8

Mathematical Preliminaries for Lossy Coding

8.1 Overview

Before we discussed lossless compression, we presented some of the mathematical background necessary for understanding and appreciating the compression schemes that followed. We will try to do the same here for lossy compression schemes. In lossless compression schemes, rate is the general concern. With lossy compression schemes, the loss of information associated with such schemes is also a concern. We will look at different ways of assessing the impact of the loss of information. We will also briefly revisit the subject of information theory, mainly to get an understanding of the part of the theory that deals with the trade-offs involved in reducing the rate, or number of bits per sample, at the expense of the introduction of distortion in the decoded information. This aspect of information theory is also known as rate distortion theory. We will also look at some of the models used in the development of lossy compression schemes.

8.2 Introduction

This chapter will provide some mathematical background that is necessary for discussing lossy compression techniques. Most of the material covered in this chapter is common to many of the compression techniques described in the later chapters. Material that is specific to a particular technique is described in the chapter in which the technique is presented. Some of the material presented in this chapter is not essential for understanding the techniques described in this book. However, to follow some of the literature in this area, familiarity with these topics is necessary. We have marked these sections with a $\star$. If you are primarily interested in the techniques, you may wish to skip these sections, at least on first reading.

On the other hand, if you wish to delve more deeply into these topics, we have included a list of resources at the end of this chapter that provide a more mathematically rigorous treatment of this material.

When we were looking at lossless compression, one thing we never had to worry about was how the reconstructed sequence would differ from the original sequence. By definition, the reconstruction of a losslessly constructed sequence is identical to the original sequence. However, there is only a limited amount of compression that can be obtained with lossless compression. There is a floor (a hard one) defined by the entropy of the source, below which we cannot drive the size of the compressed sequence. As long as we wish to preserve all of the information in the source, the entropy, like the speed of light, is a fundamental limit.

The limited amount of compression available from using lossless compression schemes may be acceptable in several circumstances. The storage or transmission resources available to us may be sufficient to handle our data requirements after lossless compression. Or the possible consequences of a loss of information may be much more expensive than the cost of additional storage and/or transmission resources. This would be the case with the storage and archiving of bank records; an error in the records could turn out to be much more expensive than the cost of buying additional storage media.

If neither of these conditions hold—that is, resources are limited and we do not require absolute integrity—we can improve the amount of compression by accepting a certain degree of loss during the compression process. Performance measures are necessary to determine the efficiency of our *lossy* compression schemes. For the lossless compression schemes we essentially used only the rate as the performance measure. That would not be feasible for lossy compression. If rate were the only criterion for lossy compression schemes, where loss of information is permitted, the best lossy compression scheme would be simply to throw away all the data! Therefore, we need some additional performance measure, such as some measure of the difference between the original and reconstructed data, which we will refer to as the *distortion* in the reconstructed data. In the next section, we will look at some of the more well-known measures of difference and discuss their advantages and shortcomings.

In the best of all possible worlds we would like to incur the minimum amount of distortion while compressing to the lowest rate possible. Obviously, there is a trade-off between minimizing the rate and keeping the distortion small. The extreme cases are when we transmit no information, in which case the rate is zero, or keep all the information, in which case the distortion is zero. The rate for a discrete source is simply the entropy. The study of the situations between these two extremes is called *rate distortion theory*. In this chapter we will take a brief look at some important concepts related to this theory.

Finally, we need to expand the dictionary of models available for our use, for several reasons. First, because we are now able to introduce distortion, we need to determine how to add distortion intelligently. For this, we often need to look at the sources somewhat differently than we have done previously. Another reason is that we will be looking at compression schemes for sources that are analog in nature, even though we have treated them as discrete sources in the past. We need models that more precisely describe the true nature of these sources. We will describe several different models that are widely used in the development of lossy compression algorithms.

We will use the block diagram and notation used in Figure 8.1 throughout our discussions. The output of the source is modeled as a random variable X. The *source coder*

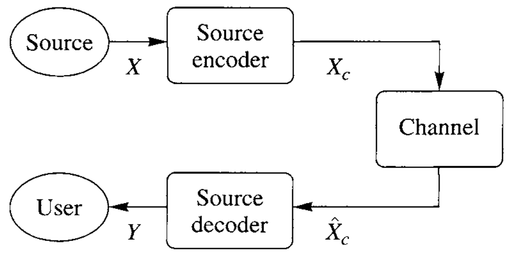

FIGURE 8.1 **Block diagram of a generic compression scheme.**

takes the source output and produces the compressed representation X_c. The channel block represents all transformations the compressed representation undergoes before the source is reconstructed. Usually, we will take the channel to be the identity mapping, which means $X_c = \hat{X}_c$. The source decoder takes the compressed representation and produces a reconstruction of the source output for the user.

8.3 Distortion Criteria

How do we measure the closeness or fidelity of a reconstructed source sequence to the original? The answer frequently depends on what is being compressed and who is doing the answering. Suppose we were to compress and then reconstruct an image. If the image is a work of art and the resulting reconstruction is to be part of a book on art, the best way to find out how much distortion was introduced and in what manner is to ask a person familiar with the work to look at the image and provide an opinion. If the image is that of a house and is to be used in an advertisement, the best way to evaluate the quality of the reconstruction is probably to ask a real estate agent. However, if the image is from a satellite and is to be processed by a machine to obtain information about the objects in the image, the best measure of fidelity is to see how the introduced distortion affects the functioning of the machine. Similarly, if we were to compress and then reconstruct an audio segment, the judgment of how close the reconstructed sequence is to the original depends on the type of material being examined as well as the manner in which the judging is done. An audiophile is much more likely to perceive distortion in the reconstructed sequence, and distortion is much more likely to be noticed in a musical piece than in a politician's speech.

In the best of all worlds we would always use the end user of a particular source output to assess quality and provide the feedback required for the design. In practice this is not often possible, especially when the end user is a human, because it is difficult to incorporate the human response into mathematical design procedures. Also, there is difficulty in objectively reporting the results. The people asked to assess one person's design may be more easygoing than the people who were asked to assess another person's design. Even though the reconstructed output using one person's design is rated "excellent" and the reconstructed output using the other person's design is only rated "acceptable," switching observers may change the ratings. We could reduce this kind of bias by recruiting a large number of observers in the hope that the various biases will cancel each other out. This is often the option used, especially in the final stages of the design of compression systems. However, the rather cumbersome nature of this process is limiting. We generally need a more practical method for looking at how close the reconstructed signal is to the original.

A natural thing to do when looking at the fidelity of a reconstructed sequence is to look at the differences between the original and reconstructed values—in other words, the distortion introduced in the compression process. Two popular measures of distortion or difference between the original and reconstructed sequences are the squared error measure and the absolute difference measure. These are called *difference distortion measures*. If $\{x_n\}$ is the source output and $\{y_n\}$ is the reconstructed sequence, then the squared error measure is given by

$$d(x, y) = (x - y)^2 \tag{8.1}$$

and the absolute difference measure is given by

$$d(x, y) = |x - y|. \tag{8.2}$$

In general, it is difficult to examine the difference on a term-by-term basis. Therefore, a number of average measures are used to summarize the information in the difference sequence. The most often used average measure is the average of the squared error measure. This is called the *mean squared error* (mse) and is often represented by the symbol σ^2 or σ_d^2:

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2. \tag{8.3}$$

If we are interested in the size of the error relative to the signal, we can find the ratio of the average squared value of the source output and the mse. This is called the *signal-to-noise ratio* (SNR).

$$\text{SNR} = \frac{\sigma_x^2}{\sigma_d^2} \tag{8.4}$$

where σ_x^2 is the average squared value of the source output, or signal, and σ_d^2 is the mse. The SNR is often measured on a logarithmic scale and the units of measurement are *decibels* (abbreviated to dB).

$$\text{SNR(dB)} = 10\log_{10}\frac{\sigma_x^2}{\sigma_d^2} \tag{8.5}$$

Sometimes we are more interested in the size of the error relative to the peak value of the signal x_{peak} than with the size of the error relative to the average squared value of the signal. This ratio is called the *peak-signal-to-noise-ratio* (PSNR) and is given by

$$\text{PSNR(dB)} = 10\log_{10}\frac{x_{\text{peak}}^2}{\sigma_d^2}. \tag{8.6}$$

Another difference distortion measure that is used quite often, although not as often as the mse, is the average of the absolute difference, or

$$d_1 = \frac{1}{N}\sum_{n=1}^{N}|x_n - y_n|. \tag{8.7}$$

This measure seems especially useful for evaluating image compression algorithms.

In some applications, the distortion is not perceptible as long as it is below some threshold. In these situations we might be interested in the maximum value of the error magnitude,

$$d_\infty = \max_n |x_n - y_n|. \tag{8.8}$$

We have looked at two approaches to measuring the fidelity of a reconstruction. The first method involving humans may provide a very accurate measure of perceptible fidelity, but it is not practical and not useful in mathematical design approaches. The second is mathematically tractable, but it usually does not provide a very accurate indication of the perceptible fidelity of the reconstruction. A middle ground is to find a mathematical model for human perception, transform both the source output and the reconstruction to this perceptual space, and then measure the difference in the perceptual space. For example, suppose we could find a transformation $\mathcal{V}$ that represented the actions performed by the human visual system (HVS) on the light intensity impinging on the retina before it is "perceived" by the cortex. We could then find $\mathcal{V}(x)$ and $\mathcal{V}(y)$ and examine the difference between them. There are two problems with this approach. First, the process of human perception is very difficult to model, and accurate models of perception are yet to be discovered. Second, even if we could find a mathematical model for perception, the odds are that it would be so complex that it would be mathematically intractable.

