Chapter 6

Network information theory

In this Chapter we do not prove any converse theorem, since there are some technicalities that complicate the proofs. We present only achievability results. To avoid serious difficulties we use special techniques, based on typicality of sequences. This allows for very simple proofs. Unfortunately there is a serious drawback. If the encoders and decoders are designed as suggested in these proofs, the blocklengths that are required turn out to be extremely large. Therefore achievability is proved, but efficient codes (if they exist) are not unveiled.

6.1 Typical sequences

Typicality of sequences (i.e., of blocks of random variables) is a simple means to prove source coding and channel coding theorems. Thus, readers may wonder why this subject shows up so late in these lecture notes. The reason is that typicality provides definitely simple proofs, yet gives no useful hints about the blocklengths that are required to achieve the prescribed performance. The flavor of proofs based on typicality is always: For sufficiently large blocklength \( N \) this probability goes to zero, or that probability goes to 1. You let \( N \) grow beyond any finite value, and you see that the theorem is proved. If one tries to use such tools to predict which value of \( N \) is indeed sufficient, one invariably finds funny results. We shall see few examples soon.

It is fair to admit that in some cases proofs based on typical sequences give some useful hints about possible coding strategies. Therefore, the pros are simple proofs and possibly some suggestions for practical coding methods. The cons are absolutely no hints about the performance that can be achieved with finite blocklengths.

Let us consider a block of \( N \) independent (discrete) random outcomes \( X_1, X_2, \ldots, X_N \). The logarithm of \( 1/P(X_1, \ldots, X_N) \) is the sum of \( N \) independent random variables, each having expected value \( H(X) \). Let \( A_N^N(X) \) (\( A_N(X) \), for short) be the set of typical \( N \)-tuples
\(x_1, \ldots, x_N\) such that
\[
\left| \frac{1}{N} \log \frac{1}{P(x_1, \ldots, x_N)} - H(X) \right| \leq \varepsilon
\] (6.1)

Then, by the (weak) law of large numbers
\[
P\left( X_1, \ldots, X_N \notin A^N(X) \right) \leq \varepsilon
\] (6.2)

for \(N\) large enough\(^1\). In a sense, for sufficiently large \(N\) almost all the \(N\)-tuples are typical. Equation (6.1) is equivalent to
\[
2^{-N(H(X)+\varepsilon)} \leq P(x_1, \ldots, x_n) \leq 2^{-N(H(X)-\varepsilon)}
\] (6.3)

It would be tempting to say that all the \(N\)-tuples have almost the same probability, namely \(2^{-NH(X)}\). This is not correct, however. Even if \(\varepsilon \ll 1\), \(N\) may be much larger than \(1/\varepsilon\) so that \(N\varepsilon\) is very large as well. Yet, in a sense \(2^{\pm N\varepsilon}\) is a small correction term to \(2^{-NH(X)}\), since \(N\varepsilon \ll NH(X)\). We know that saying that \(P(x_1, \ldots, x_n) = 2^{-NH(X)}\) is wrong. Yet we build our proofs as it were true.

We expect that the cardinality of the typical set \(A^N(X)\), i.e., the number of typical sequences, be approximately equal to \(2^{NH(X)}\) since the sequences are disjoint, have (almost) the same probability, and their union is (almost) the certain set. More precisely, we have
\[
1 = \sum_x P(x_1, \ldots, x_N) \geq \sum_{x \in A^N(X)} P(x_1, \ldots, x_N)
\]
\[
\geq \sum_{x \in A^N(X)} 2^{-N(H(X)-\varepsilon)} = |A^N(X)|2^{-N(H(X)-\varepsilon)}
\] (6.4)
i.e.,
\[
|A^N(X)| \leq 2^{N(H(X)+\varepsilon)}
\] (6.5)

But for the term \(2^{N\varepsilon}\) this is the result we expected. Again, we think of \(2^{N\varepsilon}\) as a small correction term.

