17

Communication over Constrained Noiseless Channels

In this chapter we study the task of communicating efficiently over a constrained noiseless channel – a constrained channel over which not all strings from the input alphabet may be transmitted.

We make use of the idea introduced in Chapter 16, that *global properties of graphs can be computed by a local message-passing algorithm.*

▶ 17.1 Three examples of constrained binary channels

A constrained channel can be defined by rules that define which strings are permitted.

Example 17.1. In Channel A every `1` must be followed by at least one `0`.

Channel A: the substring `11` is forbidden.

A valid string for this channel is

$$\texttt{00100101001010100010}. \tag{17.1}$$

As a motivation for this model, consider a channel in which `1`s are represented by pulses of electromagnetic energy, and the device that produces those pulses requires a recovery time of one clock cycle after generating a pulse before it can generate another.

Example 17.2. Channel B has the rule that all `1`s must come in groups of two or more, and all `0`s must come in groups of two or more.

Channel B: `101` and `010` are forbidden.

A valid string for this channel is

$$\texttt{00111001110011000011}. \tag{17.2}$$

As a motivation for this model, consider a disk drive in which successive bits are written onto neighbouring points in a track along the disk surface; the values `0` and `1` are represented by two opposite magnetic orientations. The strings `101` and `010` are forbidden because a single isolated magnetic domain surrounded by domains having the opposite orientation is unstable, so that `101` might turn into `111`, for example.

Example 17.3. Channel C has the rule that the largest permitted runlength is two, that is, each symbol can be repeated at most once.

Channel C: `111` and `000` are forbidden.

A valid string for this channel is

$$\texttt{10010011011001101001}. \tag{17.3}$$

A physical motivation for this model is a disk drive in which the rate of rotation of the disk is not known accurately, so it is difficult to distinguish between a string of two 1s and a string of three 1s, which are represented by oriented magnetizations of duration 2τ and 3τ respectively, where τ is the (poorly known) time taken for one bit to pass by; to avoid the possibility of confusion, and the resulting loss of synchronization of sender and receiver, we forbid the string of three 1s and the string of three 0s.

All three of these channels are examples of *runlength-limited channels*. The rules constrain the minimum and maximum numbers of successive 1s and 0s.

Channel	Runlength of 1s		Runlength of 0s	
	minimum	maximum	minimum	maximum
unconstrained	1	∞	1	∞
A	1	1	1	∞
B	2	∞	2	∞
C	1	2	1	2

In channel A, runs of 0s may be of any length but runs of 1s are restricted to length one. In channel B all runs must be of length two or more. In channel C, all runs must be of length one or two.

The capacity of the unconstrained binary channel is one bit per channel use. What are the capacities of the three constrained channels? [To be fair, we haven't defined the 'capacity' of such channels yet; please understand 'capacity' as meaning how many bits can be conveyed reliably per channel-use.]

Some codes for a constrained channel

Let us concentrate for a moment on channel A, in which runs of 0s may be of any length but runs of 1s are restricted to length one. We would like to communicate a random binary file over this channel as efficiently as possible.

A simple starting point is a $(2, 1)$ code that maps each source bit into two transmitted bits, C_1. This is a rate-$1/2$ code, and it respects the constraints of channel A, so the capacity of channel A is at least 0.5. Can we do better?

C_1 is redundant because if the first of two received bits is a zero, we know that the second bit will also be a zero. We can achieve a smaller average transmitted length using a code that omits the redundant zeroes in C_1.

C_2 is such a *variable-length* code. If the source symbols are used with equal frequency then the average transmitted length per source bit is

$$L = \frac{1}{2}1 + \frac{1}{2}2 = \frac{3}{2}, \tag{17.4}$$

so the average communication rate is

$$R = 2/3, \tag{17.5}$$

and the capacity of channel A must be at least $2/3$.

Can we do better than C_2? There are two ways to argue that the information rate could be increased above $R = 2/3$.

The first argument assumes we are comfortable with the entropy as a measure of information content. The idea is that, starting from code C_2, we can reduce the average message length, without greatly reducing the entropy

Code C_1

s	t
0	00
1	10

Code C_2

s	t
0	0
1	10

of the message we send, by decreasing the fraction of 1s that we transmit. Imagine feeding into C_2 a stream of bits in which the frequency of 1s is f. [Such a stream could be obtained from an arbitrary binary file by passing the source file into the decoder of an arithmetic code that is optimal for compressing binary strings of density f.] The information rate R achieved is the entropy of the source, $H_2(f)$, divided by the mean transmitted length,

$$L(f) = (1-f) + 2f = 1 + f. \tag{17.6}$$

Thus

$$R(f) = \frac{H_2(f)}{L(f)} = \frac{H_2(f)}{1+f}. \tag{17.7}$$

The original code C_2, without preprocessor, corresponds to $f = 1/2$. What happens if we perturb f a little towards smaller f, setting

$$f = \frac{1}{2} + \delta, \tag{17.8}$$

for small negative δ? In the vicinity of $f = 1/2$, the denominator $L(f)$ varies linearly with δ. In contrast, the numerator $H_2(f)$ only has a second-order dependence on δ.

