of error approaching zero must have $R \leq C$. □

What this theorem means is that if we consider a discrete memory-less channel with capacity C, and we are using a $(2^{nR}, n)$ code, where R is the code rate (information symbols per code symbols). Then

- If $R < C$ it is possible to find a code such that the error probability is arbitrarily low.
- If $R > C$ it is not possible to find a code such that the error probability is arbitrarily low.

The proof of the theorem is quite extensive. Before we go into details of this we first give an intuitive explanation of it. In Figure 7.7 the sets $\mathcal{X}$ and $\mathcal{Y}$ are shown. The transmitted codeword is taken from $\mathcal{X}$ and the received symbol from $\mathcal{Y}$. The decoding rule is decided by decoding regions. If the received symbol $\boldsymbol{y}$ is in the decoding region for $\boldsymbol{x}$, $\mathcal{D}(\boldsymbol{x})$, which is the gray area in the figure, the estimated transmitted codeword is $\boldsymbol{x}$. One way to set up these decoding regions is to say that given a received symbol $\boldsymbol{y}$ find an $\boldsymbol{x}$ such that they are jointly typical, $(\boldsymbol{x}, \boldsymbol{y}) \in A_\varepsilon^{(n)}(X,Y)$. If such $\boldsymbol{x}$ exists and is unique decode to $\boldsymbol{x}$.

Since we use typical sequences in the decoding we can directly say that (neglecting the ε) the number of such mappings is $|(\boldsymbol{x}, \boldsymbol{y})| = 2^{nH(X,Y)}$. On the other hand the number of typical transmitted sequences is $|\boldsymbol{x}| = 2^{nH(X)}$. That means, in each decoding region $\mathcal{D}(\boldsymbol{x})$ the number of possible received sequences is

$$N = |\{\boldsymbol{y} \in \mathcal{D}(\boldsymbol{x})\}| = \frac{2^{nH(X,Y)}}{2^{nH(X)}} = 2^{n(H(X,Y)-H(X))}$$

But, the number of typical received sequences is $|\boldsymbol{y}| = 2^{nH(Y)}$. To have the decoding regions as disjoint sets we therefore should have the number of decoding regions, and therefore also the number of codewords, to

$$|\mathcal{C}| = \frac{|\boldsymbol{y}|}{|\mathcal{D}(\boldsymbol{x})|} = 2^{n(H(X,Y)-H(X)-H(Y))} = 2^{nI(X;Y)}$$

that is, the optimal code rate is $R = I(X;Y)$.

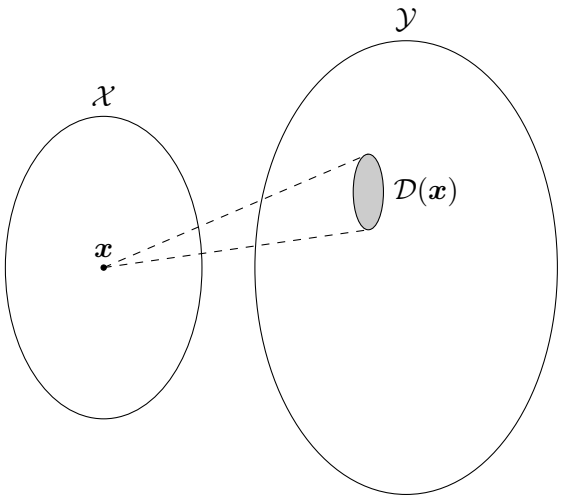


Figure 7.7: The decoding region of $\mathcal{Y}$ mapping to the codeword $\boldsymbol{x} \in \mathcal{X}$.

The more formal proof uses the same argumentation.

Proof: We start with the existence part of the proof, and will apply a random coding argument. With the capacity given by

$$C = \max_{p(x)} I(X;Y)$$

let $p^*(x)$ be the optimizing distribution of X,

$$p^*(x) = \arg\max_{p(x)} I(X;Y)$$

Consider a code with rate R. Then with a codeword length of n there are 2^{nR} codewords, and we have a $(2^{nR}, n)$ code. The codewords in the code is chosen randomly according to the distribution

$$p(\boldsymbol{x}) = \prod_i p^*(x)$$

and the decoding of a received word $\boldsymbol{y}$ is done by finding a codeword $\boldsymbol{x}$ such that $(\boldsymbol{x}, \boldsymbol{y}) \in A_\varepsilon^{(n)}(X, Y)$. If such codeword does not exists or is not unique, an error has occurred. Introduce the event that the pair $(\boldsymbol{x}, \boldsymbol{y})$ is a jointly typical sequence

$$E_{\boldsymbol{x}} = \Big\{(\boldsymbol{x}, \boldsymbol{y}) \in A_\varepsilon^{(n)}(X, Y)\Big\}$$

Then, for a codeword $\boldsymbol{x}_0$, an error occurs if this is not fulfilled, $(\boldsymbol{x}_0, \boldsymbol{y}) \in E_{\boldsymbol{x}_0}^c$ where ^{c} denote the complementary set, or if it is not unique, $(\boldsymbol{x}_0, \boldsymbol{y}) \in E_{\boldsymbol{x}_0}$ and $(\boldsymbol{x}, \boldsymbol{y}) \in E_{\boldsymbol{x}}$ for $\boldsymbol{x} \neq \boldsymbol{x}_0$ (i.e. two equally likely codewords). Then, an error event for codeword $\boldsymbol{x}_0$ can be denoted

$$E_e = E_{\boldsymbol{x}_0}^c \cup \Big(E_{\boldsymbol{x}_0} \cap \Big(\bigcup_{\boldsymbol{x} \neq \boldsymbol{x}_0} E_{\boldsymbol{x}}\Big)\Big)$$

The mathematical structure $(\mathcal{P}(\mathcal{U}), \mathcal{U}, \emptyset, \cap, \cup, ^c)$, where $\mathcal{P}(\mathcal{U})$ is the power set, i.e. the set of all subsets of $\mathcal{U}$, is a Boolean algebra. Here the complete set $\mathcal{U}$ is the element 1, the empty set $\emptyset$ is the 0, $\cap$ the AND operator, $\cup$ the OR operator and ^{c} the complement. Then we can find that[2]

$$E_e = E_{\boldsymbol{x}_0}^c \cup \Big(\bigcup_{\boldsymbol{x} \neq \boldsymbol{x}_0} E_{\boldsymbol{x}}\Big)$$

The probability of error is then the probability of the event E_e, and by using the union bound we can get

$$P_e = P(E_e) = P\Big(E_{\boldsymbol{x}_0}^c \cup \Big(\bigcup_{\boldsymbol{x} \neq \boldsymbol{x}_0} E_{\boldsymbol{x}}\Big)\Big) \leq P\big(E_{\boldsymbol{x}_0}^c\big) + \sum_{\boldsymbol{x} \neq \boldsymbol{x}_0} P\big(E_{\boldsymbol{x}}\big) \tag{7.1}$$