**Example 6.1.1.** Let \(X_1, \ldots, X_N\) be a block of independent binary variables, and let \(P(X_n = 1) = p\). The joint probability \(P(x_1, \ldots, x_N)\) depends only on the total number of zeros and ones in the \(N\)-tuple, namely
\[
P(x_1, \ldots, x_N) = p^K(1-p)^{N-K}
\] (6.6)

where \(K\) is the number of ones. It is easy to verify that (6.3) is equivalent to
\[
\delta \left| \log \frac{p}{1-p} \right| \leq \varepsilon
\] (6.7)

\(^1\) Usually the same value of \(\varepsilon\) is chosen in (6.1) and (6.2). This is definitely not the best choice, but simplifies the proofs.
where $\delta = \frac{K}{N} - p$ is the maximum deviation of the frequency of ones from the expected value $p$. If $p$ and $\varepsilon$ are given, also $\delta$ is known.

We want an $N$ that satisfies also (6.2). For instance, let $p = 0.1$ and $\varepsilon = 0.1$. The minimum blocklength that satisfies both conditions is $N = 229$. Typical sequences have a number of ones between 16 and 30. The corresponding joint probabilities are about $10^{-26}$ and $10^{-39}$, respectively. Admittedly, the (tiny) probabilities of typical sequences are hardly equal to each other. Yet, the probability that a random $N$-tuple is not typical is as large as $\varepsilon = 0.1$.

There are $3.8 \cdot 10^{37}$ typical sequences$^2$.

For large values of $N$, by the central limit theorem (6.2) is equivalent to

$$2Q\left(\frac{\delta\sqrt{N}}{\sqrt{p(1-p)}}\right) \leq \varepsilon$$

(6.8)

Combining this with (6.7) we eventually get

$$N \geq p(1-p)\left(\frac{Q^{-1}(\varepsilon/2)}{\delta}\right)^2$$

(6.9)

For instance, let $p = 0.1$ and $\varepsilon = 0.01$. It turns out that $N = 6 \cdot 10^4$, at least. The probability that a random sequence is not typical is close to 0.01. However, using the well-known inequality $Q(z) \leq \exp(-z^2/2)/2$ we see from (6.8) that the probability decreases exponentially if $N$ is increased further. The number of typical sequences is just a little less than a huge $2^{N(H(X)+\varepsilon)} \approx 10^{8650}$.

If we wanted a smaller value of $\varepsilon$ the blocklength would be amazingly large. About ten million bits, and more than one billion bits are needed to obtain $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-4}$, respectively.

## 6.2 Another look at the source coding theorem

Let us look for a proof of the source coding theorem based on typical sequences. For simplicity, we consider only sources without memory.

Once the source has produced the $N$-tuple $x_1, \ldots, x_N$ we check whether it is typical or not. For typical sequences we use a fixed-length code. Since the number of typical sequences is less than $2^{N(H(X)+\varepsilon)}$ we need at most $\lceil N(H(X) + \varepsilon) \rceil$ bits.

As to non typical sequences, we have two options. The first one is not to encode these sequences, and declare an error. Of course, this can be accepted only if $\varepsilon$ is extremely small. Otherwise, we can transmit the $N$-tuple as it is, without any attempt at compressing it. We need an additional prefix bit that specifies whether the sequence is typical or not. The average code length is upper bounded by

$$E[L] \leq N(H(X) + \varepsilon) + 1 + N\varepsilon + 1$$

(6.10)

$^2$The bound (6.5) gives $1.7 \cdot 10^{39}$. 

bits per block. Letting $\varepsilon \to 0$ and $N \to \infty$, the average number of bits per message $E[L]/N$ gets close to the source entropy $H(X)$ as much as we want.

Basically, this was the proof given by Shannon in his 1948 paper.

When we produce this proof it is tempting to take the following shortcuts. We do not mind $\varepsilon$, since it can be reduced as much as we want. We do not even mind non typical sequences, since their probability is less than $\varepsilon$. Then, the proof goes as follows. There are $2^{NH(X)}$ typical sequences, that can be encoded using $NH(X)$ bits, i.e., $H(X)$ bits per message.

