Chapter 2

Background

2.1 Sources, Channels, and Codes

Here we quickly review some basic concepts (covered more precisely in Appendix A) for those unfamiliar with the basic information theoretic formulation of communication problems.

2.1.1 Sources

For our purposes, a source is a random process $\{X_t\}$ evolving in discrete time ($t \in \{0, 1, 2, \ldots\}$) over some alphabet $\mathcal{X}$. Such sources are used as approximations to the real world data generating processes that we are interested in. Our main focus is on scalar valued linear Markov processes.

Definition 2.1.1 *Given a real number A, and real valued i.i.d. random variables $\{W_t\}$ the* scalar discrete-time Markov source with parameter A and noise W *is defined by:*

$$\begin{aligned} X_0 &= 0 \\ X_t &= AX_{t-1} + W_t \end{aligned}$$

This can be expressed without recursion as:

$$X_t = \sum_{i=1}^{t} A^{t-i} W_t$$

There are four distinct regions of the parameter A. If $A = 0$, then the source is white. If $0 < |A| < 1$, then the source is stable since even as $t \to \infty$, the random variables X_t stay well behaved. In particular, if W_t has zero-mean and variance σ^2, then X_t asymptotically has zero-mean and variance $\frac{\sigma^2}{1-A^2} < \infty$.

The other two cases are not so well behaved. If $|A| = 1$, then $\{X_t\}$ is a random walk and is unstable since the uncertainty grows unboundedly in time. It should be clear how our simple random walk falls into this category. Finally if $|A| > 1$, the process is exponentially unstable.

2.1.2 Source Codes

The idea of a source code is to translate a source process into a process taking values in the alphabet of binary strings. It is expressed in two parts: the encoder and the decoder. The encoder maps from the source alphabet $\mathcal{X}$ to the space of binary strings, while the decoder maps from binary strings back into a reconstruction alphabet $\hat{\mathcal{X}}$.

The rate of a source code is the average length of the binary strings generated by the encoder. Because in general it is impossible to have perfect translation at a finite rate, the performance of a source code is evaluated relative to some per-letter distortion function ρ by looking at

$$\sup_{t>0} E[\rho(X_t, \hat{X}_t)]$$

or

$$\lim_{N\to\infty} E[\frac{1}{N}\rho(X_1^N, \hat{X}_1^N)]$$

This is referred to as the distortion of the code. The distortion measure ρ is usually chosen for tractability and as a way to approximate the sense of fidelity we are interested in for the real world situation. Popular distortion measures that we will be interested in are squared error $\rho(X, \hat{X}) = |X - \hat{X}|^2$ and general moments $\rho_\eta(X, \hat{X}) = |X - \hat{X}|^\eta$ where $\eta > 0$.

2.1.3 Noisy Channels

Noisy channels are used to approximate real world communication media that are unreliable or introduce noise. We focus on memoryless channels in this thesis.

Definition 2.1.2 *A* memoryless noisy channel *is a stochastic kernel $P(a|b)$ from an input $a \in \mathcal{A}$ to an output $b \in \mathcal{B}$*

Erasure Channel

A popular discrete-time channel is the erasure channel. It is often used to model channels in which the receiver can detect when an error has occurred.

Definition 2.1.3 *The* binary erasure channel *with erasure probability e is a memoryless noisy channel with $\mathcal{A} = \{0, 1\}$, $\mathcal{B} = \{0, 1, \emptyset\}$ and $P(a = b) = (1 - e)$ and $P(b = \emptyset) = e$.*

AWGN Channel

Another important channel is the Additive White Gaussian Noise (AWGN) channel.

Definition 2.1.4 *A* scalar AWGN channel with variance K_V and power constraint P *is a memoryless noisy channel with $\mathcal{A} = \mathcal{B} = \Re$ and $P(B = a + V|a)$ distributed like a zero mean Gaussian random variable V with variance K_V. There is an additional constraint on the input: $E[A^2] \leq P$*

The power constraint is needed to prevent degenerate solutions. Without it, solutions might end up using arbitrarily large signals that would completely dominate the effect of the additive noise.