The two probabilities included here we can derive

$$\begin{aligned} P(E_{\boldsymbol{x}_0}^c) &= P\big((\boldsymbol{x}_0, \boldsymbol{y}) \notin A_\varepsilon^{(n)}(X, Y)\big) \to 0, \quad n \to \infty \\ P(E_{\boldsymbol{x}}) &= P\big((\boldsymbol{x}, \boldsymbol{y}) \in A_\varepsilon^{(n)}(X, Y)\big) \leq 2^{-n(I(X;Y)-3\varepsilon)} = 2^{-n(C-3\varepsilon)} \end{aligned}$$

where the last equality comes from that we assumed X is distributed according to the maximizing distribution $p^*(x)$. Then, as $n \to \infty$, (7.1) becomes

$$P_e = \sum_{\boldsymbol{x} \neq \boldsymbol{x}_0} P\big(E_{\boldsymbol{x}}\big) \leq \sum_{\boldsymbol{x} \neq \boldsymbol{x}_0} 2^{-n(C-3\varepsilon)} = (2^{nR} - 1)2^{-n(C-3\varepsilon)} < 2^{n(R-C+3\varepsilon)}$$

To get arbitrarily low error probability we require that $P_e \to 0$ as $n \to \infty$. That is, that

$$\Big(2^{R-C+3\varepsilon}\Big)^n \to 0$$

which is equivalent to $2^{R-C+3\varepsilon} < 1$, or $R - C + 3\varepsilon < 0$. This gives that

$$R < C - 3\varepsilon$$

[2]In a Boolean algebra $(\mathcal{B}, 1, 0, \wedge, \vee, ')$, where $\wedge$ is AND, $\vee$ is OR and $'$ complement, we can use the two rules consensus and absorption to get

$$a' \vee (a \wedge b) = a' \vee (a \wedge b) \vee b = a' \vee b$$

Using instead set operations and letting $a = E_{\boldsymbol{x}_0}$ and $b = \bigcup_{\boldsymbol{x} \neq \boldsymbol{x}_0} E_{\boldsymbol{x}}$ we get the result.

Since we have considered a random code, there exists at least one that fulfills this requirement, and we have shown that such code exists.[3]

We will now show the converse, that it is impossible to achieve reliable communication if coding rate exceeds the capacity. First assume that $R > C$ and that the 2^{nR} codewords are equally likely. The latter assumption means the information symbols are equally likely and hence, that we have perfect source coding. We will also assume that the codewords and received words are n dimensional vectors. If we let the decoding estimate the transmitted codeword $\boldsymbol{X}$ and assume that this uniquely maps to an estimated information symbol, the error probability is $P_e = P(\boldsymbol{X} \neq \widehat{\boldsymbol{X}})$, where $\boldsymbol{X}$ is the transmitted codeword and $\widehat{\boldsymbol{X}}$ the estimated. The, according to Fono's lemma we have

$$H(\boldsymbol{X}|\widehat{\boldsymbol{X}}) \leq h(P_e) + P_e \log\left(2^{nR} - 1\right)$$

On the other hand

$$I(\boldsymbol{X}; \widehat{\boldsymbol{X}}) = H(\boldsymbol{X}) - H(\boldsymbol{X}|\widehat{\boldsymbol{X}})$$

which leads to an expression for the left hand side of Fano's inequality

$$H(\boldsymbol{X}|\widehat{\boldsymbol{X}}) = H(\boldsymbol{X}) - I(\boldsymbol{X}; \widehat{\boldsymbol{X}}) \geq H(\boldsymbol{X}) - I(\boldsymbol{X}; \boldsymbol{Y}) \tag{7.2}$$

where we used the data processing lemma. The mutual information over the channel can be written as

$$\begin{aligned} I(\boldsymbol{X}; \boldsymbol{Y}) &= H(\boldsymbol{Y}) - H(\boldsymbol{Y}|\boldsymbol{X}) = H(\boldsymbol{Y}) - \sum_{1=1}^{n} H(Y_i|Y_1 \dots Y_{i-1}\boldsymbol{X}) \\ &= H(\boldsymbol{Y}) - \sum_{1=1}^{n} H(Y_i|X_i) \leq \sum_{1=1}^{n} H(Y_i) - \sum_{1=1}^{n} H(Y_i|X_i) \\ &= \sum_{1=1}^{n} H(Y_i) - H(Y_i|X_i) = \sum_{1=1}^{n} I(X_i; Y_i) \leq \sum_{1=1}^{n} C = nC \end{aligned}$$

where we used that the channel is memory-less and therefore Y_i conditioned on X_i does not depend on the other elements of the codeword or received word. We can rewrite this in (7.2),

$$H(\boldsymbol{X}|\widehat{\boldsymbol{X}}) \geq H(\boldsymbol{X}) - nC = nR - nC = n(R - C)$$

The first equality comes from the assumption that the codewords are equally likely with probability $P(\boldsymbol{X}) = 1/2^{nR}$, which gives the entropy $H(\boldsymbol{X}) = \log 2^{nR} = nR$. Then we can go back to Fano's inequality and write

$$h(P_e) + P_e \log\left(2^{nR} - 1\right) \geq n(R - C) > 0$$

where the strict inequality follows from the assumption that $R > C$. This means that the left hand side is strictly positive and that $P_e > 0$. Hence, the error probability will not go to zero as n goes to infinity, whatever code we choose. This concludes the proof of the

[3] this proof only shows it is possible to find such code. It does not say anything about how it should be done. Since the publication of this in [20] there has been a lot of research in this topic and error correcting codes, and today there are codes that can come very close to the capacity limit.

channel coding theorem. ■

An interesting extension of the channel coding theorem is to have a system with a dedicated feedback channel from the receiver to the transmitter, as shown in Figure 7.8. Then the transmitter can see the previous received symbol Y_{i-1}, so the transmitted symbol is $X_i = X(U, Y_{i-1})$.

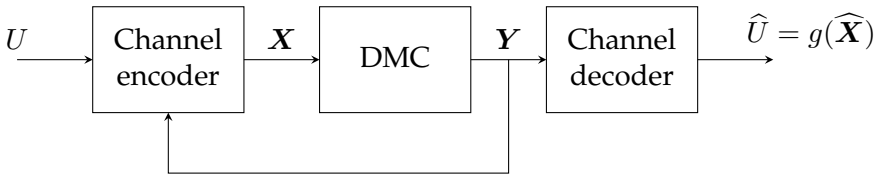


Figure 7.8: A channel with feedback.

The interesting thing is that for this channel the capacity is not improve compared to the case without feedback. In reality the feedback can have a significant meaning, making the decoding easier, but we still have the same bound. To see this we have to go through the proof once again. The first part, existence, does not depend on the feedback and can be reused entirely. But for the converse part, we cannot assume $I(\boldsymbol{X}; \boldsymbol{Y}) \leq nC$. Instead, consider the error probability $P_e = P(U \neq \widehat{U})$. Then Fano's lemma can be written as

$$H(U|\widehat{U}) \leq h(P_e) + P_e \log\bigl(2^{nR} - 1\bigr)$$

but we also have

$$H(U|\widehat{U}) = H(U) - I(U; \widehat{U}) \geq H(U) - I(U; \boldsymbol{Y})$$

where we again used the data processing lemma. The mutual information between the information symbol and the received vector can be written as

$$\begin{aligned} I(U; \boldsymbol{Y}) &= H(\boldsymbol{Y}) - H(\boldsymbol{Y}|U) \leq \sum_i H(Y_i) - \sum_i H(Y_i|Y_1 \dots Y_{i-1}W) \\ &= \sum_i H(Y_i) - \sum_i H(Y_i|Y_1 \dots Y_{i-1}WX_i) \\ &= \sum_i H(Y_i) - \sum_i H(Y_i|X_i) = \sum_i I(X_i; Y_i) \leq nC \end{aligned}$$

With this it becomes very similar to the case with no feedback,

$$H(U|\widehat{U}) \geq H(U) - I(U; \boldsymbol{Y}) \geq nR - nC = n(R - C)$$

which leads back to the same argument that the if $R > C$ it can not happen that the error probability goes to zero as n becomes large. Hence we have the following theorem.

