

- (c) Repeat your computations one more time for the case when the input is uniformly distributed inside a cube defined by

$$\{(0 < x_1 < 1), (0 < x_2 < 1), (0 < x_3 < 1)\}$$

Discuss the implications of the results of your computer simulations.

- 9.11 A problem that occasionally arises in the application of the SOM algorithm is the failure of topological ordering by creating a “folded” map. This problem arises when the neighborhood size is permitted to decay too rapidly. The creation of a folded map may be viewed as some form of a “local minimum” of the topological ordering process.

To investigate this phenomenon, consider a two-dimensional lattice of 10×20 neurons trained on a two-dimensional input uniformly distributed inside the square $\{(-1 < x_1 < +1), (-1 < x_2 < +1)\}$. Compute the map produced by the SOM algorithm, permitting the neighborhood function around the winning neuron to decay much faster than that normally used. You may have to repeat the experiment several times in order to see a failure of the ordering process.

- 9.12 The topological ordering property of the SOM algorithm may be used to form an abstract two-dimensional representation of a high-dimensional input space. To investigate this form of a representation, consider a two-dimensional lattice consisting of 10×10 neurons that is trained with an input consisting of four Gaussian clouds, $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, and $\mathcal{C}_4$, in an eight-dimensional input space. All the clouds have unit variance but different centers. The centers are located at the points $(0, 0, 0, \dots, 0)$, $(4, 0, 0, \dots, 0)$, $(4, 4, 0, \dots, 0)$, and $(0, 4, 0, \dots, 0)$. Compute the map produced by the SOM algorithm, with each neuron in the map being labeled with the particular class most represented by the input points around it.

- 9.13 Table P9.13 presents a summary of the *renormalized SOM algorithm*; a brief description of the algorithm is given in Section 9.3. Compare the conventional and renormalized SOM algorithms, keeping in mind the following two issues:

1. The coding complexity involved in algorithmic implementation.
2. The computer time taken to do the training.

Illustrate the comparison between these two algorithms using data drawn from a uniform distribution inside a square and the following two network configurations:

- (a) One-dimensional lattice of 257 neurons.
- (b) One-dimensional lattice of 2049 neurons.

In both cases, start with an initial number of code vectors equal to two.

- 9.14 Consider the signal-space diagram shown in Fig. P9.14 corresponding to M -ary pulse-amplitude modulation (PAM) with $M = 8$. The signal points correspond to Gray-encoded data blocks. Each signal point is represented by a rectangular pulse signal with appropriate amplitude scaling:

$$p(t) = \pm \frac{7}{2}, \pm \frac{5}{2}, \pm \frac{3}{2}, \pm \frac{1}{2}, \quad 0 \leq t \leq T$$

where T is the signaling interval. At the receiver input, white Gaussian noise of zero mean is added to the transmitted signal with varying signal-to-noise ratio (SNR). The SNR is defined as the ratio of the “average” transmitted signal power to the average noise power.

- (a) Using a random binary sequence as the transmitter input, generate data representing the received signal for SNR = 10, 20, and 30 dB.
- (b) For each of these SNRs, set up a self-organizing feature map. For typical values you may use:

- Input vector made up of eight elements obtained by sampling the received signal at a rate equal to eight times the signaling rate (i.e., 8 samples per signaling interval). Do not assume knowledge of timing information.
 - One-dimensional lattice of 64 neurons (i.e., eight times the size of the input vector).
- (c) Display the feature maps for each of the three SNRs, and thereby demonstrate the topological ordering property of the SOM algorithm.

TABLE P9.13 Summary of Renormalized Training Algorithm (One-Dimensional Version)

1. *Initialization.* Set the number of code vectors to be some small number (e.g., use two for simplicity or some other value more representative of the problem at hand). Initialize their positions to be those of a corresponding number of training vectors chosen randomly from the training set.
2. *Selection of an input vector.* Choose an input vector randomly from the training set.
3. *Encoding of the input vector.* Determine the “winning” code vector (i.e., the synaptic weight vector of the winning neuron). To do this, use either the “nearest neighbor” or the “minimum distortion” encoding prescription as required.
4. *Updating of the code book.* Do the usual “winner and its topological neighbors” update. You may find it sufficient to keep the learning-rate parameter η fixed (0.125, say) and to update the winning neuron using η , and its nearest neighbors using $\eta/2$, for example.
5. *Splitting of the code book.*^a Continue with the code book update (step 4), each time using a new input vector chosen randomly from the training set, until the number of code book updates is about 10–30 times the number of code vectors. When this number is reached, the code book has probably settled down, and it is time to split the code book. You may do so by taking the Peano string of code vectors that you have and interpolate their positions to generate a finer grained approximation to the Peano string; you may simply put an extra code vector halfway between each two existing code vectors.
6. *Completion of training.* The code book update and the code book splitting are continued until the total number of code vectors has reached some predetermined value (e.g., 100), at which time the training is all over.