2.1.4 Channel Codes

A channel code is an encoder/decoder pair $(\mathcal{E}, \mathcal{D})$ that can be wrapped around a noisy channel. The encoder maps strings of bits $\{S_t\}$ into channel inputs from the set $\mathcal{A}$, while the decoder maps channel outputs from the set $\mathcal{B}$ back into strings of bits $\{\hat{S}_t\}$. The correspondence between input bitstream $\{S_t\}$ and the output bitstream $\{\hat{S}_t\}$ is specified by the reconstruction profile r_1^∞.

The rate of the channel code is the number of bits the encoder takes in per unit time, which is the same as the number of bits the decoder outputs per unit time. While the goal is to reproduce the input bits perfectly at the output (generally with some delay determined by r_1^∞), this is not always possible. To measure the effectiveness, we need to define the probability of error.

Definition 2.1.5 *The* probability of error $P_{error}(\mathcal{E}, \mathcal{D}, r_1^\infty)$ *is the supremum over* $i > 0$ *of the probability of error in the* i*-th bit:*

$$P_{error}(\mathcal{E}, \mathcal{D}, r_1^\infty, i) = P(\hat{S}_i \neq S_i)$$

Block Codes

An important class of channel codes that are traditionally studied are block codes.

Definition 2.1.6 *For non-negative integers* R_{in} *and* R_{out}*, a* (R_{in}, R_{out}) block channel encoder *is a function* $\mathcal{E}$ *from* $\{0,1\}^{R_{in}}$ *into* $\mathcal{A}^{R_{out}}$*. Similarly, a* (R_{in}, R_{out}) block channel decoder *is a function* $\mathcal{D}$ *from* $\mathcal{B}^{R_{out}}$ *into* $\{0,1\}^{R_{in}}$*. The block code has rate* $\frac{R_{in}}{R_{out}}$ *bits per channel use. The range of the encoder is called the set of codewords for the code.*

The maximum end-to-end delay of a block code is approximately the sum of the amount of time it takes to get R_{in} bits plus the amount of time it takes to send out R_{out} channel symbols. This is because the entire block of input bits needs to be ready before we can compute the codeword to be transmitted. Once the codeword has been computed, we must wait another block of time for it to be transmitted to the receiver who can then decode it instantly (ignoring any computational limitations).

2.2 Reliable Communication of Bits

One of the basic motivations behind information theory is to communicate bits reliably through noisy or constrained channels.

2.2.1 "Streaming" and Delay

Before we review existing senses of reliable communication, we need to discuss the role of delay to interpret situations in a "streaming" perspective. In our discrete-time streaming view, time starts at 0 and then goes on forever. Of course, time does not really go on forever for any foreseeable system that we encounter. Even our solar system is expected to come to an end at some point! Still, the infinite horizon view is an important modeling tool that enables us to deal with situations where the time horizon for the ongoing process is far longer than any particular duration or delay that we are interested in. Yet introducing a potential infinity within our model is not without some peril. This infinite domain for time can allow for paradoxical interpretations if we do not require that our codes have some sense of bounded end-to-end delay.

The Hotel Infinity

In [26], Gardner introduces a hypothetical "Hotel Infinity" with rooms labeled by the positive integers. He illustrates the seemingly paradoxical properties of infinite sets by showing how when the hotel has no vacancies, it is still possible to make room for any finite number of incoming guests by just shifting people to larger room numbers. The more fascinating fact is that it is possible to accommodate even an infinite number of guests by asking existing guests to move to a room with double the room number and then accommodating the infinite number of new guests in the odd numbered rooms that become vacant.

The Code Infinity

We can use a similar argument[1] to construct a code for the binary erasure channel which "reliably" transmits at *any rate* regardless of the probability of erasure for the channel and without any feedback.[2] Suppose bits are coming in at a rate R per unit time. Consider an infinite sequence of buffers $\{\mathcal{B}_i\}$ all initially empty. At time l:

1. Place the R bits that just arrived into every buffer
2. Let $j = \lceil \frac{-1+\sqrt{1+8l}}{2} \rceil$ and then $k = \frac{(j+1)^2-j-1}{2} - l + 1$
3. Transmit the oldest bit still waiting in buffer $\mathcal{B}_k$ and remove that bit from $\mathcal{B}_k$. If no bit is waiting, just sent a 0 and ignore it on the receiver.