In spite of these disheartening prospects, the study of perception mechanisms is still important from the perspective of design and analysis of compression systems. Even if we cannot obtain a transformation that accurately models perception, we can learn something about the properties of perception that may come in handy in the design of compression systems. In the following, we will look at some of the properties of the human visual system and the perception of sound. Our review will be far from thorough, but the intent here is to present some properties that will be useful in later chapters when we talk about compression of images, video, speech, and audio.

8.3.1 The Human Visual System

The eye is a globe-shaped object with a lens in the front that focuses objects onto the retina in the back of the eye. The retina contains two kinds of receptors, called *rods* and *cones*. The rods are more sensitive to light than cones, and in low light most of our vision is due to the operation of rods. There are three kinds of cones, each of which are most sensitive at different wavelengths of the visible spectrum. The peak sensitivities of the cones are in the red, blue, and green regions of the visible spectrum [93]. The cones are mostly concentrated in a very small area of the retina called the *fovea*. Although the rods are more numerous than the cones, the cones provide better resolution because they are more closely packed in the fovea. The muscles of the eye move the eyeball, positioning the image of the object on

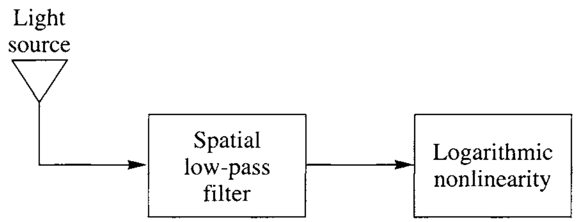

FIGURE 8.2 **A model of monochromatic vision.**

the fovea. This becomes a drawback in low light. One way to improve what you see in low light is to focus to one side of the object. This way the object is imaged on the rods, which are more sensitive to light.

The eye is sensitive to light over an enormously large range of intensities; the upper end of the range is about 10^{10} times the lower end of the range. However, at a given instant we cannot perceive the entire range of brightness. Instead, the eye adapts to an average brightness level. The range of brightness levels that the eye can perceive at any given instant is much smaller than the total range it is capable of perceiving.

If we illuminate a screen with a certain intensity I and shine a spot on it with different intensity, the spot becomes visible when the difference in intensity is ΔI. This is called the *just noticeable difference* (jnd). The ratio $\frac{\Delta I}{I}$ is known as the *Weber fraction* or *Weber ratio*. This ratio is known to be constant at about 0.02 over a wide range of intensities in the absence of background illumination. However, if the background illumination is changed, the range over which the Weber ratio remains constant becomes relatively small. The constant range is centered around the intensity level to which the eye adapts.

If $\frac{\Delta I}{I}$ is constant, then we can infer that the sensitivity of the eye to intensity is a logarithmic function ($d(\log I) = dI/I$). Thus, we can model the eye as a receptor whose output goes to a logarithmic nonlinearity. We also know that the eye acts as a spatial low-pass filter [94, 95]. Putting all of this information together, we can develop a model for monochromatic vision, shown in Figure 8.2.

How does this description of the human visual system relate to coding schemes? Notice that the mind does not perceive everything the eye sees. We can use this knowledge to design compression systems such that the distortion introduced by our lossy compression scheme is not noticeable.

8.3.2 Auditory Perception

The ear is divided into three parts, creatively named the outer ear, the middle ear, and the inner ear. The outer ear consists of the structure that directs the sound waves, or pressure waves, to the *tympanic membrane*, or eardrum. This membrane separates the outer ear from the middle ear. The middle ear is an air-filled cavity containing three small bones that provide coupling between the tympanic membrane and the *oval window*, which leads into the inner ear. The tympanic membrane and the bones convert the pressure waves in the air to acoustical vibrations. The inner ear contains, among other things, a snail-shaped passage called the *cochlea* that contains the transducers that convert the acoustical vibrations to nerve impulses.

The human ear can hear sounds from approximately 20 Hz to 20 kHz, a 1000:1 range of frequencies. The range decreases with age; older people are usually unable to hear the higher frequencies. As in vision, auditory perception has several nonlinear components. One is that loudness is a function not only of the sound level, but also of the frequency. Thus, for example, a pure 1 kHz tone presented at a 20 dB intensity level will have the same apparent loudness as a 50 Hz tone presented at a 50 dB intensity level. By plotting the amplitude of tones at different frequencies that sound equally loud, we get a series of curves called the *Fletcher-Munson curves* [96].

Another very interesting audio phenomenon is that of *masking*, where one sound blocks out or masks the perception of another sound. The fact that one sound can drown out another seems reasonable. What is not so intuitive about masking is that if we were to try to mask a pure tone with noise, only the noise in a small frequency range around the tone being masked contributes to the masking. This range of frequencies is called the *critical band*. For most frequencies, when the noise just masks the tone, the ratio of the power of the tone divided by the power of the noise in the critical band is a constant [97]. The width of the critical band varies with frequency. This fact has led to the modeling of auditory perception as a bank of band-pass filters. There are a number of other, more complicated masking phenomena that also lend support to this theory (see [97, 98] for more information). The limitations of auditory perception play a major role in the design of audio compression algorithms. We will delve further into these limitations when we discuss audio compression in Chapter 16.

8.4 Information Theory Revisited ★

In order to study the trade-offs between rate and the distortion of lossy compression schemes, we would like to have rate defined explicitly as a function of the distortion for a given distortion measure. Unfortunately, this is generally not possible, and we have to go about it in a more roundabout way. Before we head down this path, we need a few more concepts from information theory.

In Chapter 2, when we talked about information, we were referring to letters from a single alphabet. In the case of lossy compression, we have to deal with two alphabets, the source alphabet and the reconstruction alphabet. These two alphabets are generally different from each other.

Example 8.4.1:

A simple lossy compression approach is to drop a certain number of the least significant bits from the source output. We might use such a scheme between a source that generates monochrome images at 8 bits per pixel and a user whose display facility can display only 64 different shades of gray. We could drop the two least significant bits from each pixel before transmitting the image to the user. There are other methods we can use in this situation that are much more effective, but this is certainly simple.

Suppose our source output consists of 4-bit words $\{0, 1, 2, \ldots, 15\}$. The source encoder encodes each value by shifting out the least significant bit. The output alphabet for the source coder is $\{0, 1, 2, \ldots, 7\}$. At the receiver we cannot recover the original value exactly. However, we can get an approximation by shifting in a 0 as the least significant bit, or in other words, multiplying the source encoder output by two. Thus, the reconstruction alphabet is $\{0, 2, 4, \ldots, 14\}$, and the source and reconstruction do not take values from the same alphabet. ♦

As the source and reconstruction alphabets can be distinct, we need to be able to talk about the information relationships between two random variables that take on values from two different alphabets.

8.4.1 Conditional Entropy

Let X be a random variable that takes values from the source alphabet $\mathcal{X} = \{x_0, x_1, \ldots, x_{N-1}\}$. Let Y be a random variable that takes on values from the reconstruction alphabet $\mathcal{Y} = \{y_0, y_1, \ldots, y_{M-1}\}$. From Chapter 2 we know that the entropy of the source and the reconstruction are given by

$$H(X) = -\sum_{i=0}^{N-1} P(x_i) \log_2 P(x_i)$$

and

$$H(Y) = -\sum_{j=0}^{M-1} P(y_j) \log_2 P(y_j).$$

A measure of the relationship between two random variables is the *conditional entropy* (the average value of the conditional self-information). Recall that the self-information for an event A was defined as

$$i(A) = \log \frac{1}{P(A)} = -\log P(A).$$

In a similar manner, the conditional self-information of an event A, given that another event B has occurred, can be defined as

$$i(A|B) = \log \frac{1}{P(A|B)} = -\log P(A|B).$$

Suppose B is the event "Frazer has not drunk anything in two days," and A is the event "Frazer is thirsty." Then $P(A|B)$ should be close to one, which means that the conditional self-information $i(A|B)$ would be close to zero. This makes sense from an intuitive point of view as well. If we know that Frazer has not drunk anything in two days, then the statement that Frazer is thirsty would not be at all surprising to us and would contain very little information.

As in the case of self-information, we are generally interested in the average value of the conditional self-information. This average value is called the conditional entropy. The conditional entropies of the source and reconstruction alphabets are given as

$$H(X|Y) = -\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} P(x_i|y_j) P(y_j) \log_2 P(x_i|y_j) \tag{8.9}$$

and

$$H(Y|X) = -\sum_{i=0}^{N-1}\sum_{j=0}^{M-1} P(x_i|y_j)P(y_j)\log_2 P(y_j|x_i). \tag{8.10}$$

The conditional entropy $H(X|Y)$ can be interpreted as the amount of uncertainty remaining about the random variable X, or the source output, given that we know what value the reconstruction Y took. The additional knowledge of Y should reduce the uncertainty about X, and we can show that

$$H(X|Y) \leq H(X) \tag{8.11}$$

(see Problem 5).