▷ Exercise 17.4.[1] Find, to order δ^2, the Taylor expansion of $H_2(f)$ as a function of δ.

To first order, $R(f)$ increases linearly with decreasing δ. It must be possible to increase R by decreasing f. Figure 17.1 shows these functions; $R(f)$ does indeed increase as f decreases and has a maximum of about 0.69 bits per channel use at $f \simeq 0.38$.

By this argument we have shown that the capacity of channel A is at least $\max_f R(f) = 0.69$.

▷ Exercise 17.5.[2, p.257] If a file containing a fraction $f = 0.5$ 1s is transmitted by C_2, what fraction of the transmitted stream is 1s?

What fraction of the transmitted bits is 1s if we drive code C_2 with a sparse source of density $f = 0.38$?

A second, more fundamental approach *counts* how many valid sequences of length N there are, S_N. We can communicate $\log S_N$ bits in N channel cycles by giving one name to each of these valid sequences.

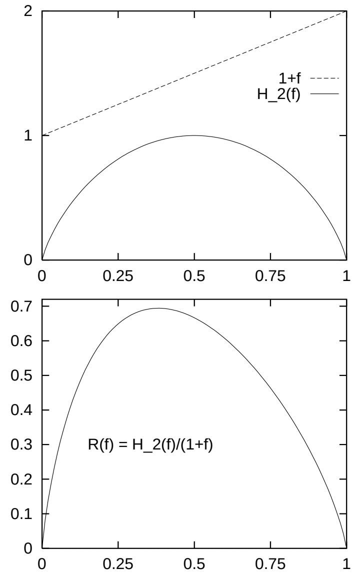


Figure 17.1. Top: The information content per source symbol and mean transmitted length per source symbol as a function of the source density. Bottom: The information content per transmitted symbol, in bits, as a function of f.

▶ 17.2 The capacity of a constrained noiseless channel

We defined the capacity of a noisy channel in terms of the mutual information between its input and its output, then we proved that this number, the capacity, was related to the number of distinguishable messages $S(N)$ that could be reliably conveyed over the channel in N uses of the channel by

$$C = \lim_{N\to\infty} \frac{1}{N} \log S(N). \tag{17.9}$$

In the case of the constrained noiseless channel, we can adopt this identity as our definition of the channel's capacity. However, the name s, which, when we were making codes for noisy channels (section 9.6), ran over messages $s = 1, \ldots, S$, is about to take on a new role: labelling the states of our channel;

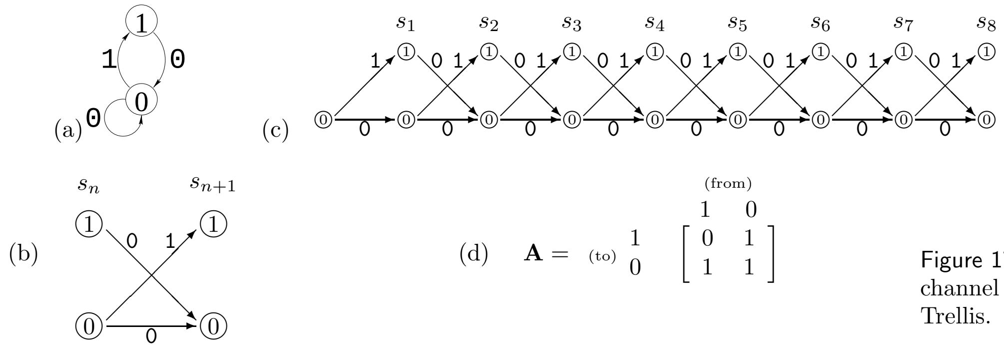


Figure 17.2. (a) State diagram for channel A. (b) Trellis section. (c) Trellis. (d) Connection matrix.