The receiver knows exactly which bits are still awaiting transmission in each buffer and so can interpret every symbol that it receives. With this strategy, every buffer is visited an infinite number of times since this is just the explicit formula for visiting the buffers in the order $[1, 2, 1, 3, 2, 1, 4, 3, 2, 1, 5, 4, \ldots]$ forever. Each positive integer appears in an infinite number of spots in the list.

Since every buffer is visited an infinite number of times, every incoming bit is sent exactly once across the channel for each buffer. This means that every bit is sent an infinite number of times since there are an infinite number of buffers. Thus the receiver receives an infinite number of samples of every source bit and so with probability 1 can eventually decode every bit correctly![3]

Is it reasonable to interpret this code as achieving reliable transmission no matter how high the erasure probability? That seems like a silly interpretation since at any finite time, only a small fraction of the bits have even had a single opportunity for transmission. In fact, the minimum delay in this scheme increases with the bit position being considered.

We consider the delay of a code to be the supremum over the end-to-end delays for each bit position. It should be clear that to avoid situations like "The Code Infinity" we must require that the delay be bounded in some sense for every bit position in the infinite stream. While we will wish to look at limits as this end-to-end delay goes to infinity, actually infinite delay can make discussions of rate meaningless.

---

[1] Actually, our argument corresponds to a situation where we accomodate an infinite number of infinite families at the Hotel Infinity giving every family member her own room.

[2] This construction is related to Bertrand Russel's Paradox of Tristram Shandy as discussed in [10]. Tristram Shandy was a hypothetical immortal historian who would take one year to transcribe the events of a single day. Although he would fall further and further behind as time went on, paradoxically he would still eventually write a history for all time!

[3] Notice that this argument is not tied to the erasure channel and could work for any channel which is suitably ergodic.

2.2.2 Classical Notions Of Capacity

Noisy channels are traditionally characterized by different measures of capacity: "upto what rate can we transmit data reliably through the channel." The difference between notions of capacity is in terms of what is meant by "reliably." There are two main classical notions of this. Although these are traditionally discussed in terms of block-codes, we will implicitly use Lemma A.5.1 and give the definitions in our terms.

Definition 2.2.1 *Let $\mathcal{R}$ be a* reliable transmission property *which is a Boolean function and either true or false for a given system consisting of a channel encoder $\mathcal{E}$, channel decoder $\mathcal{D}$ (along with a reconstruction profile r_1^∞), and a noisy channel. If it is true, we say that $(\mathcal{E}, \mathcal{D})$ achieve $\mathcal{R}$ for that channel.*

The $\mathcal{R}$-capacity $C_{\mathcal{R}}$ *of a channel is the supremal rate R at which there exist $(\mathcal{E}, \mathcal{D})$ achieving $\mathcal{R}$ for that channel.*

$$C_{\mathcal{R}} = \sup\{R \,|\, \exists(\mathcal{E}, \mathcal{D}, r_1^\infty)\; Rate(\mathcal{E}, \mathcal{D}) = R, \mathcal{R}(\mathcal{E}, \mathcal{D}, r_1^\infty) = 1\}$$

This is an operational definition and is given in terms of the existence of encoders and decoders satisfying the desired property. It tells us what capacity means in terms of getting bits across a channel, not how to calculate its value for any specific channel.