^aThe splitting of the code book approximately doubles the number of code vectors after each epoch, so it does not take too many epochs to get to any prescribed number of code vectors.

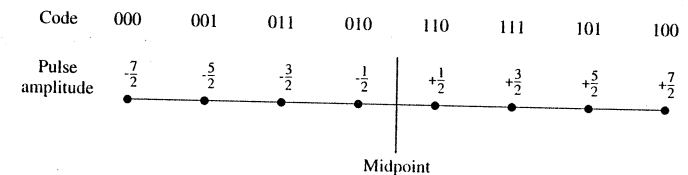


FIGURE P9.14

10

Information-Theoretic Models

0.1 INTRODUCTION

In a classic paper published in 1948, Claude Shannon laid down the foundations of *information theory*. Shannon's original work on information theory,¹ and its refinement by other researchers, was in direct response to the need for electrical engineers to design communication systems that are both *efficient* and *reliable*. In spite of its practical origins, information theory as we know it today is a deep mathematical theory concerned with the very essence of the *communication process*. The theory provides a framework for the study of fundamental issues such as the efficiency of information representation and the limitations involved in the reliable transmission of information over a communication channel. Moreover, the theory encompasses a multitude of powerful theorems for computing ideal *bounds* on the optimum representation and transmission of information-bearing signals. These bounds are important because they provide benchmarks for the improved design of information processing systems.

The main purpose of this chapter is to discuss *information-theoretic models* that lead to self-organization in a principled manner. In this context, a model that deserves special mention is the *maximum mutual information principle*² due to Linsker (1988). This principle states that the synaptic connections of a multilayered neural network develop in such a way as to *maximize the amount of information that is preserved when signals are transformed at each processing stage of the network, subject to certain constraints*. The idea that information theory may offer an explanation for perceptual processing is not new.³ For instance, we may mention an early paper by Attneave (1954), in which the following information-theoretic function is proposed for the perceptual system:

A major function of the perceptual machinery is to strip away some of the redundancy of stimulation, to describe or encode information in a form more economical than that in which it impinges on the receptors.

The main idea behind Attneave's paper is the recognition that encoding of data from a scene for the purpose of redundancy reduction is related to the identification of spe-

cific features in the scene. This important insight is related to a view of the brain described in Craik (1943), where a model of the external world is constructed so as to incorporate the regularities and constraints of the world.

Organization of the Chapter

The main body of the chapter is organized in two parts. The first part, consisting of Sections 10.2 through 10.5, provides a review of the fundamentals of information theory. In Section 10.2 we discuss the concept of entropy as a quantitative measure of information, which leads naturally to the maximum entropy principle discussed in Section 10.3. Next, in Section 10.4 we discuss the concept of mutual information and its properties, followed by a discussion of the Kullback–Leibler divergence in Section 10.5.

The second part of the chapter, consisting of Sections 10.6 through 10.14, deals with information-theoretic models for self-organizing systems. Section 10.6 highlights mutual information as an objective function to be optimized. The principle of maximum mutual information is discussed in Section 10.7, which is followed by a discussion of the relationship between this principle and that of redundancy reduction in Section 10.8. Sections 10.9 and 10.10 deal with two variants of the principle of maximum mutual information that are suitable for different applications in image processing. Sections 10.11 to 10.14 present three different methods for solving the blind source separation problem.

The chapter concludes with some final remarks in Section 10.15.

10.2 ENTROPY

Following the terminology commonly used in probability theory, we use an uppercase letter to denote a *random variable*, and the corresponding lowercase letter to denote the *value* of the random variable.

Consider then a random variable X , each realization (presentation) of which may be regarded as a *message*. Strictly speaking, if the random variable X is continuous in its amplitude range, then it carries an infinite amount of information. However, on physical and biological grounds we recognize that it is meaningless to think in terms of amplitude measurements with infinite precision, which suggests that the value of X may be uniformly *quantized* into a finite number of discrete levels. Accordingly, we may view X as a *discrete* random variable, modeled as follows:

$$X = \{x_k | k = 0, \pm 1, \dots, \pm K\} \quad (10.1)$$

where x_k is a discrete number and $(2K + 1)$ is the total number of discrete levels. The separation δx between the discrete levels is assumed to be small enough for the model of Eq. (10.1) to provide an adequate representation for the variable of interest. We may, of course, pass to the continuum limit by letting δx approach zero and K approach infinity, in which case we have a continuous random variable and (as we will see later in the section) sums become integrals.