**Example 6.2.1.** Let us consider a memoryless binary source, that produces ones with probability $p = 0.1$. We choose $\varepsilon = 0.1$ and $N = 229$. We have to manage more than $10^{37}$ codewords for the typical sequences. Yet, the predicted code performance is disappointing$^3$. According to (6.10) we need about 0.68 bits per message, whereas the source entropy is $H(X) = 0.469$.

Next we take $\varepsilon = 0.01$ and $N = 60000$. Now we need less than 0.489 bits per message, a very good result. Do not try to evaluate the size of the codebook, yet.

Readers wondering about the performance of a Huffman code with very short $N$-tuples, can check the following results. With $N = 2$ the average code length is 0.645 bits, and with $N = 6$ only 0.470 bits are needed. The codebook sizes are 4 and 64, respectively. No amazingly large blocklengths are needed to design very efficient codes for this source.

### 6.3 Jointly typical sequences

From the above discussion we see that typicality is a very good tool for producing simple proofs of theorems, but it is not to be trusted for designing efficient encoders.

Anyway, let’s go on considering joint $N$-tuples $X = X_1, \ldots, X_N$ and $Y = Y_1, \ldots, Y_N$. Let $P(x, y)$ be the joint probability distribution, and let $P(x)$ and $P(y)$ be the marginal distributions. The pair $(x, y)$ is jointly typical if$^4$

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

We let $A^N_X(Y)$ ($A^N(X, Y)$, for short) be the set of jointly typical $N$-tuples.

$^3$An error is encoding *all the typical sequences* with the same number of bits. We know that the probabilities of $N$-tuples belonging to the typical set have huge variations.

$^4$We could choose different values of $\varepsilon$, say $\varepsilon_X, \varepsilon_Y$ and $\varepsilon_{XY}$ in these inequalities. Choosing the same value simplifies the proofs based on typicality.
Some properties mimic those already known for just one $N$-tuple. For instance, for sufficiently large $N$ almost all sequences $(\mathbf{X}, \mathbf{Y})$ are jointly typical, and the cardinality of the jointly typical set $\mathcal{A}_N(X, Y)$ is less than $2^{N(H(X,Y)+\varepsilon)}$.

Let $\mathbf{X}$ be a randomly chosen codeword, transmitted through a channel, and let $\mathbf{Y}$ be the channel output. The output is generated according to the conditional probability distribution $P(\mathbf{y}|\mathbf{x})$. Therefore, the joint probability distribution is $P(\mathbf{x}, \mathbf{y}) = P(\mathbf{x})P(\mathbf{y}|\mathbf{x})$.

With high probability the input and output $N$-tuples are jointly typical, if $N$ is large.

Let $\mathbf{X}$ be any other codeword, generated independently of $\mathbf{X}$. We may wonder about the probability that $\mathbf{X}$ and $\mathbf{Y}$ are jointly typical. Since they are generated independently of each other, with probability distribution $P(\mathbf{x})P(\mathbf{y})$, we expect that the probability that they are jointly typical be small. Let us check. We have

$$P(\mathbf{X}, \mathbf{Y} \in \mathcal{A}_N(X, Y)) = \sum_{\mathbf{x}, \mathbf{y} \in \mathcal{A}_N(X, Y)} P(\mathbf{x})P(\mathbf{y})$$

$$\leq \sum_{\mathbf{x}, \mathbf{y} \in \mathcal{A}_N(X, Y)} 2^{-N(H(X)-\varepsilon)} 2^{-N(H(Y)-\varepsilon)}$$

$$\leq 2^{N(H(X,Y)+\varepsilon)} 2^{-N(H(X)-\varepsilon)} 2^{-N(H(Y)-\varepsilon)} \leq 2^{-N(I(X,Y)-3\varepsilon)} \quad (6.12)$$

As usual, $3\varepsilon$ is much smaller than $I(X,Y)$ if $N$ is large enough. The probability that a randomly generated codeword mimics the codeword that has been transmitted is very small. We can have many codewords, and still be able to pick the transmitted one just checking for joint typicality. This is the basis of a short proof of the channel coding theorem.