Zero Error Capacity

The first notion of reliable transmission is a very strong one. In words, it says that after some delay T, we know the exact value for the transmitted bit without error. Mathematically:

$$\mathcal{R}_0(\mathcal{E}, \mathcal{D}, r_1^\infty) = \begin{cases} 1 & \text{if } \exists T, \mathrm{Delay}(R, r_1^\infty) \leq T, P_{\mathrm{error}}(\mathcal{E}, \mathcal{D}, r_1^\infty) = 0 \\ 0 & \text{otherwise} \end{cases}$$

The resulting $\mathcal{R}_0$-capacity corresponds to:

Definition 2.2.2 *[61] The* Shannon zero-error capacity C_0 *of a channel is the least upper bound of the rates at which the channel can be used to transmit data with a zero probability of error.*

$$C_0 = \sup\{R \,|\, \exists(T, \mathcal{E}, \mathcal{D}, r_1^\infty)\; P_{error}(\mathcal{E}, \mathcal{D}, r_1^\infty) = 0, Rate(\mathcal{E}, \mathcal{D}) = R, Delay(R, r_1^\infty) \leq T\}$$

Calculating zero-error capacity for even simple channels is in general an open problem usually studied in combinatorics and graph theory. [38]

Shannon Capacity

A weaker notion of reliable transmission allows for some probability of error:

$$\mathcal{R}_\epsilon(\mathcal{E}, \mathcal{D}, r_1^\infty) = \begin{cases} 1 & \text{if } \exists T, \mathrm{Delay}(R, r_1^\infty) \leq T, P_{\mathrm{error}}(\mathcal{E}, \mathcal{D}, r_1^\infty) \leq \epsilon \\ 0 & \text{otherwise} \end{cases}$$

For every $\epsilon > 0$, we have a new notion of $\mathcal{R}_\epsilon$-capacity C_ϵ. If we want the probability of error to be arbitrarily small, it is natural to look at:

$$C = \lim_{\epsilon \to 0} C_\epsilon$$

Shannon's genius was to realize that this need not be the same as the zero error capacity. In fact, as the examples in Section 2.2.3 show, it can be much larger.

Definition 2.2.3 *[23] The* Shannon classical capacity C *of a channel is the least upper bound of the rates at which the channel can be used to transmit data with an arbitrarily small probability of error.*

$$C = \sup\{R \,|\forall \epsilon > 0\, \exists (T, \mathcal{E}, \mathcal{D}, r_1^\infty)\; P_{error}(\mathcal{E}, \mathcal{D}, r_1^\infty) \leq \epsilon, Rate(\mathcal{E}, \mathcal{D}) = R, Delay(R, r_1^\infty) \leq T\}$$

The definition we have given is an operational one. In this definition, just like the one for zero-error capacity, the channel encoder/decoder pair $\mathcal{E}, \mathcal{D}$ is used with a reconstruction profile r_1^∞ that has end-to-end delay less than or equal to T. The delay is shown as a function of only the reconstruction profile and the input rate since the channel's sampling time and offsets are considered to be given. The purpose of the delay is to have time to exploit laws-of-large-numbers. By inspection, we know $C_0 \leq C$.

One of the wonderful things about the Shannon classical capacity C is that it is possible to calculate it as the solution to a particular optimization problem depending only on the properties of the channel. For a memoryless channel, this is the famous maximum mutual information characterization. [23]

$$C = \sup_{P(A)} I(A; B) \tag{2.1}$$

where the random variables $A \in \mathcal{A}$ and $B \in \mathcal{B}$ are linked by the transition probability defining the channel. There has been a lot of work in getting similar characterizations to help us calculate capacity in general cases. [67]

The fact that Shannon classical capacity is computable by (2.1) and other explicit formulae also lets us prove many results about it. For example, it is well-known that the Shannon classical capacity of a memoryless channel is the same regardless of whether the codes are allowed access to feedback.[23]

2.2.3 Example Capacities

Equation (2.1) lets us calculate the capacities for various channels in closed form.

Binary Erasure

For the binary erasure channel with inter-sample time τ, it can be shown that the Shannon classical capacity of this channel is $\frac{1-e}{\tau}$ bits per unit time (or $1 - e$ bits per channel use) regardless of whether the encoder has feedback or not.[23] Furthermore, because a long string of erasures is always possible, the Shannon zero-error capacity of this channel is 0 as long as $e > 0$.