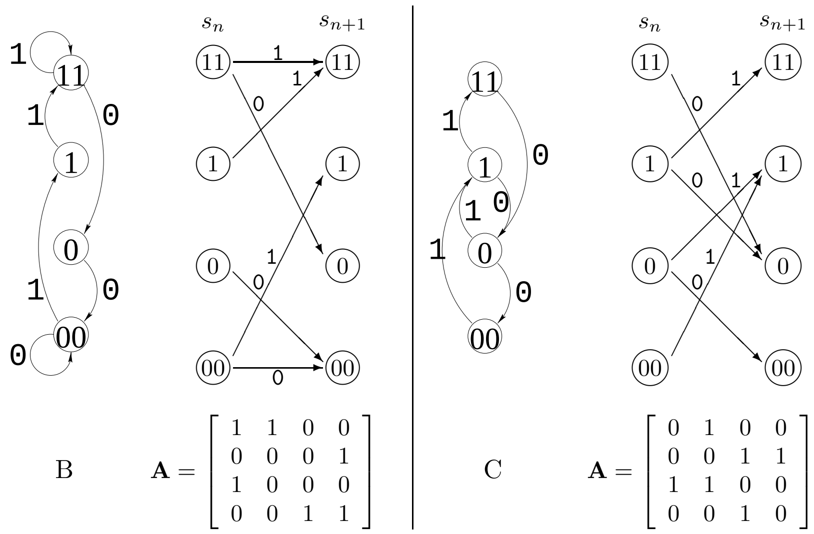


Figure 17.3. State diagrams, trellis sections and connection matrices for channels B and C.

so in this chapter we will denote the number of distinguishable messages of length N by M_N, and define the capacity to be:

$$C = \lim_{N\to\infty} \frac{1}{N} \log M_N. \tag{17.10}$$

Once we have figured out the capacity of a channel we will return to the task of making a practical code for that channel.

▶ 17.3 Counting the number of possible messages

First let us introduce some representations of constrained channels. In a *state diagram*, states of the transmitter are represented by circles labelled with the name of the state. Directed edges from one state to another indicate that the transmitter is permitted to move from the first state to the second, and a label on that edge indicates the symbol emitted when that transition is made. Figure 17.2a shows the state diagram for channel A. It has two states, 0 and 1. When transitions to state 0 are made, a `0` is transmitted; when transitions to state 1 are made, a `1` is transmitted; transitions from state 1 to state 1 are not possible.

We can also represent the state diagram by a *trellis section*, which shows two successive states in time at two successive horizontal locations (figure 17.2b). The state of the transmitter at time n is called s_n. The set of possible state sequences can be represented by a *trellis* as shown in figure 17.2c. A valid sequence corresponds to a path through the trellis, and the number of

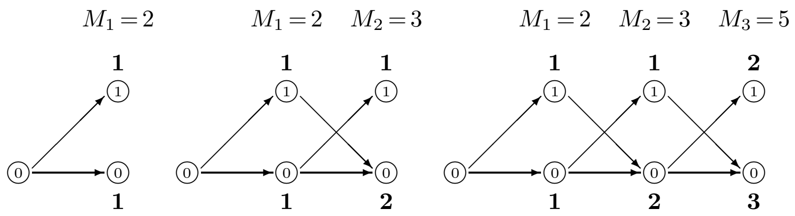


Figure 17.4. Counting the number of paths in the trellis of channel A. The counts next to the nodes are accumulated by passing from left to right across the trellises.