### 6.4 Another look at the channel coding theorem

As usual, let $2^{NR}$ codewords of length $N$ be generated independently of each other. The decoder browses the set of codewords, to find the unique codeword jointly typical with the received $N$-tuple. If there is none, or there are many, the decoder declares an error. If there is just one, but it is a wrong codeword, we have an error as well.

Then we have an error whenever the transmitted codeword and the received $N$-tuple are not jointly typical, or at least one wrong codeword is jointly typical with $\mathbf{Y}$. Since these are not disjoint events, we invoke the union bound.

The probability that $\mathbf{X}$ is not jointly typical with $\mathbf{Y}$ is less than $\varepsilon$. The probability that a wrong codeword $\overline{\mathbf{X}}$ and $\mathbf{Y}$ are jointly typical is less than $2^{-N(I(X,Y)-3\varepsilon)}$. Then, provided the transmission rate $R$ is less than $I(X, Y) - 4\varepsilon$ the probability that there is at least one typical sequence is less than $2^{-N\varepsilon}$, which is less than $\varepsilon$ for sufficiently large $N$. Then the probability of error is less than $2\varepsilon$. This holds, on average, for all possible channel codes designed randomly. There exists at least one such code whose probability of error is not greater than the average. Basically, this is the proof given by Shannon in 1948.
It can be shown that if we generate $2^{N\left(I(X,Y) + 4\varepsilon\right)}$ codewords $\mathbf{X}$ the probability that none of them is jointly typical with $Y$ is less than $2\varepsilon$, so that the decoder based on joint typicality almost surely fails to uncover the transmitted codeword. This helps realize why the transmission rate cannot be greater than $I(X,Y)$. It is not a proof, yet, since we could devise a better receiver.

### 6.5 Slepian-Wolf source coding

A source generates pairs $(X, Y)$ of correlated random variables. Of course, we can encode the pair with just a little more than $H(X, Y)$ bits per pair. The job cannot be done with less than $H(X, Y)$ bits.

Now suppose that $X$ and $Y$ are to be encoded separately, using two encoders that do not communicate with each other. Of course, $H(X) + H(Y)$ bits are sufficient. Yet, this is more than $H(X, Y)$. Surprisingly, it turns out that only $H(X, Y)$ bits are needed, even with separate encoders. The decoder receives both streams, and recovers the pair $(X, Y)$.

Let $Y$ be encoded with $H(Y)$ bits per message so that $Y$ is recovered without errors. We want to show that $X$ can be encoded spending only $H(X|Y)$ bits per message, and that the decoder is able to recover also $X$.

The trick is random labeling (or random binning). Each $x$ is given a random integer label $L(x)$, such that the probability distribution of labels is uniform in the range $(1, M)$. A suitable hash function can do this job. The label is transmitted. The cardinality of $\{X\}$ is much greater than $M$, so that the same label corresponds to many different blocks.

The decoder, upon receiving the label, looks for a block $x$ with this label that is also jointly typical with $y$. We have an error if the $N$-tuple $x$ produced by the source and $y$ are not jointly typical. The corresponding probability is less than $\varepsilon$.

An error occurs also if at least one wrong $\mathbf{x}$ having the correct label is jointly typical with $y$. If $M = 2^{N(H(X|Y)+3\varepsilon)}$ the probability of error is less than

$$P \left( L(\mathbf{X}) = L(\mathbf{X}) \right) \leq 2^{N(H(X|Y)+2\varepsilon)} 2^{-N(H(X|Y)+3\varepsilon)} = 2^{-N\varepsilon} \leq \varepsilon \quad (6.13)$$

for sufficiently large $N$.

More generally, it can be shown that we may encode $X$ and $Y$ with less than $H(X)$ and $H(Y)$ bits per message, respectively. We want to determine the allowable rate pairs $(R_X, R_Y)$.

It turns out that the following conditions are sufficient.

$$R_X \geq H(X|Y)$$

$$R_Y \geq H(Y|X)$$

$$R_X + R_Y \geq H(X,Y) \quad (6.14)$$
6.6. MULTIPLE ACCESS CHANNELS

The proof is similar. We have only to give random labels to both $X$ and $Y$. There are $2^{NR_X}$ and $2^{NR_Y}$ labels, respectively.