Theorem 39 *The capacity for a feedback channel is equal to the non-feedback channel,*

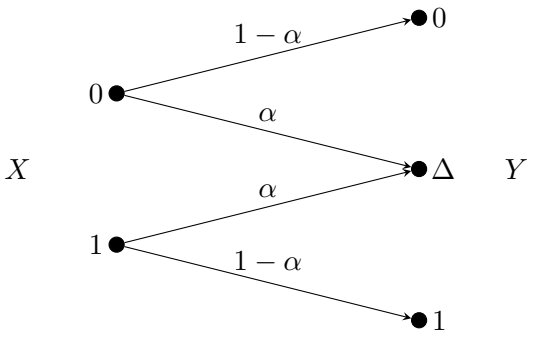
$$C_{FB} = C = \max_{p(x)} I(X; Y)$$

□

7.2 Channel Capacity

In Example 45 the capacity for the binary symmetric channel (BSC) was derived as $C = 1-h(p)$ where p is the bit error probability on the channel. We have now seen that this is a hard limit for what rates are possible to achieve reliable communication. For example, if the bit error probability on the channel is $p = 0.1$ the capacity is $C = 1 - h(0.1) = 0.9192$. Then we can achieve reliable communication if and only if $R < 0.9192$. In this section we will have a look at some other common channels and derive the channel capacity.

Example 46 [Binary Erasure Channel, BEC] The binary erasure channel is an extension of the BSC, in the sense that there is one more output symbol, Δ. If the received signal is such that it is very likely to be correct received it should be the corresponding binary symbol, while if it is more unsure it should be the symbol Δ. The graphical representation of the channel and its probability distribution is shown in Figure 7.9.



$p(y\|x)$		Y 0	Δ	1
X	0	$1-\alpha$	α	0
	1	0	α	$1-\alpha$

Figure 7.9: The binary erasure channel.

The mutual information between X and Y can be written as

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) = H(Y) - \sum_x H(Y|X=x)P(X=x) \\ &= H(Y) - h(\alpha) \end{aligned}$$

since $H(Y|X=x) = H(1-\alpha, \alpha, 0) = h(\alpha)$, both when $x = 0$ and $x = 1$. Instead of maximizing $I(X;Y)$ we can maximize $H(Y)$, for varying $p = P(X = 1)$. We know that $H(Y) \leq \log 3$, but this requires equiprobable Y, and contrary to the case for BSC, we do not have a way to guarantee this.

To derive $H(Y)$ we first need to get the distribution of Y, expressed in terms of the distribution of X. Assume that $P(X = 1) = p$, then with $p(x, y) = p(x)p(y|x)$ and $p(y) = \sum_x p(x, y)$ we can get the distribution of Y as

$p(x,y)$		Y 0	Δ	1
X	0	$(1-p)(1-\alpha)$	$(1-p)\alpha$	0
	1	0	$p\alpha$	$p(1-\alpha)$
$p(y)$		$(1-p)(1-\alpha)$	α	$p(1-\alpha)$

Hence the entropy of Y can be derived $H(Y) = H((1-p)(1-\alpha), \alpha, p(1-\alpha))$. Naturally, this function can be optimized by solving for p the derivative equal to zero. But, we can also go ahead and see

$$\begin{aligned} H(Y) &= -(1-p)(1-\alpha)\log(1-p)(1-\alpha) - \alpha\log\alpha - p(1-\alpha)\log p(1-\alpha) \\ &= (1-\alpha)\Big(-p\log p - (1-p)\log(1-p)\Big) - \alpha\log\alpha - (1-\alpha)\log(1-\alpha) \\ &= (1-\alpha)h(p) + h(\alpha) \end{aligned}$$

Hence the capacity is

$$C = \max_p H(Y) - h(\alpha) = \max_p (1-\alpha)h(p) = 1-\alpha$$

In Figure 7.10 the capacity for the binary erasure channel is plotted as a function of the erasure probability α. If the $\alpha = 0$ we get the capacity equal to one, since then we do not have any errors. On the other hand, if $\alpha = 1$ all of our received symbols will be erasures and we do not have any information about the transmitted sequence.

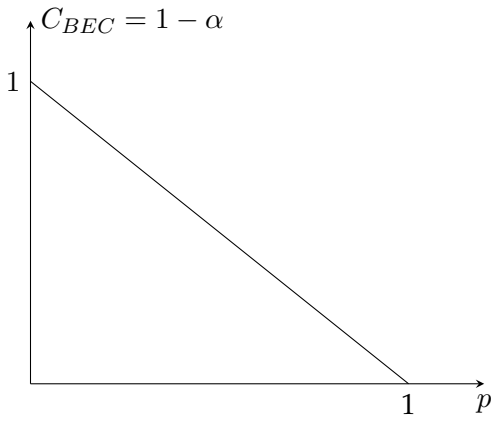


Figure 7.10: The capacity of the binary erasure channel as a function of the erasure probability.

In this example we could derive the capacity by considering the entropy of the received variable as a function of the distribution of the transmitted variable. This was possible because of the symmetry in the channel, but in many cases we have to use standard optimization techniques, like taking the derivative equal to zero and solving for the distribution of X. But there are also many cases where we can make use of the structure and the symmetry of the channel. To see this we first make a definition of what we mean by a *symmetric channel*[4].

Definition 24 (Symmetric channels) *A discrete memory-less channel is* ***symmetric*** *if*

- *for all X, the branches leaving the code symbol have the same set of probabilities, $p_1, p_2, \ldots, p_{|\mathcal{Y}|}$.*

[4] In the literature there are different names for different types of symmetries for the channel. We will in this text follow the notation in [2]

- *for all Y, the branches entering the received symbol have the same set of probabilities, $p_1, p_2, \ldots, p_{|\mathcal{X}|}$.*

Seen from the transition matrix all rows are permutations of each other and all columns are permutations of each other. □

The binary symmetric channel is an example of a symmetric channel. In the next example we show another popular channel, that is also symmetric.

Example 47 Consider a channel with four different inputs $\mathcal{X} = \{0, 1, 2, 3\}$, and the same set of outputs $\mathcal{Y} = \{0, 1, 2, 3\}$. The transmitted value is received correctly with probability p and as the next following symbol (modulo 4) with probability p. In Figure 7.11 the channel is shown.

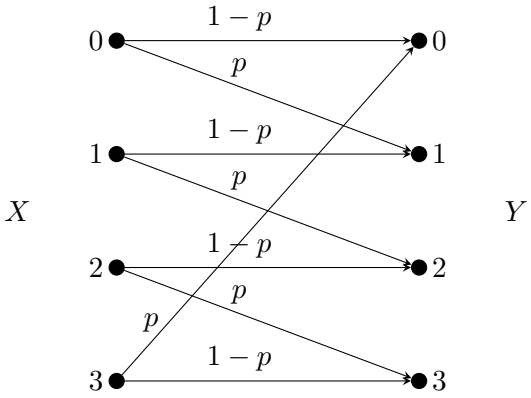


Figure 7.11: A symmetric channel.