Example 8.4.2:

Suppose we have the 4-bits-per-symbol source and compression scheme described in Example 8.4.1. Assume that the source is equally likely to select any letter from its alphabet. Let us calculate the various entropies for this source and compression scheme.

As the source outputs are all equally likely, $P(X = i) = \frac{1}{16}$ for all $i \in \{0, 1, 2, \ldots, 15\}$, and therefore

$$H(X) = -\sum_i \frac{1}{16}\log\frac{1}{16} = \log 16 = 4 \text{ bits}. \tag{8.12}$$

We can calculate the probabilities of the reconstruction alphabet:

$$P(Y = j) = P(X = j) + P(X = j+1) = \frac{1}{16} + \frac{1}{16} = \frac{1}{8}. \tag{8.13}$$

Therefore, $H(Y) = 3$ bits. To calculate the conditional entropy $H(X|Y)$, we need the conditional probabilities $\{P(x_i|y_j)\}$. From our construction of the source encoder, we see that

$$P(X = i|Y = j) = \begin{cases} \frac{1}{2} & \text{if } i = j \text{ or } i = j+1, \text{ for } j = 0, 2, 4, \ldots, 14 \\ 0 & \text{otherwise.} \end{cases} \tag{8.14}$$

Substituting this in the expression for $H(X|Y)$ in Equation (8.9), we get

$$\begin{aligned} H(X|Y) &= -\sum_i\sum_j P(X = i|Y = j)P(Y = j)\log P(X = i|Y = j) \\ &= -\sum_j [P(X = j|Y = j)P(Y = j)\log P(X = j|Y = j) \\ &\quad + P(X = j+1|Y = j)P(Y = j)\log P(X = j+1|Y = j)] \\ &= -8\left[\frac{1}{2}\cdot\frac{1}{8}\log\frac{1}{2} + \frac{1}{2}\cdot\frac{1}{8}\log\frac{1}{2}\right] && (8.15) \\ &= 1. && (8.16) \end{aligned}$$

Let us compare this answer to what we would have intuitively expected the uncertainty to be, based on our knowledge of the compression scheme. With the coding scheme described here, knowledge of Y means that we know the first 3 bits of the input X. The only thing about the input that we are uncertain about is the value of the last bit. In other words, if we know the value of the reconstruction, our uncertainty about the source output is 1 bit. Therefore, at least in this case, our intuition matches the mathematical definition.

To obtain $H(Y|X)$, we need the conditional probabilities $\{P(y_j|x_i)\}$. From our knowledge of the compression scheme, we see that

$$P(Y=j|X=i) = \begin{cases} 1 & \text{if } i=j \text{ or } i=j+1, \text{ for } j=0,2,4,\ldots,14 \\ 0 & \text{otherwise.} \end{cases} \tag{8.17}$$

If we substitute these values into Equation (8.10), we get $H(Y|X)=0$ bits (note that $0\log 0 = 0$). This also makes sense. For the compression scheme described here, if we know the source output, we know 4 bits, the first 3 of which are the reconstruction. Therefore, in this example, knowledge of the source output at a specific time completely specifies the corresponding reconstruction. ♦

8.4.2 Average Mutual Information

We make use of one more quantity that relates the uncertainty or entropy of two random variables. This quantity is called the *mutual information* and is defined as

$$i(x_k; y_j) = \log\left[\frac{P(x_k|y_j)}{P(x_k)}\right]. \tag{8.18}$$

We will use the average value of this quantity, appropriately called the *average mutual information*, which is given by

$$I(X;Y) = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} P(x_i, y_j)\log\left[\frac{P(x_i|y_j)}{P(x_i)}\right] \tag{8.19}$$

$$= \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} P(x_i|y_j)P(y_j)\log\left[\frac{P(x_i|y_j)}{P(x_i)}\right]. \tag{8.20}$$

We can write the average mutual information in terms of the entropy and the conditional entropy by expanding the argument of the logarithm in Equation (8.20).

$$I(X;Y) = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} P(x_i, y_j)\log\left[\frac{P(x_i|y_j)}{P(x_i)}\right] \tag{8.21}$$

$$= \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} P(x_i, y_j)\log P(x_i|y_j) - \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} P(x_i, y_j)\log P(x_i) \tag{8.22}$$

$$= H(X) - H(X|Y) \tag{8.23}$$

where the second term in Equation (8.22) is $H(X)$, and the first term is $-H(X|Y)$. Thus, the average mutual information is the entropy of the source minus the uncertainty that remains about the source output after the reconstructed value has been received. The average mutual information can also be written as

$$I(X;Y) = H(Y) - H(Y|X) = I(Y;X). \tag{8.24}$$

Example 8.4.3:

For the source coder of Example 8.4.2, $H(X) = 4$ bits, and $H(X|Y) = 1$ bit. Therefore, using Equation (8.23), the average mutual information $I(X;Y)$ is 3 bits. If we wish to use Equation (8.24) to compute $I(X;Y)$, we would need $H(Y)$ and $H(Y|X)$, which from Example 8.4.2 are 3 and 0, respectively. Thus, the value of $I(X;Y)$ still works out to be 3 bits. ♦

8.4.3 Differential Entropy

Up to this point we have assumed that the source picks its outputs from a discrete alphabet. When we study lossy compression techniques, we will see that for many sources of interest to us this assumption is not true. In this section, we will extend some of the information theoretic concepts defined for discrete random variables to the case of random variables with continuous distributions.

Unfortunately, we run into trouble from the very beginning. Recall that the first quantity we defined was self-information, which was given by $\log \frac{1}{P(x_i)}$, where $P(x_i)$ is the probability that the random variable will take on the value x_i. For a random variable with a continuous distribution, this probability is zero. Therefore, if the random variable has a continuous distribution, the "self-information" associated with any value is infinity.

If we do not have the concept of self-information, how do we go about defining entropy, which is the average value of the self-information? We know that many continuous functions can be written as limiting cases of their discretized version. We will try to take this route in order to define the entropy of a continuous random variable X with probability density function *(pdf)* $f_X(x)$.

While the random variable X cannot generally take on a particular value with nonzero probability, it can take on a value in an *interval* with nonzero probability. Therefore, let us divide the range of the random variable into intervals of size Δ. Then, by the mean value theorem, in each interval $[(i-1)\Delta, i\Delta)$, there exists a number x_i, such that

$$f_X(x_i)\Delta = \int_{(i-1)\Delta}^{i\Delta} f_X(x)\,dx. \tag{8.25}$$

Let us define a discrete random variable X_d with *pdf*

$$P(X_d = x_i) = f_X(x_i)\Delta. \tag{8.26}$$

Then we can obtain the entropy of this random variable as

$$H(X_d) = -\sum_{i=-\infty}^{\infty} P(x_i)\log P(x_i) \tag{8.27}$$

$$= -\sum_{i=-\infty}^{\infty} f_X(x_i)\Delta \log f_X(x_i)\Delta \tag{8.28}$$

$$= -\sum_{i=-\infty}^{\infty} f_X(x_i)\Delta \log f_X(x_i) - \sum_{i=-\infty}^{\infty} f_X(x_i)\Delta \log \Delta \tag{8.29}$$

$$= -\sum_{i=-\infty}^{\infty} [f_X(x_i)\log f_X(x_i)]\,\Delta - \log \Delta. \tag{8.30}$$

Taking the limit as $\Delta \to 0$ of Equation (8.30), the first term goes to $-\int_{-\infty}^{\infty} f_X(x)\log f_X(x)\,dx$, which looks like the analog to our definition of entropy for discrete sources. However, the second term is $-\log \Delta$, which goes to plus infinity when Δ goes to zero. It seems there is not an analog to entropy as defined for discrete sources. However, the first term in the limit serves some functions similar to that served by entropy in the discrete case and is a useful function in its own right. We call this term the *differential entropy* of a continuous source and denote it by $h(X)$.

Example 8.4.4:

Suppose we have a random variable X that is uniformly distributed in the interval $[a, b)$. The differential entropy of this random variable is given by

$$h(X) = -\int_{-\infty}^{\infty} f_X(x)\log f_X(x)\,dx \tag{8.31}$$

$$= -\int_{a}^{b} \frac{1}{b-a}\log\frac{1}{b-a}\,dx \tag{8.32}$$

$$= \log(b-a). \tag{8.33}$$

Notice that when $b-a$ is less than one, the differential entropy will become negative—in contrast to the entropy, which never takes on negative values. ♦

Later in this chapter, we will find particular use for the differential entropy of the Gaussian source.