If noiseless feedback is available, the erasure channel has a particularly intuitive encoder which achieves capacity. The encoder maintains a first-in-first-out (FIFO) buffer and retransmits any bit that does not get through. As long as the input rate is sufficiently low, the input buffer will stay finite and every bit will make it through eventually. Formally,

$$\mathcal{E}_i(s_1^{\lfloor \frac{iR}{\tau} \rfloor}, b_1^{i-1}) = s_{\lfloor \frac{iR}{\tau} \rfloor - j_i(b_1^{i-1})} \tag{2.2}$$

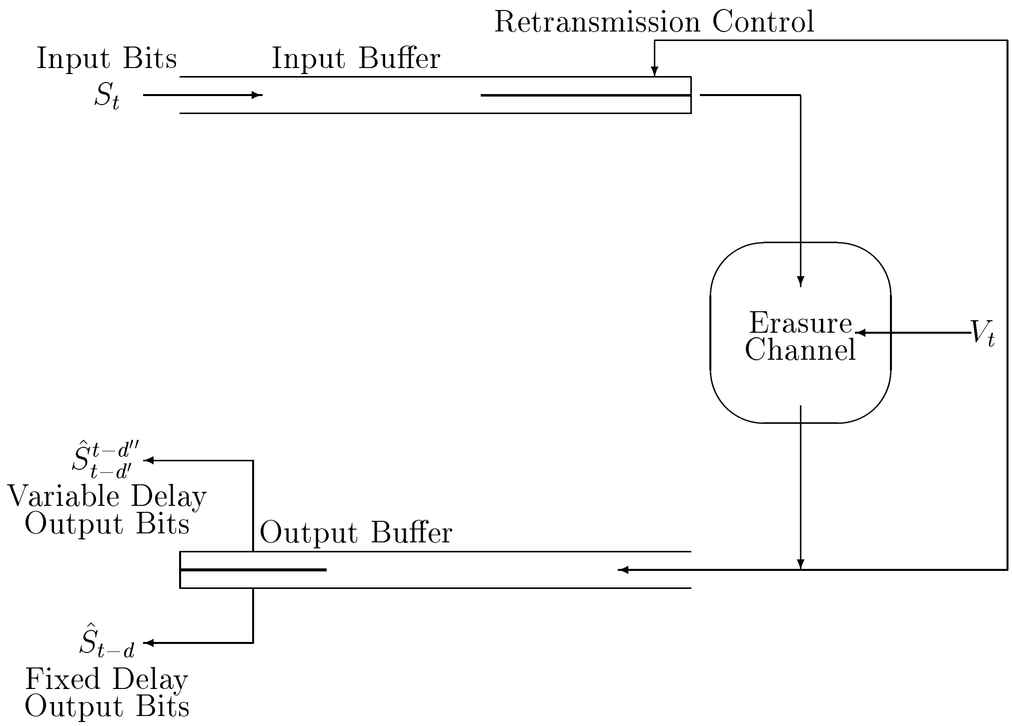

Figure 2-1: Coding for the erasure channel with noiseless feedback

where j is the current buffer size. We can define j recursively as follows: If $\lfloor \frac{iR}{\tau} \rfloor = 0$, then $j_i = 0$. Beyond that:

$$j_i(b_1^{i-1}) = \max(0, j_{i-1}(b_1^{i-2}) + \lfloor \frac{iR}{\tau} \rfloor - \lfloor \frac{(i-1)R}{\tau} \rfloor - 1 + \delta(b_{i-1}, \emptyset)) \tag{2.3}$$

where δ is the usual Kronecker delta function.

So the buffer shrinks by 1 if something useful was received last time and it grows according to the rate at which bits come in. Furthermore, since the buffer size depends only on the time and the received symbols b, the decoder can track it as well. Therefore, the decoder knows which transmission corresponds to which input bit and can give meaningful outputs. As long as the buffer size does not go to infinity, every bit is eventually successfully transmitted and received, even though we cannot predict in advance how much delay will be encountered by any given bit.

Additive White Gaussian Noise Channel

For the scalar AWGN channel with sampling time τ, variance K_V, and power constraint P, it can be shown that the Shannon classical capacity is $\frac{1}{2\tau} \log_2(1 + \frac{P}{K_V})$ bits per unit time regardless of whether the encoder has feedback or not.[23] Traditionally, zero-error capacity has not been evaluated for continuous channels. ([4] is apparently the only exception!) But the same logic applies as before and a string of rare events is always possible, so the Shannon zero-error capacity of this channel is 0 as long as $K_V > 0$.