The transition probability matrix is

$$P(Y|X) = \begin{pmatrix} 1-p & p & 0 & 0 \\ 0 & 1-p & p & 0 \\ 0 & 0 & 1-p & p \\ p & 0 & 0 & 1-p \end{pmatrix}$$

The mutual information between X and Y is given by

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) = H(Y) - \sum_x \underbrace{H(Y|X=x)}_{h(p)} p(x) \\ &= H(Y) - h(p) \leq \log 4 - h(p) = 2 - h(p) \end{aligned}$$

where we have equality if and only if $p(y) = \frac{1}{4}$. Assume that $p(x) = \frac{1}{4}$ for all x, then

$$p(y) = \sum_x p(y|x)p(x) = \sum_x \frac{1}{4} p(y|x) = \frac{1}{4}p + \frac{1}{4}(1-p) = \frac{1}{4}$$

hence, the distribution of X maximizing $I(X;Y)$ is the uniform distribution, $p(x) = \frac{1}{4}$, and the capacity for the channel is

$$C = 2 - h(p)$$

The derivation of the capacity in the previous example is typical for symmetric channels. If we assume a general symmetric channel with an input alphabet $\mathcal{X}$ of size $|\mathcal{X}|$ and an output alphabet $\mathcal{Y}$ of size $|\mathcal{Y}|$. Then, following the previous example, we write the mutual information as

$$I(X;Y) = H(Y) - H(Y|X) = H(Y) - \sum_x H(Y|X=x)p(x)$$

Since the channel is symmetric the outgoing transitions from a given x is the same independent of the x. The entropy function does not take the order of the probabilities, i.e. the semantics of the message, into consideration. Therefore, the entropy of Y conditioned on x is the same for all x,

$$H(Y|X=x) = H(p_1, p_2, \ldots, p_{|\mathcal{X}|}) = H(\boldsymbol{r})$$

where $\boldsymbol{r} = (p_1, p_2, \ldots, p_{|\mathcal{X}|})$ is one row in the transition matrix $P(Y|X)$. We can now write the information as

$$I(X;Y) = H(Y) - H(\boldsymbol{r}) \leq \log|\mathcal{Y}| - H(\boldsymbol{r})$$

with equality if and only if $p(y) = \frac{1}{|\mathcal{Y}|}$. As in Example 47 let us assume that the distribution of X is uniform, i.e. $p(x) = \frac{1}{|\mathcal{X}|}$. Then the probability of Y becomes

$$p(y) = \sum_x p(x)p(y|x) = \frac{1}{|\mathcal{X}|}\sum_x p(y|x) = \frac{1}{|\mathcal{X}|}A$$

where the constant value $A = \sum_x p(y|x)$ follows from the symmetry. Summing this over Y we get

$$\sum_y p(y) = \sum_y A\frac{1}{|\mathcal{X}|} = A\frac{\frac{1}{|\mathcal{Y}|}}{\frac{1}{|\mathcal{X}|}} = 1$$

which gives that $p(y) = \frac{1}{|\mathcal{Y}|}$, and that

$$C_{\text{Sym}} = \max_{p(x)} I(X;Y) = \log|\mathcal{Y}| - H(\boldsymbol{r})$$

With this at hand we can go back to Example 47. Since the channel is symmetric and $|\mathcal{Y}| = 4$, we get

$$C = \log 4 - H(1-p, p, 0, 0) = 2 - h(p)$$

Also the binary symmetric channel (BSC) is symmetric. There we have $|\mathcal{Y}| = 2$ and $\boldsymbol{r} = (1-p, p)$. Hence, the capacity is

$$C_{BSC} = \log 2 - H(1-p, p) = 1 - h(p)$$

Actually, when deriving C_{Sym} we did not use that the incomming transitions to a y had the same distribution, we only used that the sum was constant for all y. Therefore, we can give a weaker definition of symmetry as below, and still have the same result on the capacity.

Definition 25 (Weakly symmetric channels) *A discrete memory-less channel is **weakly symmetric** if*

- *for all X, the branches leaving the code symbol have the same set of probabilities, $p_1, p_2, \ldots, p_{|\mathcal{Y}|}$.*
- *for all Y, the branches entering the received symbol have the same sum, $\sum_{i=1}^{|\mathcal{X}|} p_i$.*

Seen from the transition matrix $[p(y|x)]$ all rows are permutations of each other and all columns have the same sum $\sum_x p(y|x)$. □

We now state the result on the capacity as a theorem.

Theorem 40 *If a channel is symmetric or weakly symmetric, the channel capacity is*

$$C = \log |\mathcal{Y}| - H(\boldsymbol{r})$$

where $\boldsymbol{r}$ is the set of probabilities labeling branches leaving a code symbol X, or, viewed in a transition matrix, one row of the transition matrix. □

In the next example we show a channel hat is weakly symmetric.

Example 48 In Figure 7.12 a channel with two erasure symbols, one closer to 0 and one closer to 1, is shown.

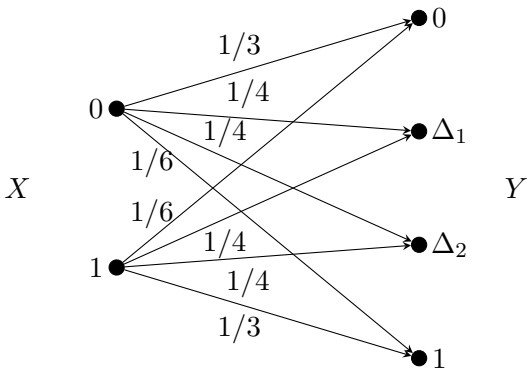


Figure 7.12: A binary double erasure channel that is weakly symmetric.

The corresponding transition probability matrix is

$$P(Y|X) = \begin{pmatrix} 1/3 & 1/4 & 1/4 & 1/6 \\ 1/6 & 1/4 & 1/4 & 1/3 \end{pmatrix}$$

We see that two rows have the same set of probabilities. Summing each column we get the constant value $1/2$, so we can conclude that the channel is weakly symmetric. The set of outputs $\mathcal{Y}$ has the cardinality $|\mathcal{Y}| = 4$, and we can calculate the capacity for this channel as

$$C = \log 4 - H\left(\tfrac{1}{3}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{6}\right) = 2 - 1.9591 = 0.0409$$

which is a very poor channel.

In all of the channels above we have used that $H(Y|X = x)$ is constant. This is true when the distribution for the outgoing branches in the channel model has the same set of probabilities. This property is called *uniformly dispersive*. All the channels above are uniformly dispersive and has

$$H(Y|X) = \sum_i p_i \log p_i$$

where $p_1, \ldots, p_{|\mathcal{Y}|}$ is the set of outgoing probabilities stemming from each x, i.e. the probabilities of one row in the transition matrix $P(Y|X)$.