To complete the model, let the event $X = x_k$ occur with *probability*

$$p_k = P(X = x_k) \quad (10.2)$$

with the requirement that

$$0 \leq p_k \leq 1 \quad \text{and} \quad \sum_{k=-K}^K p_k = 1 \quad (10.3)$$

Suppose that the event $X = x_k$ occurs with probability $p_k = 1$, which therefore requires that $p_i = 0$ for all $i \neq k$. In such a situation there is no "surprise" and therefore no "information" conveyed by the occurrence of the event $X = x_k$, since we know what the message must be. If, on the other hand, the various discrete levels were to occur with different probabilities and, in particular, the probability p_k is low, then there is more "surprise" and therefore "information" when X takes the value x_k rather than another value x_i with higher probability p_i , $i \neq k$. Thus the words "uncertainty," "surprise," and "information" are all related. Before the occurrence of the event $X = x_k$, there is an amount of uncertainty. When the event $X = x_k$ occurs, there is an amount of surprise. After the occurrence of the event $X = x_k$, there is an increase in the amount of information. These three amounts are obviously the same. Moreover, the amount of information is related to the *inverse* of the probability of occurrence.

We define the amount of information gained after observing the event $X = x_k$ with probability p_k as the logarithmic function

$$I(x_k) = \log\left(\frac{1}{p_k}\right) = -\log p_k \quad (10.4)$$

where the base of the logarithm is arbitrary. When the natural logarithm is used the units for information are *nats*, and when the base 2 logarithm is used the units are *bits*. In any case, the definition of information given in Eq. (10.4) exhibits the following properties:

$$1. \quad I(x_k) = 0 \quad \text{for } p_k = 1 \quad (10.5)$$

Obviously, if we are absolutely certain of the outcome of an event, there is *no* information gained by its occurrence.

$$2. \quad I(x_k) \geq 0 \quad \text{for } 0 \leq p_k \leq 1 \quad (10.6)$$

That is, the occurrence of an event $X = x_k$ either provides some or no information, but it never results in a loss of information.

$$3. \quad I(x_k) > I(x_i) \quad \text{for } p_k < p_i \quad (10.7)$$

That is, the less probable an event is, the more information we gain through its occurrence.

The amount of information $I(x_k)$ is a discrete random variable with probability p_k . The mean value of $I(x_k)$ over the complete range of $2K + 1$ discrete values is given by

$$\begin{aligned} H(X) &= E[I(x_k)] \\ &= \sum_{k=-K}^K p_k I(x_k) \\ &= - \sum_{k=-K}^K p_k \log p_k \end{aligned} \quad (10.8)$$

The quantity $H(X)$ is called the *entropy* of a random variable X permitted to take a finite set of discrete values; it is so called in recognition of the analogy between the definition given in Eq. (10.8) and that of entropy in statistical thermodynamics.⁴ The entropy $H(X)$ is a measure of the *average amount of information conveyed per message*. Note, however, that the X in $H(X)$ is not an argument of a function but rather a label for a random variable. Note also that in the definition of Eq. (10.8) we take $0 \log 0$ to be 0.

The entropy $H(X)$ is bounded as follows:

$$0 \leq H(X) \leq \log(2K + 1) \quad (10.9)$$

where $(2K + 1)$ is the total number of discrete levels. Furthermore, we may make the following statements:

1. $H(X) = 0$ if and only if the probability $p_k = 1$ for some k , and the remaining probabilities in the set are all zero; this lower bound on entropy corresponds to *no uncertainty*.
2. $H(X) = \log_2(2K + 1)$, if and only if $p_k = 1/(2K + 1)$ for all k (i.e., all the discrete levels are equiprobable); this upper bound on entropy corresponds to *maximum uncertainty*.

The proof of property 2 follows from the following lemma (Gray, 1990):

Given any two probability distributions $\{p_k\}$ and $\{q_k\}$ for a discrete random variable X , then

$$\sum_k p_k \log\left(\frac{p_k}{q_k}\right) \geq 0 \quad (10.10)$$

which is satisfied with equality if and only if $q_k = p_k$ for all k .

The quantity used in this lemma is of such fundamental importance that we pause to recast it in a form suitable for use in the study of stochastic systems. Let $p_X(x)$ and $q_X(x)$ denote the probabilities that random variable X is in state x under two different operating conditions. The *relative entropy* or *Kullback-Leibler divergence (distance)* between the two *probability mass functions* $p_X(x)$ and $q_X(x)$ is defined by (Kullback, 1968; Gray, 1990; Cover and Thomas, 1991)

$$D_{p\|q} = \sum_{x \in \mathcal{X}} p_X(x) \log\left(\frac{p_X(x)}{q_X(x)}\right) \quad (10.11)$$

where the sum is over all possible states of the system (i.e., the alphabet $\mathcal{X}$ of the discrete random variable X). The probability mass function $q_X(x)$ plays the role of a *reference measure*.