Example 8.4.5:

Suppose we have a random variable X that has a Gaussian *pdf*,

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp -\frac{(x-\mu)^2}{2\sigma^2}. \tag{8.34}$$

The differential entropy is given by

$$h(X) = -\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi\sigma^2}}\exp -\frac{(x-\mu)^2}{2\sigma^2}\log\left[\frac{1}{\sqrt{2\pi\sigma^2}}\exp -\frac{(x-\mu)^2}{2\sigma^2}\right]dx \tag{8.35}$$

$$= -\log\frac{1}{\sqrt{2\pi\sigma^2}}\int_{-\infty}^{\infty} f_X(x)dx + \int_{-\infty}^{\infty} \frac{(x-\mu)^2}{2\sigma^2}\log e f_X(x)dx \tag{8.36}$$

$$= \frac{1}{2}\log 2\pi\sigma^2 + \frac{1}{2}\log e \tag{8.37}$$

$$= \frac{1}{2}\log 2\pi e\sigma^2. \tag{8.38}$$

Thus, the differential entropy of a Gaussian random variable is directly proportional to its variance. ♦

The differential entropy for the Gaussian distribution has the added distinction that it is larger than the differential entropy for any other continuously distributed random variable with the same variance. That is, for any random variable X, with variance σ^2

$$h(X) \leq \frac{1}{2}\log 2\pi e\sigma^2. \tag{8.39}$$

The proof of this statement depends on the fact that for any two continuous distributions $f_X(X)$ and $g_X(X)$

$$-\int_{-\infty}^{\infty} f_X(x)\log f_X(x)dx \leq -\int_{-\infty}^{\infty} f_X(x)\log g_X(x)dx. \tag{8.40}$$

We will not prove Equation (8.40) here, but you may refer to [99] for a simple proof. To obtain Equation (8.39), we substitute the expression for the Gaussian distribution for $g_X(x)$. Noting that the left-hand side of Equation (8.40) is simply the differential entropy of the random variable X, we have

$$\begin{aligned} h(X) &\leq -\int_{-\infty}^{\infty} f_X(x)\log \frac{1}{\sqrt{2\pi\sigma^2}}\exp -\frac{(x-\mu)^2}{2\sigma^2}dx \\ &= \frac{1}{2}\log(2\pi\sigma^2) + \log e\int_{-\infty}^{\infty} f_X(x)\frac{(x-\mu)^2}{2\sigma^2}dx \\ &= \frac{1}{2}\log(2\pi\sigma^2) + \frac{\log e}{2\sigma^2}\int_{-\infty}^{\infty} f_X(x)(x-\mu)^2 dx \\ &= \frac{1}{2}\log(2\pi e\sigma^2). \end{aligned} \tag{8.41}$$

We seem to be striking out with continuous random variables. There is no analog for self-information and really none for entropy either. However, the situation improves when we look for an analog for the average mutual information. Let us define the random variable Y_d in a manner similar to the random variable X_d, as the discretized version of a continuous valued random variable Y. Then we can show (see Problem 4)

$$H(X_d|Y_d) = -\sum_{i=-\infty}^{\infty}\sum_{j=-\infty}^{\infty}\left[f_{X|Y}(x_i|y_j)f_Y(y_j)\log f_{X|Y}(x_i|y_j)\right]\Delta\Delta - \log\Delta. \tag{8.42}$$

Therefore, the average mutual information for the discretized random variables is given by

$$I(X_d; Y_d) = H(X_d) - H(X_d|Y_d) \tag{8.43}$$

$$= -\sum_{i=-\infty}^{\infty} f_X(x_i)\Delta \log f_X(x_i) \tag{8.44}$$

$$-\sum_{i=-\infty}^{\infty}\left[\sum_{j=-\infty}^{\infty} f_{X|Y}(x_i|y_j)f_Y(y_j)\log f_{X|Y}(x_i|y_j)\Delta\right]\Delta. \tag{8.45}$$

Notice that the two $\log \Delta$s in the expression for $H(X_d)$ and $H(X_d|Y_d)$ cancel each other out, and as long as $h(X)$ and $h(X|Y)$ are not equal to infinity, when we take the limit as $\Delta \to 0$ of $I(X_d; Y_d)$ we get

$$I(X; Y) = h(X) - h(X|Y). \tag{8.46}$$

The average mutual information in the continuous case can be obtained as a limiting case of the average mutual information for the discrete case and has the same physical significance.

We have gone through a lot of mathematics in this section. But the information will be used immediately to define the rate distortion function for a random source.

8.5 Rate Distortion Theory ★

Rate distortion theory is concerned with the trade-offs between distortion and rate in lossy compression schemes. Rate is defined as the average number of bits used to represent each sample value. One way of representing the trade-offs is via a *rate distortion function R(D)*. The rate distortion function $R(D)$ specifies the lowest rate at which the output of a source can be encoded while keeping the distortion less than or equal to D. On our way to mathematically defining the rate distortion function, let us look at the rate and distortion for some different lossy compression schemes.

In Example 8.4.2, knowledge of the value of the input at time k completely specifies the reconstructed value at time k. In this situation,

$$P(y_j|x_i) = \begin{cases} 1 & \text{for some } j = j_i \\ 0 & \text{otherwise.} \end{cases} \tag{8.47}$$

Therefore,

$$D = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} P(y_j|x_i)P(x_i)d(x_i, y_j) \tag{8.48}$$

$$= \sum_{i=0}^{N-1} P(x_i)d(x_i, y_{j_i}) \tag{8.49}$$

where we used the fact that $P(x_i, y_j) = P(y_j|x_i)P(x_i)$ in Equation (8.48). The rate for this source coder is the output entropy $H(Y)$ of the source decoder. If this were always the case, the task of obtaining a rate distortion function would be relatively simple. Given a distortion constraint D^*, we could look at all encoders with distortion less than D^* and pick the one with the lowest output entropy. This entropy would be the rate corresponding to the distortion D^*. However, the requirement that knowledge of the input at time k completely specifies the reconstruction at time k is very restrictive, and there are many efficient compression techniques that would have to be excluded under this requirement. Consider the following example.

Example 8.5.1:

With a data sequence that consists of height and weight measurements, obviously height and weight are quite heavily correlated. In fact, after studying a long sequence of data, we find that if we plot the height along the x axis and the weight along the y axis, the data points cluster along the line $y = 2.5x$. In order to take advantage of this correlation, we devise the following compression scheme. For a given pair of height and weight measurements, we find the orthogonal projection on the $y = 2.5x$ line as shown in Figure 8.3. The point on this line can be represented as the distance to the nearest integer from the origin. Thus, we encode a pair of values into a single value. At the time of reconstruction, we simply map this value back into a pair of height and weight measurements.

For instance, suppose somebody is 72 inches tall and weighs 200 pounds (point A in Figure 8.3). This corresponds to a point at a distance of 212 along the $y = 2.5x$ line. The reconstructed values of the height and weight corresponding to this value are 79 and 197. Notice that the reconstructed values differ from the original values. Suppose we now have

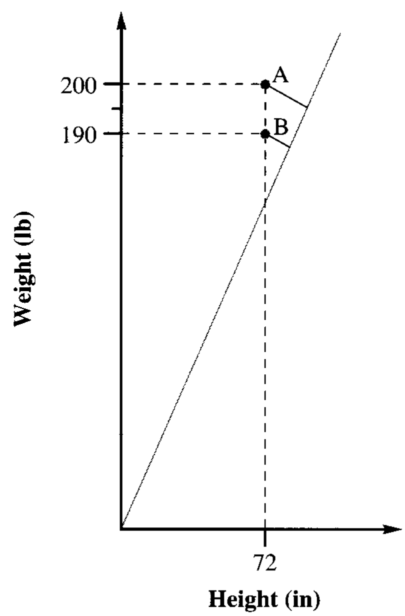

FIGURE 8.3 **Compression scheme for encoding height-weight pairs.**

another individual who is also 72 inches tall but weighs 190 pounds (point B in Figure 8.3). The source coder output for this pair would be 203, and the reconstructed values for height and weight are 75 and 188, respectively. Notice that while the height value in both cases was the same, the reconstructed value is different. The reason for this is that the reconstructed value for the height depends on the weight. Thus, for this particular source coder, we do not have a conditional probability density function $\{P(y_j|x_i)\}$ of the form shown in Equation (8.47). ♦

Let us examine the distortion for this scheme a little more closely. As the conditional probability for this scheme is not of the form of Equation (8.47), we can no longer write the distortion in the form of Equation (8.49). Recall that the general form of the distortion is

$$D = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} d(x_i, y_j) P(x_i) P(y_j|x_i). \tag{8.50}$$

Each term in the summation consists of three factors: the distortion measure $d(x_i, y_j)$, the source density $P(x_i)$, and the conditional probability $P(y_j|x_i)$. The distortion measure is a measure of closeness of the original and reconstructed versions of the signal and is generally determined by the particular application. The source probabilities are solely determined by the source. The third factor, the set of conditional probabilities, can be seen as a description of the compression scheme.