We will finish the section on capacity calculations by considering a channel that is neither symmetric nor uniformly dispersive. In this case we are often left with standard optimization methods. If the channel is even more complicated it might be better to solve it with numerical methods. In that case one should keep in mind that the mutual information is a convex function, which simplify things.

Example 49 Consider the three input and three output channel described in Figure 7.13. To derive the capacity for this channel we need to assign a distribution for X over which we can optimize. In a general approach we can set probability variables for two of the values for X. However, we can use the symmetry in the channel together with the fact that the mutual information is convex in the probabilities. Therefore, we conclude that the probability for 0 and 2 should be equal in the optimization. Hence, we set $P(X = 0) = P(X = 2) = p$. Then $P(X = 1) = 1 - 2p$.

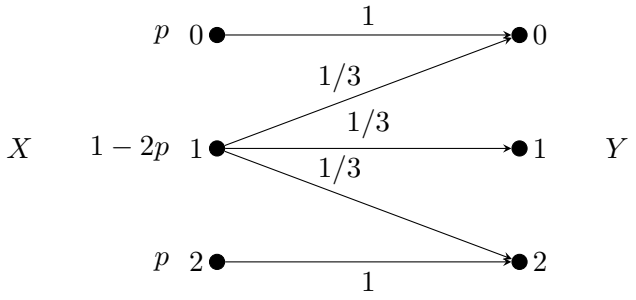


Figure 7.13: A ternary channel.

For this channel we can derive the mutual information as

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H(Y) - ph(0) - (1-2p)H\left(\tfrac{1}{3}, \tfrac{1}{3}, \tfrac{1}{3}\right) - ph(0) \\ &= H(Y) - (1-2p)\log 3 \end{aligned}$$

The distribution of Y can be found by $P(Y = 0) = P(Y = 2) = p + \frac{1}{3}(1-2p) = \frac{1+p}{3}$ and $P(Y = 1) = \frac{1-2p}{3}$. Now we can express the mutual information as

$$\begin{aligned} I(X;Y) &= H\Big(\frac{1+p}{3}, \frac{1+p}{3}, \frac{1-2p}{3}\Big) - (1-2p)\log 3 \\ &= -2\frac{1+p}{3}\log\frac{1+p}{3} - \frac{1-2p}{3}\log\frac{1-2p}{3} - (1-2p)\log 3 \end{aligned}$$

To maximize, set the derivative equal to zero,

$$\begin{aligned}\frac{\partial}{\partial p} I(X;Y) &= -\frac{2}{3\ln 2} - \frac{2}{3}\log\frac{1+p}{3} + \frac{2}{3\ln 2} + \frac{2}{3}\log\frac{1-2p}{3} + 2\log 3 \\ &= \frac{2}{3}\log\frac{1-2p}{1+p} + 2\log 3 = 0\end{aligned}$$

where we used that $\frac{\partial}{\partial p}\big(f(p)\log f(p)\big) = \frac{\partial f(p)}{\partial p}\big(\frac{1}{\ln 2} + \log f(p)\big)$. The above equation is equivalent to

$$\frac{1-2p}{1+p} = \frac{1}{27}$$

which leads to

$$p = \frac{26}{55}$$

So the optimizing distribution of X is $P(X=0) = P(X=2) = \frac{26}{55}$ and $P(X=1) = \frac{3}{55}$. To get the capacity we use this in $I(X;Y)$,

$$C = I(X;Y)\Big|_{p=\frac{26}{55}} = H\Big(\frac{27}{55}, \frac{27}{55}, \frac{1}{55}\Big) - \frac{3}{55}\log 3 \approx 1.0265$$

Example 50 [Binary asymetric channel]

To be done.

Chapter 8

Channel Coding

In the previous chapter the capacity for a channel was introduced as the limit for how much information can be transmitted. The question now is how to reach the limit. By just changing the modulation scheme itself is not enough, even if that is also an important part of the solution although not covered in this text. In this chapter another important part of the puzzle will be introduced, namely channel coding.

After the source encoding in the system the redundancy of the data should be, if not removed, so at least significantly reduced. That means that U in the figure is more or less a representation of the pure information from the source. It can therefore be assumed that the symbols in U are very close to equally likely. On the other hand, during the transmission over the channel there will be disturbances that will either alter or erase some of the transmitted symbols. By adding redundancy in a controlled way it is possible to correct some of these errors. In the figure the encoded symbol is denoted X. This is then transmitted over the channel and the received symbol is denoted Y. The purpose of the channel decoder is to find and, if possible, correct the errors from the channel.

8.1 Channel coding

Knowing that it is possible to construct encoding and decoding rules that are good, the question of how remains. We will now give two examples of codes that we will look a bit more into in the continuation of this section. The first one is a very simple and intuitive code, called the repetition code.

Example 51 [Repetition code]
Instead of transmitting each bit uncoded we transmit three copies of each bit. That is, the encoding rule becomes

U	X
0	000
1	111

then the codeword is transmitted over the channel, where it is corrupted by noise. If the channel is assumed to be a binary symmetric channel, there might be some of the bits

altered. Assuming that the crossover probability is relatively small, the most probable error event is that one bit is altered. Then receiving e.g. 101, we would assume that it is the middle bit that is altered and the estimate of the transmitted codeword 111. therefore, we decode the received word 101 to the transmitted $\widehat{U} = 1$. Of course, it might be that the transmitted bit was 0 and that it is the first and third bits that has been altered, but the probability for this is very small.

This code has $M = 2$ codewords and they are vectors of length $n = 3$, so the repetition code is a $(2, 3)$ code. The coding rate is $R = \frac{\log 2}{3} = \frac{1}{3}$, meaning there are in average $1/3$ bit information in each code symbol.

The repetition code is the simplest example that shows it is possible to correct errors that has occurred on the channel. However, it is not a very efficient way to do this, and the remaining error probability after decoding of the sequence is not really improved. There are much more efficient ways to so this, and in this chapter we will introduce some of the ideas. Still, to get codes that can be realised in real communication systems we recommend a good book in coding theory, e.g. [15, 16, 11, 9].

Normally, a code is defined as the set of codewords, and not dependent on the mapping of the information words. In the next example we introduce a code with four codewords.

Example 52 The code $\mathcal{B}_1 = \{0000, 1011, 0110, 1101\}$ has four codewords, meaning $M = 4$. There are four bits in each codewords, so $n = 4$. Hence, we have an $(M, n) = (4, 4)$ code. The code rate is $R = \frac{2}{4}$.

Often we want to have a linear encoding rule, which put constraints on the codewords. Before we continue to the mapping for the encoder we need to consider linear codes.

Definition 26 *A code $\mathcal{B}$ is **linear** if for every pair of codewords $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ their sum is also a codeword,*

$$\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{B} \Rightarrow \boldsymbol{x}_i \oplus \boldsymbol{x}_j \in \mathcal{B}$$

where $\oplus$ denote the modulo 2 addition. □

It can directly be seen that the repetition code is linear. Also the code $\mathcal{B}_1$ in the previous example can easily be verified, by viewing the addition between all pairs of codewords, to be linear. Since a codeword plus itself is the all-zero vector,

$$\text{if } \boldsymbol{x}_i = \boldsymbol{x}_j \quad \Rightarrow \quad \boldsymbol{x}_i \oplus \boldsymbol{x}_j = \mathbf{0}$$

since over the binary field addition and subtraction are identical. We can conclude that this has to be a codeword in any linear code, $\mathbf{0} \in \mathcal{B}$.