Readers can verify that the reason why $R_X$ cannot be less than $H(X|Y)$ is that there would be too many candidates $\bar{x}$ having the right label. Almost surely at least one of them would be jointly typical with $y$, and the decoder would fail.

Comment: We could even consider the case in which there is no $Y$ to encode. Only $X$ is to be encoded. The conditional entropy is $H(X|Y) = H(X)$. This is means that at least $H(X)$ bits are needed to encode each message produced by the source, as expected. The intriguing point is that we could design the source code by random labeling. The decoder has to look for the unique block $x$ which is typical and has the label that has been received. A failure occurs if $x$ is not typical or more than one typical sequences have the correct label. Needless to say, this is only theory. Huge blocklengths would be required, even for simple sources.

6.6 Multiple access channels

A multiple access channel (MAC) has (at least) two inputs $X_1, X_2$ and one output $Y$. The channel transition probabilities are given by the conditional probabilities $P(y|x_1, x_2)$, for all possible values of $x_1, x_2$ and $y$.

The peculiarity of the MAC model is that the inputs are independent of each other. The two transmitters are not allowed to select the pair $(X_1, X_2)$ jointly. They can not cooperate with each other. The channel capacity is the maximum of $I(X_1X_2, Y)$ with respect to the joint probability distribution $P(x_1, x_2)$. Thus each transmitter sends its own information, and the task of the receiver is to recover both streams by some suitable decoder.

The probability distribution of the inputs must be the product of the marginal distributions, namely $P(x_1, x_2) = P(x_1)P(x_2)$. In general, we are free to choose the marginal distributions. For a while, yet, let us assume that $P(x_1)$ and $P(x_2)$ are fixed.

We are interested in the rate pairs $(R_1, R_2)$ that can be achieved. Also the sum of the rates, namely the sum-rate $R_1 + R_2$, i.e. the maximum rate of information travelling through the MAC, plays an interesting role.

It can be shown that the MAC capacity region, namely the set of all possible rate pairs, is determined by the following inequalities:

$$
0 \leq R_1 \leq I(X_1, Y|X_2) \\
0 \leq R_2 \leq I(X_2, Y|X_1) \\
0 \leq R_1 + R_2 \leq I(X_1X_2, Y)
$$

(6.15)

All points inside this pentagon are achievable.

5For instance, if $R_X = 0$ all the $N$-tuples have the same label.
6Or the conditional probability densities, for continuous alphabets.
7If cooperation is allowed we have a single-input single output channel, where the input is the pair $(X_1, X_2)$.
Before going on, let us discuss the two corners that characterize the pentagon, giving an intuitive explanation of the operation of the encoders and of the decoder. One corner is

$$R_1 = I(X_1, Y|X_2), \quad R_2 = I(X_2, Y).$$

The corresponding sum-rate is

$$R_1 + R_2 = I(X_1, Y|X_2) + I(X_2, Y) = I(X_1X_2, Y).$$

This point is achieved in the following way. Sender 1 transmits at its maximum rate $R_1 = I(X_1, Y|X_2)$, with probability distribution $P(x_1)$. For the channel from $X_2$ to $Y$ this is random noise, independent of $X_2$. Therefore sender 2 can transmit at rates up to the mutual information $I(X_2, Y)$, evaluated with the input distribution $P(x_1, x_2) = P(x_1)P(x_2)$ that is being used, and the channel transition probabilities $P(y|x_1, x_2)$. Of course, achieving a rate $R_2$ close to the mutual information requires very good codes, almost capacity achieving.

Thus the decoder can recover the second stream $X_2$. In a sense, the decoder can then subtract the effect of $X_2$. More precisely, $X_1$ is decoded taking care of the channel transition probabilities $P(y|x_1, x_2)$, where $x_2$ is known. The corresponding mutual information is $I(X_1, Y|x_2)$, whose average with respect to $P(x_2)$ is $I(X_1, Y|X_2)$. This is the maximum rate for the first sender.