Therefore, for a given source with some *pdf* $\{P(x_i)\}$ and a specified distortion measure $d(\cdot, \cdot)$, the distortion is a function only of the conditional probabilities $\{P(y_j|x_i)\}$; that is,

$$D = D(\{P(y_j|x_i)\}). \tag{8.51}$$

Therefore, we can write the constraint that the distortion D be less than some value D^* as a requirement that the conditional probabilities for the compression scheme belong to a set of conditional probabilities Γ that have the property that

$$\Gamma = \{\{P(y_j|x_i\} \text{ such that } D(\{P(y_j|x_i)\}) \leq D^*\}. \tag{8.52}$$

Once we know the set of compression schemes to which we have to confine ourselves, we can start to look at the rate of these schemes. In Example 8.4.2, the rate was the entropy of Y. However, that was a result of the fact that the conditional probability describing that particular source coder took on only the values 0 and 1. Consider the following trivial situation.

Example 8.5.2:

Suppose we have the same source as in Example 8.4.2 and the same reconstruction alphabet. Suppose the distortion measure is

$$d(x_i, y_j) = (x_i - y_j)^2$$

and $D^* = 225$. One compression scheme that satisfies the distortion constraint randomly maps the input to any one of the outputs; that is,

$$P(y_j|x_i) = \frac{1}{8} \qquad \text{for } i = 0, 1, \ldots, 15 \text{ and } j = 0, 2, \ldots, 14.$$

We can see that this conditional probability assignment satisfies the distortion constraint. As each of the eight reconstruction values is equally likely, $H(Y)$ is 3 bits. However, we are not transmitting *any* information. We could get exactly the same results by transmitting 0 bits and randomly picking Y at the receiver. ♦

Therefore, the entropy of the reconstruction $H(Y)$ cannot be a measure of the rate. In his 1959 paper on source coding [100], Shannon showed that the minimum rate for a given distortion is given by

$$R(D) = \min_{\{P(y_j|x_i)\}\in\Gamma} I(X; Y). \tag{8.53}$$

To prove this is beyond the scope of this book. (Further information can be found in [3] and [4].) However, we can at least convince ourselves that defining the rate as an average mutual information gives sensible answers when used for the examples shown here. Consider Example 8.4.2. The average mutual information in this case is 3 bits, which is what we said the rate was. In fact, notice that whenever the conditional probabilities are constrained to be of the form of Equation (8.47),

$$H(Y|X) = 0,$$

then

$$I(X; Y) = H(Y),$$

which had been our measure of rate.

In Example 8.5.2, the average mutual information is 0 bits, which accords with our intuitive feeling of what the rate should be. Again, whenever

$$H(Y|X) = H(Y),$$

that is, knowledge of the source gives us no knowledge of the reconstruction,

$$I(X; Y) = 0,$$

which seems entirely reasonable. We should not have to transmit any bits when we are not sending any information.

At least for the examples here, it seems that the average mutual information does represent the rate. However, earlier we had said that the average mutual information between the source output and the reconstruction is a measure of the information conveyed by the reconstruction about the source output. Why are we then looking for compression schemes that *minimize* this value? To understand this, we have to remember that the process of finding the performance of the optimum compression scheme had two parts. In the first part we specified the desired distortion. The entire set of conditional probabilities over which the average mutual information is minimized satisfies the distortion constraint. Therefore, we can leave the question of distortion, or fidelity, aside and concentrate on minimizing the rate.

Finally, how do we find the rate distortion function? There are two ways: one is a computational approach developed by Arimoto [101] and Blahut [102]. While the derivation of the algorithm is beyond the scope of this book, the algorithm itself is relatively simple. The other approach is to find a lower bound for the average mutual information and then show that we can achieve this bound. We use this approach to find the rate distortion functions for two important sources.

Example 8.5.3: Rate distortion function for the binary source

Suppose we have a source alphabet $\{0, 1\}$, with $P(0) = p$. The reconstruction alphabet is also binary. Given the distortion measure

$$d(x_i, y_j) = x_i \oplus y_j, \tag{8.54}$$

where $\oplus$ is modulo 2 addition, let us find the rate distortion function. Assume for the moment that $p < \frac{1}{2}$. For $D > p$ an encoding scheme that would satisfy the distortion criterion would be not to transmit anything and fix $Y = 1$. So for $D \geq p$

$$R(D) = 0. \tag{8.55}$$

We will find the rate distortion function for the distortion range $0 \leq D < p$.

Find a lower bound for the average mutual information:

$$\begin{aligned} I(X; Y) &= H(X) - H(X|Y) && (8.56)\\ &= H(X) - H(X \oplus Y|Y) && (8.57)\\ &\geq H(X) - H(X \oplus Y) \qquad \text{from Equation (8.11).} && (8.58) \end{aligned}$$

In the second step we have used the fact that if we know Y, then knowing X we can obtain $X \oplus Y$ and vice versa as $X \oplus Y \oplus Y = X$.

Let us look at the terms on the right-hand side of (8.11):

$$H(X) = -p \log_2 p - (1-p) \log_2(1-p) = H_b(p), \tag{8.59}$$

where $H_b(p)$ is called the *binary entropy function* and is plotted in Figure 8.4. Note that $H_b(p) = H_b(1-p)$.

Given that $H(X)$ is completely specified by the source probabilities, our task now is to find the conditional probabilities $\{P(x_i|y_j)\}$ such that $H(X \oplus Y)$ is maximized while the average distortion $E[d(x_i, y_j)] \leq D$. $H(X \oplus Y)$ is simply the binary entropy function $H_b(P(X \oplus Y = 1))$, where

$$P(X \oplus Y = 1) = P(X = 0, Y = 1) + P(X = 1, Y = 0). \tag{8.60}$$

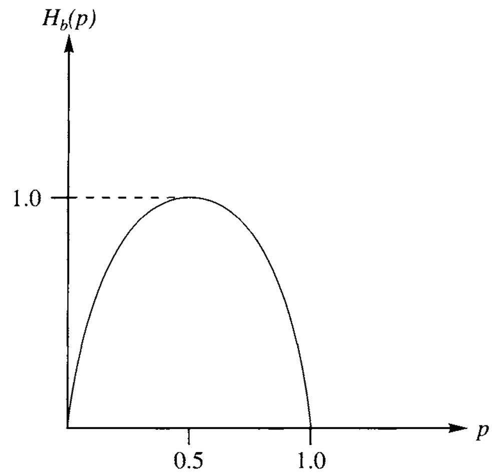

FIGURE 8. 4 The binary entropy function.

Therefore, to maximize $H(X \oplus Y)$, we would want $P(X \oplus Y = 1)$ to be as close as possible to one-half. However, the selection of $P(X \oplus Y)$ also has to satisfy the distortion constraint. The distortion is given by

$$\begin{aligned} E[d(x_i, y_j)] &= 0 \times P(X=0, Y=0) + 1 \times P(X=0, Y=1) \\ &\quad + 1 \times P(X=1, Y=0) + 0 \times P(X=1, Y=1) \\ &= P(X=0, Y=1) + P(X=1, Y=0) \\ &= P(Y=1|X=0)p + P(Y=0|X=1)(1-p). \end{aligned} \tag{8.61}$$

But this is simply the probability that $X \oplus Y = 1$. Therefore, the maximum value that $P(X \oplus Y = 1)$ can have is D. Our assumptions were that $D < p$ and $p \leq \frac{1}{2}$, which means that $D < \frac{1}{2}$. Therefore, $P(X \oplus Y = 1)$ is closest to $\frac{1}{2}$ while being less than or equal to D when $P(X \oplus Y = 1) = D$. Therefore,

$$I(X; Y) \geq H_b(p) - H_b(D). \tag{8.62}$$

We can show that for $P(X=0|Y=1) = P(X=1|Y=0) = D$, this bound is achieved. That is, if $P(X=0|Y=1) = P(X=1|Y=0) = D$, then

$$I(X; Y) = H_b(p) - H_b(D). \tag{8.63}$$

Therefore, for $D < p$ and $p \leq \frac{1}{2}$,

$$R(D) = H_b(p) - H_b(D). \tag{8.64}$$

Finally, if $p > \frac{1}{2}$, then we simply switch the roles of p and $1-p$. Putting all this together, the rate distortion function for a binary source is

$$R(D) = \begin{cases} H_b(p) - H_b(D) & \text{for } D < \min\{p, 1-p\} \\ 0 & \text{otherwise.} \end{cases} \tag{8.65}$$

♦

Example 8.5.4: Rate distortion function for the Gaussian source

Suppose we have a continuous amplitude source that has a zero mean Gaussian *pdf* with variance σ^2. If our distortion measure is given by

$$d(x, y) = (x-y)^2, \tag{8.66}$$

our distortion constraint is given by

$$E\left[(X-Y)^2\right] \leq D. \tag{8.67}$$

Our approach to finding the rate distortion function will be the same as in the previous example; that is, find a lower bound for $I(X; Y)$ given a distortion constraint, and then show that this lower bound can be achieved.