From algebra it is known that a (binary) linear code $\mathcal{B}$, with rate $R = \frac{k}{n}$, is a k-dimensional subspace of the binary n-dimensional space $\mathbb{F}_2^n$. Then each codeword is a linear combination of k linearly independent codewords, $\boldsymbol{g}_1, \ldots, \boldsymbol{g}_k$, where $\boldsymbol{g}_i \in \mathcal{B}$. Since the k codewords are linearly independent all different linear combinations of them will give

different codewords as results. Use the binary vector $\boldsymbol{u} = (u_1, \ldots, u_k)$ as coefficients for the linear combinations. Then

$$\boldsymbol{x} = u_1\boldsymbol{g}_1 \oplus \cdots \oplus u_k\boldsymbol{g}_k = (u_1 \ldots u_k) \begin{pmatrix} \boldsymbol{g}_1 \\ \vdots \\ \boldsymbol{g}_k \end{pmatrix} = \underbrace{(u_1 \ldots u_k)}_{\boldsymbol{u}} \underbrace{\begin{pmatrix} g_{11} & \cdots & g_{1n} \\ \vdots & & \vdots \\ g_{k1} & \cdots & g_{kn} \end{pmatrix}}_{G} \tag{8.1}$$

will describe 2^k codewords, each with length n. The equation above can be used as the encoding rule for an $(M, n) = (2^k, n)$ linear code, where $\boldsymbol{u}$ is the information word and $\boldsymbol{x}$ the codeword. The matrix G is named the *generator matrix* for the code and determines the mapping. Of course, by choosing another order or another set of the codewords in the generator matrix the mapping will be altered, but the set of codewords, i.e. the code, will be the same.

Example 53 In the code $\mathcal{B}_1$ the two codewords $\boldsymbol{x}_1 = 0110$ and $\boldsymbol{x}_2 = 1101$ are linearly independent. Therefore we can form the generator matrix

$$G = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

The mapping between the information words and the codewords then becomes

$\boldsymbol{u}$	$\boldsymbol{x} = \boldsymbol{u}G$
00	0000
01	1101
10	0110
11	1011

which is, of course, the same code as in the previous example.

The next problem is to find out if this is a good code or not. For that we first assume that errors that occur on the channel are independent. This model can be discussed, but in for example satellite communication where the most pronounced disturbance is the background noise it is a fairly good model. There are a lot of other models for disturbances but independent errors is the most used in this topic. With this assumption the case we had earlier with one error in a codeword is the most probable error. After that comes two errors, and so on.

Then need to consider what decoding rule should be used for this type of disturbances. One direct first decoding rule is to chose the codeword that is the most likely to be sent, conditioned on the received word. This decoding rule is called the **maximum a posterirori** (MAP) decoder,

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x} \in \mathcal{B}} P(\boldsymbol{X} = \boldsymbol{x} | \boldsymbol{y})$$

Earlier we viewed the case when the source encoding preceded the channel coding. Then it is reasonable to assume that all codewords, or information words, are equally likely,

$P(\boldsymbol{x}) = 2^{-k}$. We can then expand the MAP rule according to

$$\begin{aligned}\hat{\boldsymbol{x}} &= \arg\max_{\boldsymbol{x}\in\mathcal{B}} P(\boldsymbol{x}|\boldsymbol{y}) = \arg\max_{\boldsymbol{x}\in\mathcal{B}} P(\boldsymbol{x},\boldsymbol{y})\frac{1}{P(\boldsymbol{y})} \\ &= \arg\max_{\boldsymbol{x}\in\mathcal{B}} P(\boldsymbol{y}|\boldsymbol{x})\frac{P(\boldsymbol{x})}{P(\boldsymbol{y})} = \arg\max_{\boldsymbol{x}\in\mathcal{B}} P(\boldsymbol{y}|\boldsymbol{x})\end{aligned}$$

since $\frac{P(\boldsymbol{x})}{P(\boldsymbol{y})}$ can be considered as a constant. This decoding rule is called *maximum likelihood* (ML), and is formulated as

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x}\in\mathcal{B}} P(\boldsymbol{y}|\boldsymbol{X} = \boldsymbol{x})$$

To say something about what errors should be possible to correct we need a way to measure how many errors there are between two codewords, at minimal. Then the Hamming distance and Hamming weight are very handy.

Definition 27 *The **Hamming distance** between two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$, $d_H(\boldsymbol{x},\boldsymbol{y})$, is the number of positions in which they differ.*

*The **Hamming weight** of a vector $\boldsymbol{x}$, $w_H(\boldsymbol{x})$, is the number of non-zero positions.* □

For binary vectors we can derive the Hamming distance from the Hamming weight.

$$d_H(\boldsymbol{x},\boldsymbol{y}) = w_H(\boldsymbol{x}-\boldsymbol{y}) = w_H(\boldsymbol{x}\oplus\boldsymbol{y})$$

Example 54 The Hamming distance between the vectors 0011010 and 0111001 is the number of positions they differ

$$d_H(0\underline{0}110\underline{10}, 0\underline{1}110\underline{01}) = 3$$

It can also be derived as the weight of the difference,

$$w_H(0011010 \oplus 0111001) = w_H 0100011 = 3$$

If we again consider the ML decoding, but instead of maximizing the probability we maximize the logarithm of it,

$$\begin{aligned}\hat{\boldsymbol{x}} &= \arg\max_{\boldsymbol{x}\in\mathcal{B}} P(\boldsymbol{y}|\boldsymbol{x}) = \arg\max_{\boldsymbol{x}\in\mathcal{B}} \log P(\boldsymbol{y}|\boldsymbol{x}) \\ &= \arg\max_{\boldsymbol{x}\in\mathcal{B}} \log\prod_i P(y_i|x_i) = \arg\max_{\boldsymbol{x}\in\mathcal{B}} \sum_i \log P(y_i|x_i)\end{aligned}$$

where we assumed that the errors on the channel occur independent of each other. If an error occurs with probability p and there are $d_H(\boldsymbol{x},\boldsymbol{y})$ errors in the transmission, and $n - d_H(\boldsymbol{x},\boldsymbol{y})$ correct positions, we get

$$\begin{aligned}\hat{\boldsymbol{x}} &= \arg\max_{\boldsymbol{x}\in\mathcal{B}} d_H(\boldsymbol{x},\boldsymbol{y})\log p + (n - d_H(\boldsymbol{x},\boldsymbol{y}))\log(1-p) \\ &= \arg\max_{\boldsymbol{x}\in\mathcal{B}} d_H(\boldsymbol{x},\boldsymbol{y})\log\frac{p}{1-p} + n\log 1 - p = \arg\max_{\boldsymbol{x}\in\mathcal{B}} d_H(\boldsymbol{x},\boldsymbol{y})\end{aligned}$$

since $\log \frac{p}{1-p} < 0$ for small error probabilities, $p < \frac{1}{2}$, and $n \log 1 - p$ constant for all codewords. This decoding rule is called *minimum distance* (MD) decoding and is an easy to use alternative for a BSC.

The minimum distance decoding rule shows that the Hamming distance between codewords should be made large, to allow for errors to be corrected. Therefore the minimum distance between two codewords is an important measure of how good a code is.