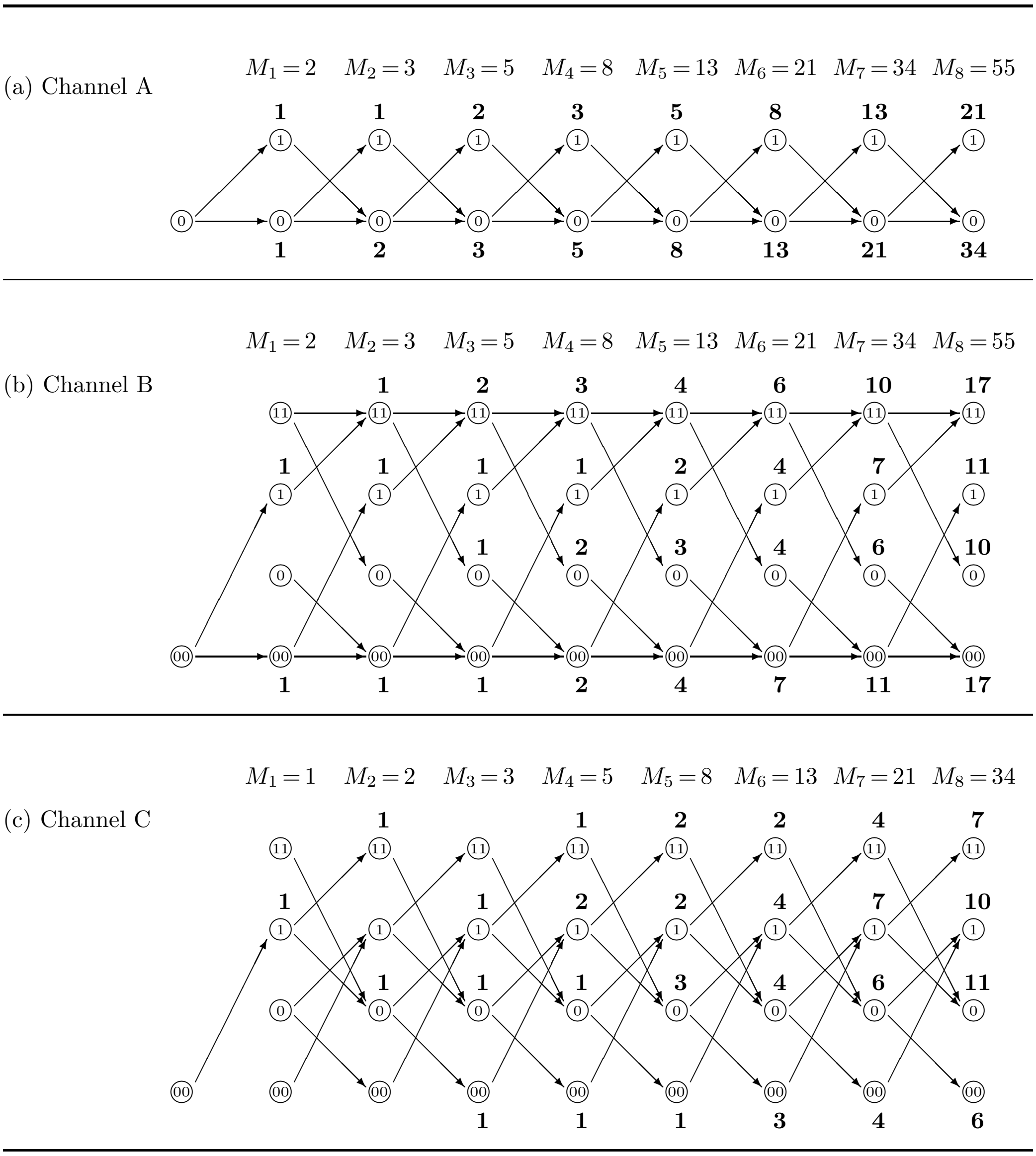


Figure 17.5. Counting the number of paths in the trellises of channels A, B, and C. We assume that at the start the first bit is preceded by `00`, so that for channels A and B, any initial character is permitted, but for channel C, the first character must be a `1`.

n	M_n	M_n/M_{n-1}	$\log_2 M_n$	$\frac{1}{n}\log_2 M_n$
1	2		1.0	1.00
2	3	1.500	1.6	0.79
3	5	1.667	2.3	0.77
4	8	1.600	3.0	0.75
5	13	1.625	3.7	0.74
6	21	1.615	4.4	0.73
7	34	1.619	5.1	0.73
8	55	1.618	5.8	0.72
9	89	1.618	6.5	0.72
10	144	1.618	7.2	0.72
11	233	1.618	7.9	0.71
12	377	1.618	8.6	0.71
100	9×10^{20}	1.618	69.7	0.70
200	7×10^{41}	1.618	139.1	0.70
300	6×10^{62}	1.618	208.5	0.70
400	5×10^{83}	1.618	277.9	0.69

Figure 17.6. Counting the number of paths in the trellis of channel A.

valid sequences is the number of paths. For the purpose of counting how many paths there are through the trellis, we can ignore the labels on the edges and summarize the trellis section by the *connection matrix* $\mathbf{A}$, in which $A_{ss'} = 1$ if there is an edge from state s to s', and $A_{ss'} = 0$ otherwise (figure 17.2d). Figure 17.3 shows the state diagrams, trellis sections and connection matrices for channels B and C.

Let's count the number of paths for channel A by message-passing in its trellis. Figure 17.4 shows the first few steps of this counting process, and figure 17.5a shows the number of paths ending in each state after n steps for $n = 1, \ldots, 8$. The total number of paths of length n, M_n, is shown along the top. We recognize M_n as the Fibonacci series.