First we find the rate distortion function for $D < \sigma^2$.

$$\begin{aligned} I(X; Y) &= h(X) - h(X|Y) &(8.68)\\ &= h(X) - h(X-Y|Y) &(8.69)\\ &\geq h(X) - h(X-Y) &(8.70) \end{aligned}$$

In order to minimize the right-hand side of Equation (8.70), we have to maximize the second term subject to the constraint given by Equation (8.67). This term is maximized if $X-Y$ is Gaussian, and the constraint can be satisfied if $E\left[(X-Y)^2\right] = D$. Therefore, $h(X-Y)$ is the differential entropy of a Gaussian random variable with variance D, and the lower bound becomes

$$\begin{aligned} I(X; Y) &\geq \frac{1}{2}\log(2\pi e\sigma^2) - \frac{1}{2}\log(2\pi e D) &(8.71)\\ &= \frac{1}{2}\log\frac{\sigma^2}{D}. &(8.72) \end{aligned}$$

This average mutual information can be achieved if Y is zero mean Gaussian with variance $\sigma^2 - D$, and

$$f_{X|Y}(x|y) = \frac{1}{\sqrt{2\pi D}}\exp\frac{-x^2}{2D}. \tag{8.73}$$

For $D > \sigma^2$, if we set $Y = 0$, then

$$I(X; Y) = 0 \tag{8.74}$$

and

$$E\left[(X-Y)^2\right]=\sigma^2<D. \tag{8.75}$$

Therefore, the rate distortion function for the Gaussian source can be written as

$$R(D)=\begin{cases}\frac{1}{2}\log\frac{\sigma^2}{D} & \text{for } D<\sigma^2\\ 0 & \text{for } D>\sigma^2.\end{cases} \tag{8.76}$$

♦

Like the differential entropy for the Gaussian source, the rate distortion function for the Gaussian source also has the distinction of being larger than the rate distortion function for any other source with a continuous distribution and the same variance. This is especially valuable because for many sources it can be very difficult to calculate the rate distortion function. In these situations, it is helpful to have an upper bound for the rate distortion function. It would be very nice if we also had a lower bound for the rate distortion function of a continuous random variable. Shannon described such a bound in his 1948 paper [7], and it is appropriately called the *Shannon lower bound*. We will simply state the bound here without derivation (for more information, see [4]).

The Shannon lower bound for a random variable X and the magnitude error criterion

$$d(x,y)=|x-y| \tag{8.77}$$

is given by

$$R_{SLB}(D)=h(X)-\log(2eD). \tag{8.78}$$

If we used the squared error criterion, the Shannon lower bound is given by

$$R_{SLB}(D)=h(X)-\frac{1}{2}\log(2\pi eD). \tag{8.79}$$

In this section we have defined the rate distortion function and obtained the rate distortion function for two important sources. We have also obtained upper and lower bounds on the rate distortion function for an arbitrary *iid* source. These functions and bounds are especially useful when we want to know if it is possible to design compression schemes to provide a specified rate and distortion given a particular source. They are also useful in determining the amount of performance improvement that we could obtain by designing a better compression scheme. In these ways the rate distortion function plays the same role for lossy compression that entropy plays for lossless compression.

8.6 Models

As in the case of lossless compression, models play an important role in the design of lossy compression algorithms; there are a variety of approaches available. The set of models we can draw on for lossy compression is much wider than the set of models we studied for lossless compression. We will look at some of these models in this section. What is presented here is by no means an exhaustive list of models. Our only intent is to describe those models that will be useful in the following chapters.

8.6.1 Probability Models

An important method for characterizing a particular source is through the use of probability models. As we shall see later, knowledge of the probability model is important for the design of a number of compression schemes.

Probability models used for the design and analysis of lossy compression schemes differ from those used in the design and analysis of lossless compression schemes. When developing models in the lossless case, we tried for an exact match. The probability of each symbol was estimated as part of the modeling process. When modeling sources in order to design or analyze lossy compression schemes, we look more to the general rather than exact correspondence. The reasons are more pragmatic than theoretical. Certain probability distribution functions are more analytically tractable than others, and we try to match the distribution of the source with one of these "nice" distributions.

Uniform, Gaussian, Laplacian, and Gamma distribution are four probability models commonly used in the design and analysis of lossy compression systems:

- **Uniform Distribution:** As for lossless compression, this is again our ignorance model. If we do not know anything about the distribution of the source output, except possibly the range of values, we can use the uniform distribution to model the source. The probability density function for a random variable uniformly distributed between a and b is

$$f_X(x) = \begin{cases} \frac{1}{b-a} & \text{for } a \leq x \leq b \\ 0 & \text{otherwise.} \end{cases} \tag{8.80}$$

- **Gaussian Distribution:** The Gaussian distribution is one of the most commonly used probability models for two reasons: it is mathematically tractable and, by virtue of the central limit theorem, it can be argued that in the limit the distribution of interest goes to a Gaussian distribution. The probability density function for a random variable with a Gaussian distribution and mean μ and variance σ^2 is

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp - \frac{(x-\mu)^2}{2\sigma^2}. \tag{8.81}$$

- **Laplacian Distribution:** Many sources that we deal with have distributions that are quite peaked at zero. For example, speech consists mainly of silence. Therefore, samples of speech will be zero or close to zero with high probability. Image pixels themselves do not have any attraction to small values. However, there is a high degree of correlation among pixels. Therefore, a large number of the pixel-to-pixel differences will have values close to zero. In these situations, a Gaussian distribution is not a very close match to the data. A closer match is the Laplacian distribution, which is peaked at zero. The distribution function for a zero mean random variable with Laplacian distribution and variance σ^2 is

$$f_X(x) = \frac{1}{\sqrt{2\sigma^2}} \exp \frac{-\sqrt{2}\,|x|}{\sigma}. \tag{8.82}$$

- **Gamma Distribution:** A distribution that is even more peaked, though considerably less tractable, than the Laplacian distribution is the Gamma distribution. The distribution function for a Gamma distributed random variable with zero mean and variance σ^2 is given by

$$f_X(x) = \frac{\sqrt[4]{3}}{\sqrt{8\pi\sigma\,|x|}} \exp \frac{-\sqrt{3}\,|x|}{2\sigma}. \tag{8.83}$$

The shapes of these four distributions, assuming a mean of zero and a variance of one, are shown in Figure 8.5.

One way of obtaining the estimate of the distribution of a particular source is to divide the range of outputs into "bins" or intervals I_k. We can then find the number of values n_k that fall into each interval. A plot of $\frac{n_k}{n_T}$, where n_T is the total number of source outputs being considered, should give us some idea of what the input distribution looks like. Be aware that this is a rather crude method and can at times be misleading. For example, if we were not careful in our selection of the source output, we might end up modeling some local peculiarities of the source. If the bins are too large, we might effectively filter out some important properties of the source. If the bin sizes are too small, we may miss out on some of the gross behavior of the source.

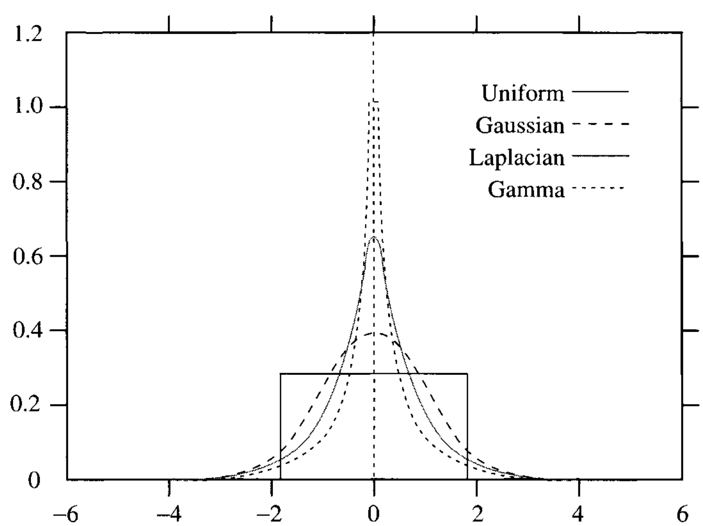

FIGURE 8. 5 Uniform, Gaussian, Laplacian, and Gamma distributions.

Once we have decided on some candidate distributions, we can select between them using a number of sophisticated tests. These tests are beyond the scope of this book but are described in [103].

Many of the sources that we deal with when we design lossy compression schemes have a great deal of structure in the form of sample-to-sample dependencies. The probability models described here capture none of these dependencies. Fortunately, we have a lot of models that can capture most of this structure. We describe some of these models in the next section.

8.6.2 Linear System Models

A large class of processes can be modeled in the form of the following difference equation:

$$x_n = \sum_{i=1}^{N} a_i x_{n-i} + \sum_{j=1}^{M} b_j \epsilon_{n-j} + \epsilon_n, \tag{8.84}$$

where $\{x_n\}$ are samples of the process we wish to model, and $\{\epsilon_n\}$ is a white noise sequence. We will assume throughout this book that we are dealing with real valued samples. Recall that a zero-mean wide-sense-stationary noise sequence $\{\epsilon_n\}$ is a sequence with autocorrelation function

$$R_{\epsilon\epsilon}(k) = \begin{cases} \sigma_\epsilon^2 & \text{for } k = 0 \\ 0 & \text{otherwise.} \end{cases} \tag{8.85}$$

In digital signal-processing terminology, Equation (8.84) represents the output of a linear discrete time invariant filter with N poles and M zeros. In the statistical literature, this model is called an autoregressive moving average model of order (N,M), or an ARMA (N,M) model. The autoregressive label is because of the first summation in Equation (8.84), while the second summation gives us the moving average portion of the name.