2.2.4 Trading off Delay and Errors

Our discussion in Section 2.2.1 already tells us that a sense of bounded delay is important for the notion of reliable transmission to be meaningful in the streaming context. In both the zero-error and Shannon capacities, this is satisfied since the codes are required to have a hard bound on delay for every bit position. While in zero-error capacity, the probability of error is required to be exactly 0, the sense of reliable transmission for Shannon capacity allows us to talk about a tradeoff between the required delay and the probability of error.

Error Exponents

Since Shannon capacity C is considered the limit of C_ϵ as $\epsilon \to 0$, it is natural to wonder how $C_\epsilon \to C$. Traditionally, we are interested in more than the tradeoff between rate R and probability of error ϵ. It is clear that to achieve smaller probability of error, we have to tolerate longer delays if we hold the rate constant. This happens whenever the zero error capacity does not equal the Shannon classical capacity. We could just as well define another function $d(\epsilon, R)$ of the rate R and probability of error ϵ that expresses the minimum delay required to achieve that particular (ϵ, R) performance pair.

The traditional tool to characterize this tradeoff between delay and probability of error is the reliability function which gives an "error-exponent" for a given rate. The reliability function is generally defined in terms of block codes.[23]

Definition 2.2.4 *A* reliability function $E(R)$ *for a particular channel is a function which relates the probability of error to the block length as follows:* $\exists \mathcal{E}_0, \mathcal{D}_0, R_{in}, R_{out}$ *such that* $P_{error} \leq 2^{-E(\frac{R_{in}}{R_{out}})R_{out}}$.

Traditionally, this has been considered a measure of code complexity rather than of delay *per se*. Block codes with larger block lengths were considered more complex than ones with short block-lengths. As a result, similar definitions have also been given in terms of the constraint length of convolutional codes.

While reliability functions are certainly very useful in characterizing the tradeoff between delay and probability of error, their tight connection to measures of encoder complexity like block length (R_{out}) is problematic if what we are really interested in is end-to-end delay. Recall that the end-to-end delay for a block channel code is at least $\tau(2R_{out} - 1 - \frac{R_{out}}{R_{in}})$ units of time. This means that in terms of delay, the exponent should be considered half of $E(R)$. Also, the fact that both R_{in} and R_{out} must be integers means that delay cannot even be defined for irrational rates and is in fact very discontinuous as a function of rate.

For example, assume $\tau = 1$. $R = \frac{1}{2} = 0.5$ and $R = \frac{233}{467} = 0.498929\ldots$ are very close to each other as real numbers, but since 467 is prime, any block code for the second rate must have a block length that is an integral multiple of 467 and a delay of at least 932 units! For evaluating performance at moderate values of delay, it matters little that the two error exponents might differ very slightly.

The error exponents given in terms of the constraint length or number of states of a convolutional code are even less illuminating when it comes to evaluating the tradeoff between delay and probability of error. This comes from the fact that a short constraint length or a small number of states can still introduce long-range dependencies in the codewords. As a result, optimum Viterbi decoding of convolutional codes can introduce long delays, even with infinite speed computations. The same is true for sequential decoding. However, Forney has studied the tradeoff of probability of error with respect to delay if we force the decoder to make a decision after a certain delay d. By introducing a way of getting block codes from convolutional ones, he was able to show that the block coding error exponents govern the tradeoff with delay, even though the error exponents relative to the constraint length can be much bigger. [19, 20] Bounds of a similar spirit extending to nonblock codes are also available in [50].

Other Ideas

The main other work relating to trading off delays and errors is that of Krich [42]. This, along with his dissertation [40], discusses weighting the probability of error on every bit by a function $f(d)$ which depends on the delay experienced by that bit. Although he is motivated by the stochastic control problem as we are, he restricts attention throughout his work to block codes. He shows that unlike the unweighted average probability of error case where the best codes have very long block lengths, with many weighting functions the optimal block-lengths are finite to balance the cost of increasing delay for every bit with the benefits of higher accuracy.