Of course, $R_2$ can be reduced as much as we want. We have simply to use less powerful codes for the second sender. We can even transmit no information at all ($R_2 = 0$), just keeping sending random (and useless) sequences with probability distribution $P(x_2)$.

We can exchange the roles of the two senders, achieving the pair $R_1 = I(X_1, Y), \quad R_2 = I(X_2, Y|X_1)$. In this case, the receiver decodes the first stream, subtracts its effect, and finally decodes the second stream. Again the sum-rate is $R_1 + R_2 = I(X_1, Y) + I(X_2, Y|X_1) = I(X_1X_2, Y)$.

Finally, if two pairs $(R_1, R_2)$ and $(R'_1, R'_2)$ are achievable also the pair $(\lambda R_1 + (1-\lambda)R'_1, \lambda R_2 + (1-\lambda)R'_2)$, with $0 \leq \lambda \leq 1$, is achievable by time-sharing (i.e., time division). The two strategies are used for a fraction of time $\lambda$ and $1-\lambda$, respectively.

An important consequence of the time-sharing strategy is that the capacity region is convex. Since we can modify the input distributions $P(x_1)$ and $P(x_2)$, obtaining different pentagons, the capacity region is the convex closure (i.e., the convex hull) of the union of all these regions.

### 6.6.1 Proof of achievability

The above reasoning was based on intuition. Yet, it gave the idea of successive decoding, with interference cancellation as a means to design practical multiple access transmission systems. Now we want to give a formal proof of achievability, based on typical sequences. Transmission is very simple. Each sender has a random codebook, with $2^{NR_1}$ and $2^{NR_2}$ codewords. Upon receiving $y$ the decoder looks for the unique pair $(x_1, x_2)$ which is jointly typical with $y$ (i.e., $(x_1, x_2, y) \in \mathcal{A}^N(X_1, X_2, Y)$). We have the following errors: i) the transmitted pair is not jointly typical with $y$; ii) there exist a pair $(\widetilde{x}_1, x_2)$ jointly typical
6.7. GAUSSIAN MULTIPLE ACCESS CHANNEL

with $y$; iii) there exist a pair $(x_1, x_2)$ jointly typical with $y$; and iv) there exist a pair $(\bar{x}_1, \bar{x}_2)$ jointly typical with $y$.

The probability of the first error is less than $\varepsilon$. The probability of the second error, for a specific $\bar{x}_1$, is upper bounded by

$$P \left( (\bar{x}_1, x_2, y) \in A^N(X_1, X_2, Y) \right) \leq 2^{-N(I(X_1, X_2 Y) - 3\varepsilon)}$$  \hspace{1cm} (6.16)

The derivation is similar to (6.12). The only difference is that $y$ is substituted by the pair $(x_2, y)$. There are $2^{NR_1}$ such terms. If $R_1 \leq I(X_1, Y|X_2) - 4\varepsilon$, the probability of the second type of errors is less than

$$2^{N(I(X_1, Y|R_2) - 4\varepsilon)} 2^{-N(I(X_1, X_2 Y) - 3\varepsilon)} \leq 2^{-N\varepsilon}$$  \hspace{1cm} (6.17)

since $I(X_1, X_2) = 0$. Similarly, errors of the third type have probability less than $2^{-N\varepsilon}$.

Finally the probability of the fourth error, for a specific pair $(\bar{x}_1, \bar{x}_2)$, is upper bounded by

$$P \left( (\bar{x}_1, \bar{x}_2, y) \in A^N(X_1, X_2, Y) \right) \leq 2^{-N(I(X_1, X_2 Y) - 3\varepsilon)}$$  \hspace{1cm} (6.18)

There are $2^{N(R_1 + R_2)}$ such terms. If $R_1 + R_2 \leq I(X_1 X_2, Y) - 4\varepsilon$, the probability of the fourth type of errors is less than

$$2^{N(I(X_1 X_2 Y) - 4\varepsilon)} 2^{-N(I(X_1 X_2 Y) - 3\varepsilon)} \leq 2^{-N\varepsilon}$$  \hspace{1cm} (6.19)

Since $2^{-N\varepsilon}$ can be reduced as much as we want, achievability is proved. We must observe that this proof does not suggest successive decoding, nor interference cancellation either.