If all the b_j were zero in Equation (8.84), only the autoregressive part of the ARMA model would remain:

$$x_n = \sum_{i=1}^{N} a_i x_{n-i} + \epsilon_n. \tag{8.86}$$

This model is called an Nth-order autoregressive model and is denoted by AR(N). In digital signal-processing terminology, this is an *all pole filter*. The AR(N) model is the most popular of all the linear models, especially in speech compression, where it arises as a natural consequence of the speech production model. We will look at it a bit more closely.

First notice that for the AR(N) process, knowing all the past history of the process gives no more information than knowing the last N samples of the process; that is,

$$P(x_n | x_{n-1}, x_{n-2}, \ldots) = P(x_n | x_{n-1}, x_{n-2}, \ldots, x_{n-N}), \tag{8.87}$$

which means that the AR(N) process is a Markov model of order N.

The autocorrelation function of a process can tell us a lot about the sample-to-sample behavior of a sequence. A slowly decaying autocorrelation function indicates a high sample-to-sample correlation, while a fast decaying autocorrelation denotes low sample-to-sample correlation. In the case of *no* sample-to-sample correlation, such as white noise, the autocorrelation function is zero for lags greater than zero, as seen in Equation (8.85). The autocorrelation function for the AR(N) process can be obtained as follows:

$$R_{xx}(k) = E[x_n x_{n-k}] \tag{8.88}$$

$$= E\left[\left(\sum_{i=1}^{N} a_i x_{n-i} + \epsilon_n\right)(x_{n-k})\right] \tag{8.89}$$

$$= E\left[\sum_{i=1}^{N} a_i x_{n-i} x_{n-k}\right] + E[\epsilon_n x_{n-k}] \tag{8.90}$$

$$= \begin{cases} \sum_{i=1}^{N} a_i R_{xx}(k-i) & \text{for } k > 0 \\ \sum_{i=1}^{N} a_i R_{xx}(i) + \sigma_\epsilon^2 & \text{for } k = 0. \end{cases} \tag{8.91}$$

Example 8.6.1:

Suppose we have an AR(3) process. Let us write out the equations for the autocorrelation coefficient for lags 1, 2, 3:

$$R_{xx}(1) = a_1 R_{xx}(0) + a_2 R_{xx}(1) + a_3 R_{xx}(2)$$

$$R_{xx}(2) = a_1 R_{xx}(1) + a_2 R_{xx}(0) + a_3 R_{xx}(1)$$

$$R_{xx}(3) = a_1 R_{xx}(2) + a_2 R_{xx}(1) + a_3 R_{xx}(0).$$

If we know the values of the autocorrelation function $R_{xx}(k)$, for $k = 0, 1, 2, 3$, we can use this set of equations to find the AR(3) coefficients $\{a_1, a_2, a_3\}$. On the other hand, if we know the model coefficients and σ_ϵ^2, we can use the above equations along with the equation for $R_{xx}(0)$ to find the first four autocorrelation coefficients. All the other autocorrelation values can be obtained by using Equation (8.91). ♦

To see how the autocorrelation function is related to the temporal behavior of the sequence, let us look at the behavior of a simple AR(1) source.

Example 8.6.2:

An AR(1) source is defined by the equation

$$x_n = a_1 x_{n-1} + \epsilon_n. \tag{8.92}$$

The autocorrelation function for this source (see Problem 8) is given by

$$R_{xx}(k) = \frac{1}{1-a_1^2} a_1^k \sigma_\epsilon^2. \tag{8.93}$$

From this we can see that the autocorrelation will decay more slowly for larger values of a_1. Remember that the value of a_1 in this case is an indicator of how closely the current

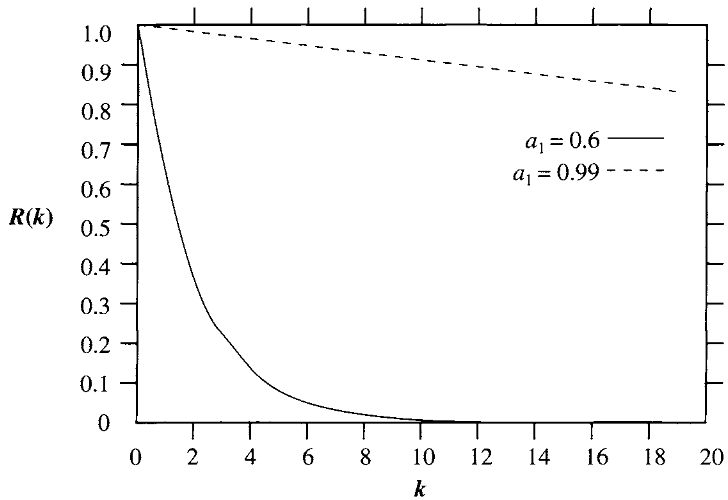

FIGURE 8.6 **Autocorrelation function of an AR(1) process with two values of a_1.**

sample is related to the previous sample. The autocorrelation function is plotted for two values of a_1 in Figure 8.6. Notice that for a_1 close to 1, the autocorrelation function decays extremely slowly. As the value of a_1 moves farther away from 1, the autocorrelation function decays much faster.

Sample waveforms for $a_1 = 0.99$ and $a_1 = 0.6$ are shown in Figures 8.7 and 8.8. Notice the slower variations in the waveform for the process with a higher value of a_1. Because

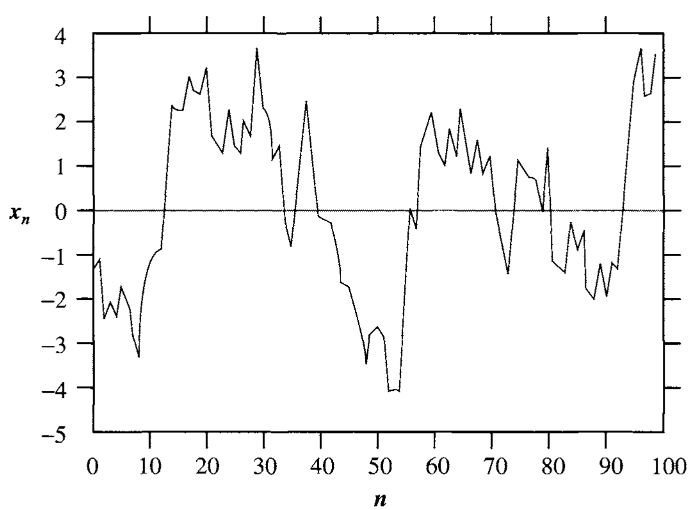

FIGURE 8.7 **Sample function of an AR(1) process with $a_1 = 0.99$.**

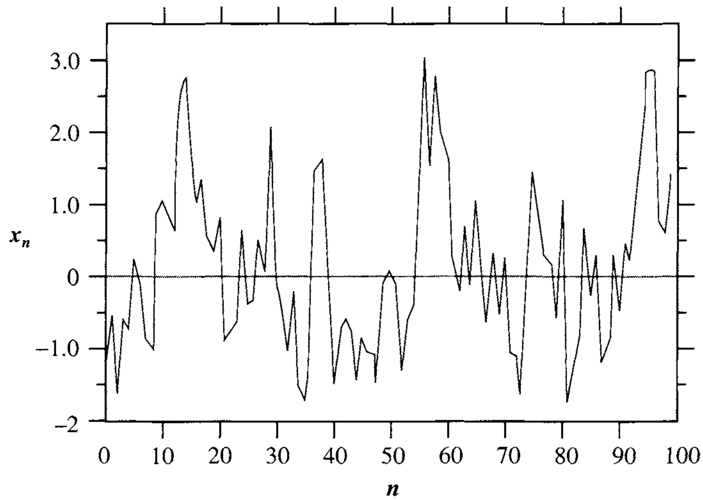

FIGURE 8.8 Sample function of an AR(1) process with $a_1 = 0.6$.

the waveform in Figure 8.7 varies more slowly than the waveform in Figure 8.8, samples of this waveform are much more likely to be close in value than the samples of the waveform of Figure 8.8.

Let's look at what happens when the AR(1) coefficient is negative. The sample waveforms are plotted in Figures 8.9 and 8.10. The sample-to-sample variation in these waveforms

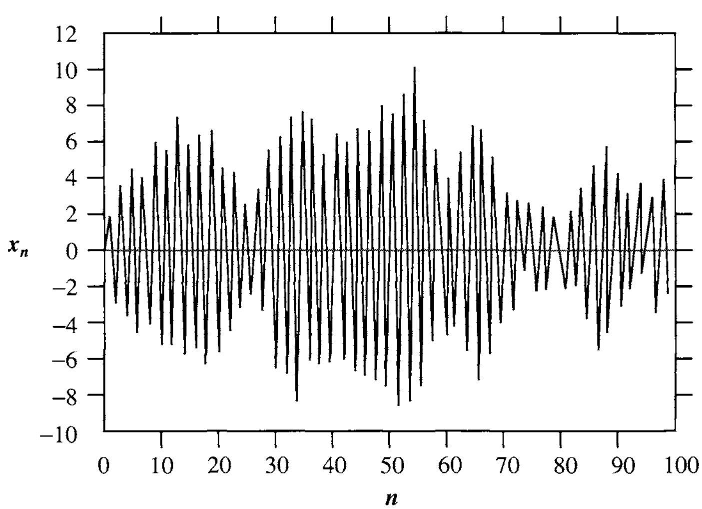

FIGURE 8.9 Sample function of an AR(1) process with $a_1 = -0.99$.