Although Krich introduces a weighting function, he never uses it to come up with a new sense of reliable transmission. Instead, he uses it to help guide the choice of block-codes designed to achieve Shannon's sense of reliable transmission. As section 2.4.2 will demonstrate, for unstable processes the Shannon sense of reliable transmission is not good enough and hence Krich's approach of using block-codes will not suffice for us.

2.3 Communicating Sources over Channels

Now, we will attempt to put together the traditional results on communication over noisy channels with the problem of tracking sources. First, we will recall the results for stable Markov sources with $\|A\| < 1$.

For convenience, we will assume that W_t is zero mean. Notice that by our definition, the stable source is tracked by even the trivial zero-rate process $\hat{X}_t = 0$ since X_t has asymptotic variance equal to $\frac{\sigma^2}{1-A^2} < \infty$. But if we are allowed more rate in encoding, we can usually do even better.

2.3.1 Source Coding

Definition 2.3.1 *For a random source $\{X_t\}$, define the operational rate-distortion function $R^{oper}(D)$ by:*

$$\inf \left\{ R \,\middle|\, \forall \epsilon > 0, \exists (n, F_n^R, G_n^R), \lim_{i \to \infty} E\left[\frac{1}{n}\sum_{j=1}^{n} \rho(X_{in+j}, \left(G_n(F_n(X_{in+1}^{(i+1)n}))\right)_j)\right] < D + \epsilon \right\}$$

where (F_n^R, G_n^R) is a block source-encoder and source-decoder pair with Rate$(F_n^R, G_n^R) = R$ and block-length n.

Notice that the operational rate-distortion function above is defined in terms of block-codes. For memoryless processes, Shannon showed that the rate-distortion function can also be computed as the solution to a mutual information optimization problem.[62]

$$R(D) = \inf_{P(\hat{X}|X): E[\rho(X,\hat{X})] \le D} I(X;\hat{X}) \tag{2.4}$$

For processes which are not i.i.d., we can consider them as the limit of larger and larger blocks and thereby get an asymptotic expression for the rate-distortion function as:

$$R(D) = \lim_{N\to\infty} \frac{1}{N} \inf_{P(\hat{X}_1^N|X_1^N):E[\rho^N(X^N,\hat{X}^N)]\leq ND} I(X_1^N;\hat{X}_1^N) \tag{2.5}$$

For stationary ergodic processes and a general class of distortion measures [6], $R^{\text{oper}}(D) = R(D)$. The basic idea is that such processes have fading memories. Thus, disjoint long blocks have roughly the same distribution to each other and the dependencies between the long blocks are very slight. This flavor of result can be extended to some more ill-behaved processes as well [30], though the interpretation in such cases is a little subtle.

2.3.2 The Separation Theorem

The keystone result of Shannon's program is that the restriction to binary for source coding is justified since any rate/distortion performance that can be achieved with nonbinary codes can also be achieved with binary ones. This is called the separation theorem or information transmission theorem:

Theorem 2.3.1 *For a channel, independent stable source $\{X_t\}$, and per-letter distortion measure $\rho(X,\hat{X})$, the performance of any pair of joint source-channel encoder/decoder pairs $\mathcal{E},\mathcal{D}$ is bounded as follows: $E(\rho(X,\hat{X})) \geq R^{-1}(C)$ where $R^{-1}(C)$ is the distortion-rate function (the inverse of the rate-distortion function) evaluated at the Shannon capacity C of the channel. Moreover, we can get arbitrarily close to $R^{-1}(C)$ by using a good source coder followed by a good channel code.*

The traditional proofs rely on the fact that we have characterizations of both Shannon capacity C and rate-distortion function $R(D)$ in terms of mutual information optimization problems. The most sophisticated existing theorems in this regard are those found in [66], but their interpretation is subtle in nonstationary cases. They do not apply within a truly streaming view where the source is realized over time rather than all at once.