Definition 28 *The **minimum distance** for a code is the minimum Hamming distance between different codewords,*

$$d_{\mathsf{min}} = \min_{\substack{\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{B} \\ \boldsymbol{x}_1 \neq \boldsymbol{x}_2}} d_H(\boldsymbol{x}_1, \boldsymbol{x}_2)$$

□

For a linear code we can utilize that the sum of two codewords is a codeword. Since, in the binary case, the distance between two vectors is the weight of their sum we can derive the minimum distance as

$$d_{\mathsf{min}} = \min_{\substack{\boldsymbol{x} \in \mathcal{B} \\ \boldsymbol{x} \neq \mathbf{0}}} w_H(\boldsymbol{x})$$

Example 55 The minimum distance for the repetition code is

$$d_{\mathsf{min}}^{(\mathrm{Rep})} = \min\{w_H(111)\} = 3$$

and for $\mathcal{B}_1$

$$d_{\mathsf{min}}^{(1)} = \min\{w_H(1101), w_H(0110), w_H(1011)\} = \min\{3, 2, 3\} = 2$$

Since the codewords are n dimensional vectors they can be viewed in an n dimensional binary space. In this space the codewords are represented by 2^k positions, while the possible received positions are all possible in the space, i.e. 2^n. In Figure 8.1 the two codewords $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in the n dimensional space are shown (in two dimensions). The Hamming distance between them is at least d_{min}. As a decoding rule we can now use the minimum distance decoder. Then we surround each codeword with a sphere of radius t bits, such that there are no points in the sphere that is in more than one spere. Then all received sequences within such spere should be decoded to the codeword in its centre. For exampe, the received vector $\boldsymbol{y}$ in the figure is closest to the codeword $\boldsymbol{x}_1$ so that should be our estimate of the transmitted codeword, $\hat{\boldsymbol{x}} = \boldsymbol{x}_1$.

By making the the sphere radius t as large as possible, we can correct as many errors as possible. Since it is a discrete space and we should have no overlapping of the speres t must fullfil

$$2t + 1 \leq d_{\mathsf{min}}$$

or, in other words

$$t \leq \frac{d_{\mathsf{min}} - 1}{2}$$

Since the distance in the figure is the Hamming distance, t is the maximum number of errors allowed for the received sequence to stay in the sphere. Hence, this is the maximum number of errors that can allways be corrected by the decoder.

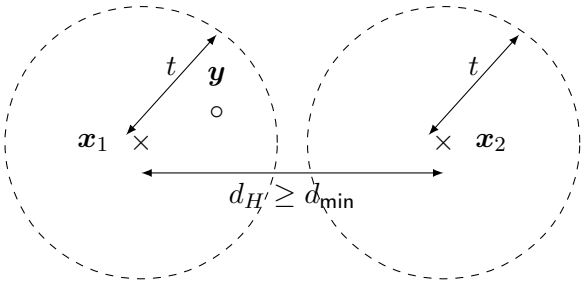


Figure 8.1: Two codewords, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, in the n dimensional binary space, projected into two dimensions.

If instead our communication system is working with e.g. retransmissions, the question for the decoder would be to detect if there has occured errors on the chennel. If the transmission was error-free, the received sequence is a codeword. So, receiving a codeword we assume that this was transmitted and that it is error-free. On the other hand, if a vector not corresponding to a codeword is received we can directly say the transmission was not error-free, and we have detected an error. An errort that is not detected must therefore be such that it gives a new codeword. The least weight of such a codeword must be d_{min}. Hence, all errors with weight $d_{\mathsf{min}} - 1$ can be detected. We summarize the above in the following theorem.

Theorem 41 *When using a code with minimum distance d_{min}, it is allways possible to either **detect** an error $\boldsymbol{e}$ if*

$$w_H(\boldsymbol{e}) < d_{\mathsf{min}}$$

*or **correct** an error $\boldsymbol{e}$ if*

$$w_H(\boldsymbol{e}) \leq \frac{d_{\mathsf{min}} - 1}{2}$$

□

Example 56 The repetition code has $d_{\mathsf{min}} = 3$, and can therefore either detect two errors or correct one error. The code $\mathcal{B}_1$ has $d_{\mathsf{min}} = 2$ and can detect all errors of weight one, but there is no guarantee of correcting errors. For example if the received vector is $\boldsymbol{y} = 1001$, this differs one bit from both the codewords 1101 and 1011. Since there are two codewords that are closest we have a 50-50 chance to choose the correct one.

There are two main classes of error correcting codes, block codes and covolutional codes. In the continuation of this chapter we will first describe the basic principles for block codes by considering the most simple case, Hammingcodes. Then follows a short introduction to convolutional codes.

8.2 Hamming codes

The codes considered in the previous part of this chapter are block codes. The are characterised from that an information word consisting of a block, or vector, of finite length is encoded into a codeword, also a vector of finite length. The vectors are typically considered over a finite field, which in this text will be the binary field. For a linear code we can find a generator matrix G such that the mapping between information word $\boldsymbol{u}$ and the codeword $\boldsymbol{x}$ as $\boldsymbol{x} = \boldsymbol{u}G$. The generator matrix has dimensions $k \times n$ and spans a k-dimensional sub-space of the binary n-dimensional space $\mathbb{F}_2^n$. Then the null-space of it is spanned by the *parity check matrix* H, defined by

$$GH^T = 0$$

where ^{T} denote the matrix transpose. Since a length n vector $\boldsymbol{x}$ is a codeword if and only if $\boldsymbol{u}G$ for some length k vector $\boldsymbol{u}$. Then,

$$\boldsymbol{x}H^T = \boldsymbol{u}GH^T = \boldsymbol{u}0 = 0$$

which shows the following theorem

Theorem 42 *Let H be a parity check matrix for a $(2^k, n)$ linear block code. Then an n-dimensional vector $\boldsymbol{x}$ is a codeword if and only if*

$$\boldsymbol{x}H^T = 0 \tag{8.2}$$

□

Since the minimum distance is the minimum weight of a non-zero codeword we can use the parity check matrix to find it. The requirements in (8.2) means that using the coefficients in $\boldsymbol{x} = (x_1 \dots x_n)$ the linear combination of the columns in H equals zero. For a codeword of least weight, this corresponds to a linear combination of d_{min} columns of H giving a zero. It also means there are no linear combination of less number of columns summing to zero. We state this as a theorem.

Theorem 43 *The minimum distance d_{min} of a linear code equals the minimum number of linearly dependent columns in the parity check matrix H.* □

With this we can find a way to design a code through the parity check matrix. By choosing all non-zero vectors of length m as columns we get the parity check matrix H. This defines the *Hamming code*. The length of the codewords is the number of columns in H, i.e $n = 2^m - 1$, and the length of the information vector $k = 2^m - 1 - m$. Hence we have defined a $(2^{2^m-1-m}, 2^m - 1)$ binary linear code with rate

$$R = \frac{2^m - 1 - m}{2^m - 1}$$

Since all columns are different we cannot find two that adds to zero. On the other hand we can always find three that does. That is, the minimum number of linearly dependent columns is 3, and we conclude that the minimum distance of a Hamming code is $d_{\mathsf{min}} = 3$.