▷ Exercise 17.6.[*1*] Show that the ratio of successive terms in the Fibonacci series tends to the golden ratio,

$$\gamma \equiv \frac{1+\sqrt{5}}{2} = 1.618. \tag{17.11}$$

Thus, to within a constant factor, M_N scales as $M_N \sim \gamma^N$ as $N \to \infty$, so the capacity of channel A is

$$C = \lim \frac{1}{N} \log_2 \left[\text{constant} \cdot \gamma^N\right] = \log_2 \gamma = \log_2 1.618 = 0.694. \tag{17.12}$$

How can we describe what we just did? The count of the number of paths is a vector $\mathbf{c}^{(n)}$; we can obtain $\mathbf{c}^{(n+1)}$ from $\mathbf{c}^{(n)}$ using:

$$\mathbf{c}^{(n+1)} = \mathbf{A}\mathbf{c}^{(n)}. \tag{17.13}$$

So

$$\mathbf{c}^{(N)} = \mathbf{A}^N \mathbf{c}^{(0)}, \tag{17.14}$$

where $\mathbf{c}^{(0)}$ is the state count before any symbols are transmitted. In figure 17.5 we assumed $\mathbf{c}^{(0)} = [0,1]^{\mathsf{T}}$, i.e., that either of the two symbols is permitted at the outset. The total number of paths is $M_n = \sum_s c_s^{(n)} = \mathbf{c}^{(n)} \cdot \mathbf{n}$. In the limit, $\mathbf{c}^{(N)}$ becomes dominated by the principal right-eigenvector of $\mathbf{A}$.

$$\mathbf{c}^{(N)} \to \text{constant} \cdot \lambda_1^N \mathbf{e}_{\mathsf{R}}^{(0)}. \tag{17.15}$$

Here, λ_1 is the principal eigenvalue of $\mathbf{A}$.

So to find the capacity of any constrained channel, all we need to do is find the principal eigenvalue, λ_1, of its connection matrix. Then

$$C = \log_2 \lambda_1. \tag{17.16}$$

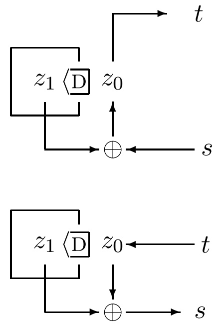


Figure 17.7. An accumulator and a differentiator.

17.4 Back to our model channels

Comparing figure 17.5a and figures 17.5b and c it looks as if channels B and C have the same capacity as channel A. The principal eigenvalues of the three trellises are the same (the eigenvectors for channels A and B are given at the bottom of table C.4, p.608). And indeed the channels are intimately related.

Equivalence of channels A and B

If we take any valid string $\mathbf{s}$ for channel A and pass it through an *accumulator*, obtaining $\mathbf{t}$ defined by:

$$\begin{aligned} t_1 &= s_1 \\ t_n &= t_{n-1} + s_n \bmod 2 \quad \text{for } n \geq 2, \end{aligned} \tag{17.17}$$

then the resulting string is a valid string for channel B, because there are no `11`s in $\mathbf{s}$, so there are no isolated digits in $\mathbf{t}$. The accumulator is an invertible operator, so, similarly, any valid string $\mathbf{t}$ for channel B can be mapped onto a valid string $\mathbf{s}$ for channel A through the *binary differentiator*,

$$\begin{aligned} s_1 &= t_1 \\ s_n &= t_n - t_{n-1} \bmod 2 \quad \text{for } n \geq 2. \end{aligned} \tag{17.18}$$

Because $+$ and $-$ are equivalent in modulo 2 arithmetic, the differentiator is also a blurrer, convolving the source stream with the filter $(1, 1)$.

Channel C is also intimately related to channels A and B.

▷ Exercise 17.7.[1, p.257] What is the relationship of channel C to channels A and B?

17.5 Practical communication over constrained channels

OK, how to do it in practice? Since all three channels are equivalent, we can concentrate on channel A.

Fixed-length solutions

We start with explicitly-enumerated codes. The code in the table 17.8 achieves a rate of $3/5 = 0.6$.

s	$c(s)$
1	00000
2	10000
3	01000
4	00100
5	00010
6	10100
7	01010
8	10010

Table 17.8. A runlength-limited code for channel A.

▷ Exercise 17.8.[1, p.257] Similarly, enumerate all strings of length 8 that end in the zero state. (There are 34 of them.) Hence show that we can map 5 bits (32 source strings) to 8 transmitted bits and achieve rate $5/8 = 0.625$.

What rate can be achieved by mapping an integer number of source bits to $N = 16$ transmitted bits?

Optimal variable-length solution

The optimal way to convey information over the constrained channel is to find the optimal transition probabilities for all points in the trellis, $Q_{s'|s}$, and make transitions with these probabilities.

When discussing channel A, we showed that a sparse source with density $f = 0.38$, driving code C_2, would achieve capacity. And we know how to make sparsifiers (Chapter 6): we design an arithmetic code that is optimal for compressing a sparse source; then its associated decoder gives an optimal mapping from dense (i.e., random binary) strings to sparse strings.

The task of finding the optimal probabilities is given as an exercise.

Exercise 17.9.[*3*] Show that the optimal transition probabilities $\mathbf{Q}$ can be found as follows.