2.4 Trying Classical Ideas with Unstable Sources

The stable case in Theorem 2.3.1 seems to contrast sharply with Proposition 1.2.1 and Corollary 1.2.1 which assert the nonexistence of any block codes capable of tracking our simple random walk. However, the existence of the source encoder (1.1) and decoder (1.2) pair which does track our unstable source might give us some hope. Let us see what happens if we try to combine these with codes satisfying the traditional notions of channel capacity and reliable transmission.

2.4.1 Zero-Error Capacity

It is easy to see that if the zero-error capacity $C_0 > 1$, then we can find a delay T and $(\mathcal{E}_z,\mathcal{D}_z)$ such that we can transmit 1 bit without error every time step. Then, by using the source encoder F_t from equation (1.1) to generate the bits $\{S_t\}$, by time t we have $\hat{S}_1^{t-T} = S_1^{t-T}$ available at the output of the channel decoder. By combining this with the the source decoder G from equation (1.2) we can get $\hat{S}_{t-T} = X_{t-T}$. Even if we use this value as our estimate $\hat{X}_t$ for time t itself, we know by the nature of the source that $(X_t - X_{t-T})^2 \leq T^2$. So we can track the source over the channel.

2.4.2 Shannon Classical Capacity

The question is more interesting if $C_0 = 0$ as it is for many well-known channels like the binary symmetric channel, the binary erasure channel, or the power-constrained AWGN channel. Consider a case where $C > 1$. Then, we know by the definition of Shannon classical capacity that we have a delay T and $\mathcal{E}$, $\mathcal{D}$ such that we can transmit 1 bit every time step. If we once again use the source encoder F_t from (1.1) to generate the bits $\{S_t\}$, by time t we have $\hat{S}_1^{t-T}$. We can use the source decoder G from (1.2) to get $\hat{X}_{t-T} = \sum_{i=1}^{t-T}(2\hat{S}_i - 1)$. But we also have a probability of error ϵ and so it makes sense to look at the error: $(X_{t-T} - \hat{X}_{t-T}) = \sum_{i=1}^{t-T} 2(S_i - \hat{S}_i)$.

From the existing theorems about channel capacity, all we know about this process is that $S_i - \hat{S}_i = 0$ with probability at least $1-\epsilon$. To see that this is not enough for $\{\hat{X}_t\}$ to track $\{X_t\}$, consider the possibility that $S_i - \hat{S}_i = \pm 1$ with probability $\frac{\epsilon}{2}$ each and that they are i.i.d. In that case, $E[\rho(X_t, \hat{X}_t)] = \sum_{i=1}^{t} 4E[(S_i - \hat{S}_i)^2] = 4\epsilon t$. Clearly, this has no finite bound and grows to infinity, no matter how small we make ϵ as long as it is not exactly zero.

The problem is that as time goes on, the number of errors made in the past can only grow. Unlike the stable case where the past gradually gets less and less important, for unstable processes the past continues to strongly influence the present. Since the past errors corrupt current estimates, this means that we slowly lose track of where we should be. Figure 2-2 shows what happens in the case where $A > 1$. As soon as a single uncorrected bit error occurs, the estimates diverge exponentially from the true process. The issue here is not that the capacity is too small, but that the very sense of reliable transmission itself is not adequate. Philosophically speaking, this means that the kinds of "bits" that we are using in our source encoders for the simple random walk are somehow incompatible with the Shannon classical capacity.

2.4.3 Binary Erasure Channel With Feedback

If we are using a binary erasure channel with the encoder allowed access to noiseless feedback, then we can use encoder (2.2) to get every bit through eventually. We know that the delay experienced by any given bit is the time it has to wait in the encoder's queue. If τ is the time between channel uses, as long as the incoming rate is less than $\frac{1-e}{\tau}$, the encoder's queue stays stable and its length is random and can be bounded by an exponential distribution. Since the squared-error distortion for our simple random walk is at most T^2 with a delay of T, we know that its expectation will stay finite for all time if the queue is stable. This is true even though the binary erasure channel with feedback has no strict zero error capacity! So zero-error capacity is not necessary, even though it is sufficient.