It does not even suggest time-sharing.

6.7 Gaussian multiple access channel

Now we consider a Gaussian channel with two inputs $X_1$, $X_2$ and power limitations $P_1$, $P_2$. The channel output is $Y = X_1 + X_2 + Z$, where $Z$ is Gaussian noise with variance $\sigma^2$.

To evaluate easily every mutual information we need, we assume that $X_1$ and $X_2$ are Gaussian, too.\footnote{Yes, we know that Gaussian inputs are never used. Yet, we also know that at small transmission rates a two-level input $X = \pm 1$ works as well as a Gaussian input. Even at high rates there exist multilevel constellations that incur no capacity degradation.}

We assume, almost without saying, that the above formulas for the MAC capacity region hold also for continuous inputs and outputs.
Readers can easily check the almost obvious equations

\[
I(X_1, Y|X_2) = \frac{1}{2} \log \left(1 + \frac{P_1}{\sigma^2}\right)
\]
\[
I(X_2, Y|X_1) = \frac{1}{2} \log \left(1 + \frac{P_2}{\sigma^2}\right)
\]
\[
I(X_1X_2, Y) = \frac{1}{2} \log \left(1 + \frac{P_1 + P_2}{\sigma^2}\right)
\]

(6.20)

Note that, due to the power limitations we are not free to choose the probability densities \(p(x_1)\) and \(p(x_2)\) of the inputs: \(X_1\) and \(X_2\) are Gaussian (or equivalent to Gaussian) random variables and have variance \(P_1\) and \(P_2\), respectively. Therefore the capacity region is not a union of pentagons. It is just one pentagon.

Interestingly, the sum-rate (i.e., the total information that can be sent through the channel) is the same of a single transmitter that uses the total power \(P_1 + P_2\), even if two independent streams are transmitted and these streams interfere with each other. Note however, that the total sum-rate can not be assigned to just one user.

Let us consider one corner of the capacity region. First, the receiver decodes \(X_1\). The second signal is Gaussian interference, with variance \(P_2\). Then, the maximum code rate is

\[
R_1 = \frac{1}{2} \log \left(1 + \frac{P_1}{P_2 + \sigma^2}\right)
\]

(6.21)

After successful decoding, \(X_1\) is subtracted\(^9\). Since the interference for \(X_2\) has been removed, the rate of the second stream can be as large as

\[
R_2 = \frac{1}{2} \log \left(1 + \frac{P_2}{\sigma^2}\right)
\]

(6.22)

Finally, it is easy to check that

\[
R_1 + R_2 = \frac{1}{2} \log \left(\frac{P_1 + P_2 + \sigma^2}{P_2 + \sigma^2}\right) + \frac{1}{2} \log \left(\frac{P_2 + \sigma^2}{\sigma^2}\right) = \frac{1}{2} \log \left(1 + \frac{P_1 + P_2}{\sigma^2}\right)
\]

(6.23)

as expected. Of course, the roles of \(X_1\) and \(X_2\) can be exchanged. Time-sharing allows for all intermediate combinations of \(R_1\) and \(R_2\).

### 6.7.1 Simple time-sharing

One could expect that time-sharing produces the best channel utilization, since it avoids mutual interference. Let the first user transmit with power \(P_1\) for a fraction \(\alpha\) of time, and