Find the principal right- and left-eigenvectors of $\mathbf{A}$, that is the solutions of $\mathbf{A}\mathbf{e}^{(R)} = \lambda \mathbf{e}^{(R)}$ and $\mathbf{e}^{(L)^{\mathsf{T}}}\mathbf{A} = \lambda \mathbf{e}^{(L)^{\mathsf{T}}}$ with largest eigenvalue λ. Then construct a matrix $\mathbf{Q}$ whose invariant distribution is proportional to $e_i^{(R)} e_i^{(L)}$, namely

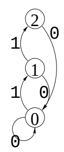
$$Q_{s'|s} = \frac{e_{s'}^{(L)} A_{s's}}{\lambda e_s^{(L)}}. \tag{17.19}$$

[Hint: exercise 16.2 (p.245) might give helpful cross-fertilization here.]

▷ Exercise 17.10.[*3*, p.258] Show that when sequences are generated using the optimal transition probability matrix (17.19), the entropy of the resulting sequence is asymptotically $\log_2 \lambda$ per symbol. [Hint: consider the conditional entropy of just one symbol given the previous one, assuming the previous one's distribution is the invariant distribution.]

In practice, we would probably use finite-precision approximations to the optimal variable-length solution. One might dislike variable-length solutions because of the resulting unpredictability of the actual encoded length in any particular case. Perhaps in some applications we would like a guarantee that the encoded length of a source file of size N bits will be less than a given length such as $N/(C + \epsilon)$. For example, a disk drive is easier to control if all blocks of 512 bytes are known to take exactly the same amount of disk real-estate. For some constrained channels we can make a simple modification to our variable-length encoding and offer such a guarantee, as follows. We find two codes, two mappings of binary strings to variable-length encodings, having the property that for any source string $\mathbf{x}$, if the encoding of $\mathbf{x}$ under the first code is shorter than average, then the encoding of $\mathbf{x}$ under the second code is longer than average, and *vice versa*. Then to transmit a string $\mathbf{x}$ we encode the whole string with both codes and send whichever encoding has the shortest length, prepended by a suitably encoded single bit to convey which of the two codes is being used.

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$



$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

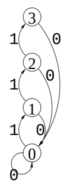


Figure 17.9. State diagrams and connection matrices for channels with maximum runlengths for 1s equal to 2 and 3.

▷ Exercise 17.11.[*3C*, p.258] How many valid sequences of length 8 starting with a 0 are there for the run-length-limited channels shown in figure 17.9?

What are the capacities of these channels?

Using a computer, find the matrices $\mathbf{Q}$ for generating a random path through the trellises of the channel A, and the two run-length-limited channels shown in figure 17.9.

▷ Exercise 17.12.[3, p.258] Consider the run-length-limited channel in which any length of run of 0s is permitted, and the maximum run length of 1s is a large number L such as nine or ninety.

Estimate the capacity of this channel. (Give the first two terms in a series expansion involving L.)

What, roughly, is the form of the optimal matrix $\mathbf{Q}$ for generating a random path through the trellis of this channel? Focus on the values of the elements $Q_{1|0}$, the probability of generating a 1 given a preceding 0, and $Q_{L|L-1}$, the probability of generating a 1 given a preceding run of $L-1$ 1s. Check your answer by explicit computation for the channel in which the maximum runlength of 1s is nine.

▶ 17.6 Variable symbol durations

We can add a further frill to the task of communicating over constrained channels by assuming that the symbols we send have different *durations*, and that our aim is to communicate at the maximum possible rate per unit time. Such channels can come in two flavours: unconstrained, and constrained.

Unconstrained channels with variable symbol durations

We encountered an unconstrained noiseless channel with variable symbol durations in exercise 6.18 (p.125). Solve that problem, and you've done this topic. The task is to determine the optimal frequencies with which the symbols should be used, given their durations.

There is a nice analogy between this task and the task of designing an optimal symbol code (Chapter 4). When we make an binary symbol code for a source with unequal probabilities p_i, the optimal message lengths are $l_i^* = \log_2 1/p_i$, so

$$p_i = 2^{-l_i^*}. \tag{17.20}$$

Similarly, when we have a channel whose symbols have durations l_i (in some units of time), the optimal probability with which those symbols should be used is

$$p_i^* = 2^{-\beta l_i}, \tag{17.21}$$

where β is the capacity of the channel in bits per unit time.

Constrained channels with variable symbol durations

Once you have grasped the preceding topics in this chapter, you should be able to figure out how to define and find the capacity of these, the trickiest constrained channels.

Exercise 17.13.[3] A classic example of a constrained channel with variable symbol durations is the 'Morse' channel, whose symbols are

the dot	d,
the dash	D,
the short space (used between letters in morse code)	s, and
the long space (used between words)	S;

the constraints are that spaces may only be followed by dots and dashes.