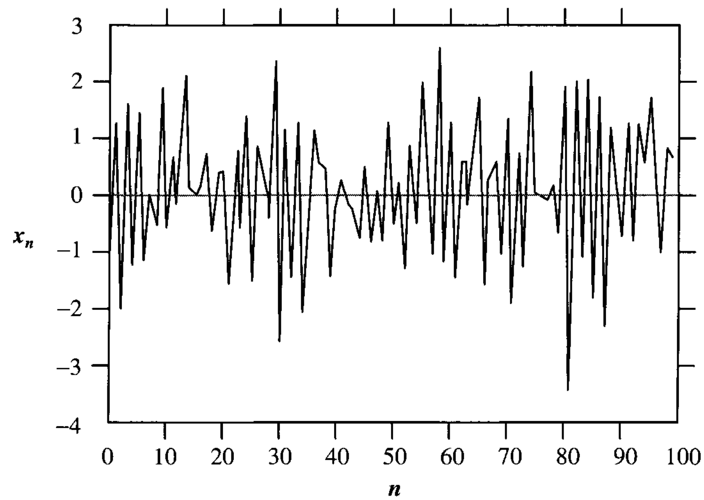

FIGURE 8.10 **Sample function of an AR(1) process with $a_1 = -0.6$.**

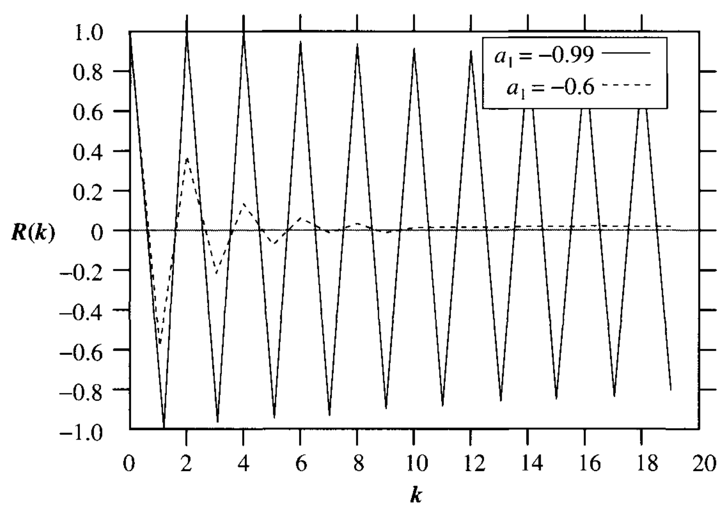

FIGURE 8.11 **Autocorrelation function of an AR(1) process with two negative values of a_1.**

is much higher than in the waveforms shown in Figures 8.7 and 8.8. However, if we were to look at the variation in magnitude, we can see that the higher value of a_1 results in magnitude values that are closer together.

This behavior is also reflected in the autocorrelation function, shown in Figure 8.11, as we might expect from looking at Equation (8.93). ♦

In Equation (8.84), instead of setting all the $\{b_j\}$ coefficients to zero, if we set all the $\{a_i\}$ coefficients to zero, we are left with the moving average part of the ARMA process:

$$x_n = \sum_{j=1}^{M} b_j \epsilon_{n-j} + \epsilon_n. \tag{8.94}$$

This process is called an Mth-order moving average process. This is a weighted average of the current and M past samples. Because of the form of this process, it is most useful when modeling slowly varying processes.

8.6.3 Physical Models

Physical models are based on the physics of the source output production. The physics are generally complicated and not amenable to a reasonable mathematical approximation. An exception to this rule is speech generation.

Speech Production

There has been a significant amount of research conducted in the area of speech production [104], and volumes have been written about it. We will try to summarize some of the pertinent aspects in this section.

Speech is produced by forcing air first through an elastic opening, the vocal cords, and then through cylindrical tubes with nonuniform diameter (the laryngeal, oral, nasal, and pharynx passages), and finally through cavities with changing boundaries such as the mouth and the nasal cavity. Everything past the vocal cords is generally referred to as the *vocal tract*. The first action generates the sound, which is then modulated into speech as it traverses through the vocal tract.

We will often be talking about filters in the coming chapters. We will try to describe filters more precisely at that time. For our purposes at present, a filter is a system that has an input and an output, and a rule for converting the input to the output, which we will call the *transfer function*. If we think of speech as the output of a filter, the sound generated by the air rushing past the vocal cords can be viewed as the input, while the rule for converting the input to the output is governed by the shape and physics of the vocal tract.

The output depends on the input and the transfer function. Let's look at each in turn. There are several different forms of input that can be generated by different conformations of the vocal cords and the associated cartilages. If the vocal cords are stretched shut and we force air through, the vocal cords vibrate, providing a periodic input. If a small aperture is left open, the input resembles white noise. By opening an aperture at different locations along the vocal cords, we can produce a white-noise–like input with certain dominant frequencies that depend on the location of the opening. The vocal tract can be modeled as a series of tubes of unequal diameter. If we now examine how an acoustic wave travels through this series of tubes, we find that the mathematical model that best describes this process is an autoregressive model. We will often encounter the autoregressive model when we discuss speech compression algorithms.

8.7 Summary

In this chapter we have looked at a variety of topics that will be useful to us when we study various lossy compression techniques, including distortion and its measurement, some new concepts from information theory, average mutual information and its connection to the rate of a compression scheme, and the rate distortion function. We have also briefly looked at some of the properties of the human visual system and the auditory system—most importantly, visual and auditory masking. The masking phenomena allow us to incur distortion in such a way that the distortion is not perceptible to the human observer. We also presented a model for speech production.

Further Reading

There are a number of excellent books available that delve more deeply in the area of information theory:

1. *Information Theory*, by R.B. Ash [15].
2. *Information Transmission*, by R.M. Fano [16].
3. *Information Theory and Reliable Communication*, by R.G. Gallagher [11].
4. *Entropy and Information Theory*, by R.M. Gray [17].
5. *Elements of Information Theory*, by T.M. Cover and J.A. Thomas [3].
6. *The Theory of Information and Coding*, by R.J. McEliece [6].

The subject of rate distortion theory is discussed in very clear terms in *Rate Distortion Theory*, by T. Berger [4].

For an introduction to the concepts behind speech perception, see *Voice and Speech Processing*, by T. Parsons [105].

8.8 Projects and Problems

1. Although SNR is a widely used measure of distortion, it often does not correlate with perceptual quality. In order to see this we conduct the following experiment. Using one of the images provided, generate two "reconstructed" images. For one of the reconstructions add a value of 10 to each pixel. For the other reconstruction, randomly add either $+10$ or -10 to each pixel.
   - **(a)** What is the SNR for each of the reconstructions? Do the relative values reflect the difference in the perceptual quality?
   - **(b)** Devise a mathematical measure that will better reflect the difference in perceptual quality for this particular case.
2. Consider the following lossy compression scheme for binary sequences. We divide the binary sequence into blocks of size M. For each block we count the number

of 0s. If this number is greater than or equal to $M/2$, we send a 0; otherwise, we send a 1.

(a) If the sequence is random with $P(0) = 0.8$, compute the rate and distortion (use Equation (8.54)) for $M = 1, 2, 4, 8, 16$. Compare your results with the rate distortion function for binary sources.

(b) Repeat assuming that the output of the encoder is encoded at a rate equal to the entropy of the output.

3. Write a program to implement the compression scheme described in the previous problem.

(a) Generate a random binary sequence with $P(0) = 0.8$, and compare your simulation results with the analytical results.

(b) Generate a binary first-order Markov sequence with $P(0|0) = 0.9$, and $P(1|1) = 0.9$. Encode it using your program. Discuss and comment on your results.

4. Show that

$$H(X_d|Y_d) = -\sum_{j=-\infty}^{\infty}\sum_{i=-\infty}^{\infty} f_{X|Y}(x_i|y_j)f_Y(y_j)\Delta\Delta \log f_{X|Y}(x_i|y_j) - \log \Delta. \tag{8.95}$$

5. For two random variables X and Y, show that

$$H(X|Y) \leq H(X)$$

with equality if X is independent of Y.
Hint: $E[\log(f(x))] \leq \log\{E[f(x)]\}$ (Jensen's inequality).

6. Given two random variables X and Y, show that $I(X; Y) = I(Y; X)$.

7. For a binary source with $P(0) = p$, $P(X = 0|Y = 1) = P(X = 1|Y = 0) = D$, and distortion measure

$$d(x_i, y_j) = x_i \oplus y_j,$$

show that

$$I(X; Y) = H_b(p) - H_b(D). \tag{8.96}$$

8. Find the autocorrelation function in terms of the model coefficients and σ_ϵ^2 for

(a) an AR(1) process,

(b) an MA(1) process, and

(c) an AR(2) process.