\(^9\)Note that in the Gaussian MAC cancellation means subtraction. Of course, one must take care of subtracting \(X_1\) with its exact amplitude, and time delay.
the second user transmit with power $P_2$ for the rest of time. The total rate
\[
R_1 + R_2 = \frac{\alpha}{2} \log \left( 1 + \frac{P_1}{\sigma^2} \right) + \frac{1-\alpha}{2} \log \left( 1 + \frac{P_2}{\sigma^2} \right)
\]
is smaller than possible. We can observe that the average power is $\alpha P_1 + (1-\alpha)P_2$, which is less than $P_1 + P_2$. Maybe the instantaneous powers can be increased\textsuperscript{10}. User one increases its power to $P_1/\alpha$, so that its average power is $P_1$. The same does user two. The total rate
\[
R_1 + R_2 = \frac{\alpha}{2} \log \left( 1 + \frac{P_1}{\alpha \sigma^2} \right) + \frac{1-\alpha}{2} \log \left( 1 + \frac{P_2}{(1-\alpha)\sigma^2} \right)
\]
is strictly less than possible, but for one point. If $\alpha = P_1/(P_1 + P_2)$ it is easy to verify that
\[
\begin{align*}
R_1 + R_2 &= \frac{1}{2} \log \left( 1 + \frac{P_1 + P_2}{\sigma^2} \right) \\
R_1 &= \alpha (R_1 + R_2) \\
R_2 &= (1-\alpha)(R_1 + R_2)
\end{align*}
\]
So there exist just one point on the boundary of the MAC capacity region that can be achieved also by time-sharing. The rates $R_1$ and $R_2$ are proportional to the transmitted powers. If we want to design a MAC that works at these rates, the advantage is that no joint receiver is needed.

Similar results are obtained for frequency division. The MAC is efficient if the bandwidth allocated to each user is proportional to its power.

All the above results can be extended to multiple access channels with many users. For instance, for three users the capacity region is three-dimensional. Instead of the two corners that characterize the capacity region for two users, there are six corners. Each corner corresponds to a specific ordering of successive decoding. In fact, there are $3! = 6$ possible orderings.

### 6.8 Broadcast channels

A broadcast channel has one input terminal that wants to transmit information to two (or more than two) end users. In many cases there is some common information sent to both terminals, and some additional information only for the better receiver.

The capacity region is still unknown, in general. It is known, and easy to evaluate, for the Gaussian channel. It is known, but not as easy to evaluate, also for the so called *degraded* channels. For simplicity we shall consider only the Gaussian case.

\textsuperscript{10}Provided the amplifiers allow for this. Transmitting 10 W continuously is different from transmitting 20 W half the time.
6.8.1 The Gaussian broadcast channel

Let $P$ be the power of the channel input $X$. As we did for the MAC, we assume that $X$ is Gaussian (or equivalent to a Gaussian random variable) without spending time to prove that this is the best we can do.

The channel outputs are $Y_1 = X + Z_1$ and $Y_2 = X + Z_2$, where $Z_1$ and $Z_2$ are independent and have variance $\sigma_1^2$ and $\sigma_2^2$. Without loss of generality let $\sigma_1^2 \leq \sigma_2^2$.

The transmitted signal is $X = X_1 + X_2$. Fractions $\alpha$ and $1 - \alpha$ of the power are used to transmit information towards the first and second terminal. The rate for the second user does not exceed

$$R_2 = \frac{1}{2} \log \left( 1 + \frac{(1 - \alpha)P}{\alpha P + \sigma_2^2} \right) \quad (6.27)$$

Note that the power $\alpha P$ transmitted to the first user is Gaussian noise for the second one.

The decoder of the first user can decode the information sent to the second user, since its channel is better. This is the common information. Once $X_2$ is available, it can be subtracted from $Y$. Since there is no residual interference the rate $R_1$ can be as high as

$$R_1 = \frac{1}{2} \log \left( 1 + \frac{\alpha P}{\sigma_1^2} \right) \quad (6.28)$$

The capacity region is obtained exploring all values of $\alpha$. Note that if $\sigma_1^2 = \sigma_2^2 = \sigma^2$, the sum-rate turns out to be a constant, namely

$$R_1 + R_2 = \frac{1}{2} \log \left( 1 + \frac{P}{\sigma^2} \right) \quad (6.29)$$

This can be achieved also by time-sharing. However, if the noise variances are different the sum-rate obtained by successive decoding and cancellation is always greater than that obtained by time-sharing (for $\alpha$ different from 0 or 1, of course).