Find the capacity of this channel in bits per unit time assuming (a) that all four symbols have equal durations; or (b) that the symbol durations are 2, 4, 3 and 6 time units respectively.

Exercise 17.14.[4] How well-designed is Morse code for English (with, say, the probability distribution of figure 2.1)?

Exercise 17.15.[3C] How difficult is it to get DNA into a narrow tube?

To an information theorist, the entropy associated with a constrained channel reveals how much information can be conveyed over it. In statistical physics, the same calculations are done for a different reason: to predict the thermodynamics of polymers, for example.

As a toy example, consider a polymer of length N that can either sit in a constraining tube, of width L, or in the open where there are no constraints. In the open, the polymer adopts a state drawn at random from the set of one dimensional random walks, with, say, 3 possible directions per step. The entropy of this walk is $\log 3$ per step, i.e., a total of $N \log 3$. [The free energy of the polymer is defined to be $-kT$ times this, where T is the temperature.] In the tube, the polymer's one-dimensional walk can go in 3 directions unless the wall is in the way, so the connection matrix is, for example (if $L = 10$),

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ & & & & \ddots & \ddots & \ddots & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}.$$

Now, what is the entropy of the polymer? What is the *change* in entropy associated with the polymer entering the tube? If possible, obtain an expression as a function of L. Use a computer to find the entropy of the walk for a particular value of L, e.g. 20, and plot the probability density of the polymer's transverse location in the tube.

Notice the difference in capacity between two channels, one constrained and one unconstrained, is directly proportional to the force required to pull the DNA into the tube.

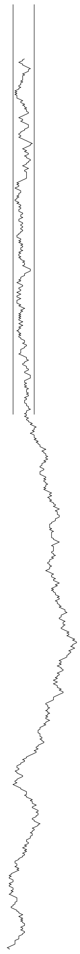


Figure 17.10. Model of DNA squashed in a narrow tube. The DNA will have a tendency to pop out of the tube, because, outside the tube, its random walk has greater entropy.

17.7 Solutions

Solution to exercise 17.5 (p.250). A file transmitted by C_2 contains, on average, one-third 1s and two-thirds 0s.

If $f = 0.38$, the fraction of 1s is $f/(1+f) = (\gamma - 1.0)/(2\gamma - 1.0) = 0.2764$.

Solution to exercise 17.7 (p.254). A valid string for channel C can be obtained from a valid string for channel A by first inverting it [1 $\rightarrow$ 0; 0 $\rightarrow$ 1], then passing it through an accumulator. These operations are invertible, so any valid string for C can also be mapped onto a valid string for A. The only proviso here comes from the edge effects. If we assume that the first character transmitted over channel C is preceded by a string of zeroes, so that the first character is forced to be a 1 (figure 17.5c) then the two channels are exactly equivalent only if we assume that channel A's first character must be a zero.

Solution to exercise 17.8 (p.254). With $N = 16$ transmitted bits, the largest integer number of source bits that can be encoded is 10, so the maximum rate of a fixed length code with $N = 16$ is 0.625.

Solution to exercise 17.10 (p.255). Let the invariant distribution be

$$P(s) = \alpha e_s^{(L)} e_s^{(R)}, \tag{17.22}$$

where α is a normalization constant. The entropy of S_t given S_{t-1}, assuming S_{t-1} comes from the invariant distribution, is

Here, as in Chapter 4, S_t denotes the ensemble whose random variable is the state s_t.

$$H(S_t|S_{t-1}) = -\sum_{s,s'} P(s)P(s'|s)\log P(s'|s) \tag{17.23}$$

$$= -\sum_{s,s'} \alpha e_s^{(L)} e_s^{(R)} \frac{e_{s'}^{(L)} A_{s's}}{\lambda e_s^{(L)}} \log \frac{e_{s'}^{(L)} A_{s's}}{\lambda e_s^{(L)}} \tag{17.24}$$

$$= -\sum_{s,s'} \alpha\, e_s^{(R)} \frac{e_{s'}^{(L)} A_{s's}}{\lambda} \left[\log e_{s'}^{(L)} + \log A_{s's} - \log\lambda - \log e_s^{(L)}\right]. \tag{17.25}$$