Example 57 Using $m = 3$ we get a $(16, 7)$ Hamming code, with rate $R = \frac{4}{7}$. The parity check matrix is formed by all non-zero (column-)vectors of length $m = 3$ in some order,

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

The codewords can be found from extensive search as

$\boldsymbol{x}$	$w_H(\boldsymbol{x})$	$\boldsymbol{x}$	$w_H(\boldsymbol{x})$	$\boldsymbol{x}$	$w_H(\boldsymbol{x})$	$\boldsymbol{x}$	$w_H(\boldsymbol{x})$
0000000	0	0100101	3	1000111	4	1100010	3
0001110	3	0101011	4	1001001	3	1101100	4
0010011	3	0110110	4	1010100	3	1110001	4
0011101	4	0111000	3	1011010	4	1111111	7

from the table of codewords we verify that $d_{\mathsf{min}} = \min_{\boldsymbol{x} \notin \mathcal{B}} w_H(\boldsymbol{x}) = 3$. This code can correct all single errors that occur on the channel.

So far we have only constructed the code, and not taken the mapping from information words to codewords into account. This should still be decided by the generator matrix. Of course it can be formed by choosing k linearly independent codewords but we can also utilize that we have the parity check matrix and that $GH^T = 0$. In H we have m columns with weight one, and together they form an identity matrix. In the example above we have chosen to set this sub-matrix to the left in the matrix, so we can write

$$H = \begin{pmatrix} I & P^T \end{pmatrix}$$

where I is the $m \times m$ unit matrix and P^T an $m \times 2^m - 1 - m$ binary matrix. Then, we can form a generator matrix as

$$G = \begin{pmatrix} P & I \end{pmatrix}$$

where P is the transpose of P^T and I the unit matrix of size $2^m - 1 - m$. Then, since over the binary field addition and subtraction are equivalent,

$$GH^T = \begin{pmatrix} P & I \end{pmatrix} \begin{pmatrix} I \\ P \end{pmatrix} = P \oplus P = 0$$

and we can conclude that G is a valid generator matrix.

Example 58 In the parity check matrix for the $(16, 7)$ Hamming code we identify

$$P^T = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}$$

and we can get the generator matrix

$$G = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

With this we have a more convenient way to list the codewords, by using $\boldsymbol{x} = \boldsymbol{u}G$,

$\boldsymbol{u}$	$\boldsymbol{x}$	$\boldsymbol{u}$	$\boldsymbol{x}$	$\boldsymbol{u}$	$\boldsymbol{x}$	$\boldsymbol{u}$	$\boldsymbol{x}$
0000	0000000	0100	1010100	1000	0111000	1100	1101100
0001	1110001	0101	0100101	1001	1001001	1101	0011101
0010	1100010	0110	0110110	1010	1011010	1110	0001110
0011	0010011	0111	1000111	1011	0101011	1111	1111111

Which should be the same set of codewords as in the previous example, but in another order. Notice that since the generator matrix has the identity matrix as one part we can directly find the information word in the codeword as the four last digits.

Next on the agenda is to find an efficient decoding algorithm for the code. If we assume a BSC we know that the minimum distance decoder is optimal. The errors in an BSC can be viewed as an additive error, where the error vector has a 1 in the positions that are changed and a 0 in the others. Then the received vector is

$$\boldsymbol{y} = \boldsymbol{x} \oplus \boldsymbol{e}$$

For example, if the codeword $\boldsymbol{x} = (0110110)$ is transmitted and there are errors in the 4th and 6th positions, the error vector is $\boldsymbol{e} = (0001010)$. The received vector is $\boldsymbol{y} = (0110110) \oplus (0001010) = (0111100)$. By using the parity check matrix again we can get

$$\boldsymbol{y}H^T = (\boldsymbol{x} \oplus \boldsymbol{e})H^T = \boldsymbol{x}H^T \oplus \boldsymbol{e}H^T = \boldsymbol{e}H^T$$

since $\boldsymbol{x}$ is a codeword. We define this as the syndrome for the received vector.

Definition 29 (Syndrom) *Let $\boldsymbol{x}$ be a codeword and H the corresponding parity check matrix. Then if the received vector is $\boldsymbol{y} = \boldsymbol{x} \oplus \boldsymbol{e}$, we can form the syndrome*

$$\boldsymbol{s} = \boldsymbol{y}H^T = \boldsymbol{e}H^T.$$

□

We can now make a table that maps the syndrome to the least weight error vectors and subtract it from the received vector. Then we can achieve an estimate of the error vector $\hat{\boldsymbol{e}}$. If $\hat{\boldsymbol{e}} = \boldsymbol{e}$ we get

$$\hat{\boldsymbol{x}} = \boldsymbol{y} \oplus \hat{\boldsymbol{e}} = \boldsymbol{x} \oplus \boldsymbol{e} \oplus \boldsymbol{e} = \boldsymbol{x}$$

Example 59 For the $(16, 7)$ Hamming code the syndrome for the least weight error patterns is

$\boldsymbol{e}$	$\boldsymbol{s} = \boldsymbol{e}H^T$	$\boldsymbol{e}$	$\boldsymbol{s} = \boldsymbol{e}H^T$
0000000	000	0001000	011
1000000	100	0000100	101
0100000	010	0000010	110
0010000	001	0000001	111

This table lists all possible syndromes, and we can use this as our decoding procedure.

Assume that the codeword $\boldsymbol{x} = (0110110)$ is transmitted and that there is an error in the 3rd bit, i.e. $\boldsymbol{e} = (0010000)$. Then the received vector is $\boldsymbol{y} = (0100110)$. The syndrome for this vector is $\boldsymbol{s} = \boldsymbol{y}H^T = (001)$ which translates to the estimated error vector $\hat{\boldsymbol{e}} = (001000)$ and the estimated codeword $\hat{\boldsymbol{x}} = \boldsymbol{y} \oplus \hat{\boldsymbol{e}} = (0110110)$. Here the four last bits equals the estimated information word, $\hat{\boldsymbol{u}} = (0110)$. This procedure will be able to correct all single errors.

If we instead try to introduce two errors, $\boldsymbol{e} = (0001010)$, the received vector is $\boldsymbol{y} = (0111100)$. Then the syndrome becomes $\boldsymbol{s} = (101)$, which according to the table corresponds to the error $\hat{\boldsymbol{e}} = (0000100)$ and the estimated codeword is $\hat{\boldsymbol{x}} = (0111000)$ and information word $\hat{\boldsymbol{u}} = (1000)$. This shows that we cannot correct double errors with this type of code.

To summarize *syndrome decoding* the steps are:

- Form a list with the most probable (least Hamming weight) error patterns for each possible syndrome.
- Derive the syndrome of the received vector and use the list to estimate the error pattern, $\hat{\boldsymbol{e}}$.
- Estimate the transmitted codeword as $\hat{\boldsymbol{x}} = \boldsymbol{y} \oplus \hat{\boldsymbol{e}}$.
- Derive the estimated information word $\hat{\boldsymbol{x}} \rightarrow \hat{\boldsymbol{u}}$

8.2.1 Bounds on block codes

To be done.

Hamming bound

Gilbert-Varshamov bound

8.3 Convolutional code

To be done.

8.3.1 Bounds on convolutional codes

To be done.