Now, $A_{s's}$ is either 0 or 1, so the contributions from the terms proportional to $A_{s's}\log A_{s's}$ are all zero. So

$$H(S_t|S_{t-1}) = \log\lambda + -\frac{\alpha}{\lambda}\sum_{s'}\left(\sum_s A_{s's} e_s^{(R)}\right) e_{s'}^{(L)} \log e_{s'}^{(L)} + \frac{\alpha}{\lambda}\sum_s\left(\sum_{s'} e_{s'}^{(L)} A_{s's}\right) e_s^{(R)} \log e_s^{(L)} \tag{17.26}$$

$$= \log\lambda - \frac{\alpha}{\lambda}\sum_{s'} \lambda e_{s'}^{(R)} e_{s'}^{(L)} \log e_{s'}^{(L)} + \frac{\alpha}{\lambda}\sum_s \lambda e_s^{(L)} e_s^{(R)} \log e_s^{(L)} \tag{17.27}$$

$$= \log\lambda. \tag{17.28}$$

Solution to exercise 17.11 (p.255). The principal eigenvalues of the connection matrices of the two channels are 1.839 and 1.928. The capacities ($\log\lambda$) are 0.879 and 0.947 bits.

Solution to exercise 17.12 (p.256). The channel is similar to the unconstrained binary channel; runs of length greater than L are rare if L is large, so we only expect weak differences from this channel; these differences will show up in contexts where the run length is close to L. The capacity of the channel is very close to one bit.

A lower bound on the capacity is obtained by considering the simple variable-length code for this channel which replaces occurrences of the maximum runlength string `111...1` by `111...10`, and otherwise leaves the source file unchanged. The average rate of this code is $1/(1+2^{-L})$ because the invariant distribution will hit the 'add an extra zero' state a fraction 2^{-L} of the time.

We can reuse the solution for the variable-length channel in exercise 6.18 (p.125). The capacity is the value of β such that the equation

$$Z(\beta) = \sum_{l=1}^{L+1} 2^{-\beta l} = 1 \tag{17.29}$$

is satisfied. The $L+1$ terms in the sum correspond to the $L+1$ possible strings that can be emitted, `0`, `10`, `110`, ... , `11...10`. The sum is exactly given by:

$$Z(\beta) = 2^{-\beta}\frac{\left(2^{-\beta}\right)^{L+1} - 1}{2^{-\beta} - 1}. \tag{17.30}$$

$$\left[\text{Here we used } \sum_{n=0}^{N} ar^n = \frac{a(r^{N+1}-1)}{r-1}.\right]$$

We anticipate that β should be a little less than 1 in order for $Z(\beta)$ to equal 1. Rearranging and solving approximately for β, using $\ln(1+x) \simeq x$,

$$Z(\beta) = 1 \tag{17.31}$$

$$\Rightarrow \beta \simeq 1 - 2^{-(L+2)}/\ln 2. \tag{17.32}$$

We evaluated the true capacities for $L = 2$ and $L = 3$ in an earlier exercise. The table compares the approximate capacity β with the true capacity for a selection of values of L.

L	β	True capacity
2	0.910	0.879
3	0.955	0.947
4	0.977	0.975
5	0.9887	0.9881
6	0.9944	0.9942
9	0.9993	0.9993

The element $Q_{1|0}$ will be close to 1/2 (just a tiny bit larger), since in the unconstrained binary channel $Q_{1|0} = 1/2$. When a run of length $L-1$ has occurred, we effectively have a choice of printing `10` or `0`. Let the probability of selecting `10` be f. Let us estimate the entropy of the *remaining* N characters in the stream as a function of f, assuming the rest of the matrix $\mathbf{Q}$ to have been set to its optimal value. The entropy of the next N characters in the stream is the entropy of the first bit, $H_2(f)$, plus the entropy of the remaining characters, which is roughly $(N-1)$ bits if we select `0` as the first bit and $(N-2)$ bits if `1` is selected. More precisely, if C is the capacity of the channel (which is roughly 1),

$$\begin{aligned} H(\text{the next } N \text{ chars}) &\simeq H_2(f) + [(N-1)(1-f) + (N-2)f]\, C \\ &= H_2(f) + NC - fC \simeq H_2(f) + N - f. \end{aligned} \tag{17.33}$$

Differentiating and setting to zero to find the optimal f, we obtain:

$$\log_2 \frac{1-f}{f} \simeq 1 \;\Rightarrow\; \frac{1-f}{f} \simeq 2 \;\Rightarrow f \simeq 1/3. \tag{17.34}$$

The probability of emitting a `1` thus decreases from about 0.5 to about 1/3 as the number of emitted `1`s increases.

Here is the optimal matrix:

$$\begin{bmatrix} 0 & .3334 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .4287 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & .4669 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & .4841 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & .4923 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & .4963 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & .4983 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .4993 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & .4998 \\ 1 & .6666 & .5713 & .5331 & .5159 & .5077 & .5037 & .5017 & .5007 & .5002 \end{bmatrix}. \tag{17.35}$$

Our rough theory works.