Differential Entropy of Continuous Random Variables

The discussion of information-theoretic concepts has thus far involved ensembles of random variables that are discrete in their amplitude values. We now extend some of these concepts to continuous random variables.

Consider a continuous random variable X with the probability density function $f_X(x)$. By analogy with the entropy of a discrete random variable, we introduce the following definition:

$$\begin{aligned} h(X) &= - \int_{-\infty}^{\infty} f_X(x) \log f_X(x) dx \\ &= -E[\log f_X(x)] \end{aligned} \quad (10.12)$$

We refer to $h(X)$ as the *differential entropy* of X to distinguish it from the ordinary or *absolute entropy*. We do so in recognition of the fact that although $h(X)$ is a useful mathematical quantity to know, it is *not* in any sense a measure of the randomness of X .

We justify the use of Eq. (10.12) as follows. We begin by viewing the continuous random variable X as the limiting form of a discrete random variable that assumes the value $x_k = k \delta x$, where $k = 0, \pm 1, \pm 2, \dots$, and δx approaches zero. By definition, the continuous random variable X assumes a value in the interval $[x_k, x_k + \delta x]$ with probability $f_X(x_k) \delta x$. Hence, permitting δx to approach zero, the ordinary entropy of the continuous random variable X may be written in the limit as

$$\begin{aligned} H(X) &= - \lim_{\delta x \rightarrow 0} \sum_{k=-\infty}^{\infty} f_X(x_k) \delta x \log (f_X(x_k) \delta x) \\ &= - \lim_{\delta x \rightarrow 0} \left[\sum_{k=-\infty}^{\infty} f_X(x_k) (\log f_X(x_k)) \delta x + \log \delta x \sum_{k=-\infty}^{\infty} f_X(x_k) \delta x \right] \\ &= - \int_{-\infty}^{\infty} f_X(x) \log f_X(x) dx - \lim_{\delta x \rightarrow 0} \log \delta x \int_{-\infty}^{\infty} f_X(x) dx \\ &= h(X) - \lim_{\delta x \rightarrow 0} \log \delta x \end{aligned} \quad (10.13)$$

where in the last line we have made use of Eq. (10.12) and the fact that the total area under the curve of the probability density function $f_X(x)$ is unity. In the limit as δx approaches zero, $-\log \delta x$ approaches infinity. This means that the entropy of a continuous random variable is infinitely large. Intuitively, we would expect this to be true because a continuous random variable may assume a value anywhere in the open interval $(-\infty, \infty)$ and the uncertainty associated with the variable is on the order of infinity. We avoid the problem associated with the term $\log \delta x$ by adopting $h(X)$ as a differential entropy, with the term $-\log \delta x$ serving as a reference. Moreover, since the information processed by a stochastic system as an entity of interest is actually the difference between two entropy terms that have a common reference, the information will be the same as the difference between the corresponding differential entropy terms. We are therefore perfectly justified in using the term $h(X)$, defined in Eq. (10.13), as the differential entropy of the continuous random variable X .

When we have a continuous random vector $\mathbf{X}$ consisting of n random variables $X_1, X_2, \dots, X_n$, we define the differential entropy of $\mathbf{X}$ as the n -fold integral

$$\begin{aligned} h(\mathbf{X}) &= - \int_{-\infty}^{\infty} f_{\mathbf{X}}(\mathbf{x}) \log f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} \\ &= -E[\log f_{\mathbf{X}}(\mathbf{x})] \end{aligned} \quad (10.14)$$

where $f_{\mathbf{X}}(\mathbf{x})$ is the joint probability density function of $\mathbf{X}$.

Example 10.1 Uniform Distribution

Consider a random variable X uniformly distributed inside the interval $[0, 1]$, as shown by

$$f_X(x) = \begin{cases} 1 & \text{for } 0 \leq x \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

By applying Eq. (10.12), we find that the differential entropy of X is

$$\begin{aligned} h(X) &= - \int_{-\infty}^{\infty} 1 \cdot \log 1 dx \\ &= - \int_{-\infty}^{\infty} 1 \cdot 0 dx \\ &= 0 \end{aligned}$$

The entropy of X is therefore zero.

Properties of Differential Entropy

From the definition of differential entropy $h(X)$ given in Eq. (10.12), we readily see that translation does not change its value; that is,

$$h(X + c) = h(X) \quad (10.15)$$

where c is constant.

Another useful property of $h(X)$ is described by

$$h(aX) = h(X) + \log |a| \quad (10.16)$$

where a is a scaling factor. To prove this property, we first recognize that since the area under the curve of a probability density function is unity, then

$$f_Y(y) = \frac{1}{|a|} f_X\left(\frac{y}{a}\right) \quad (10.17)$$

Next, using the formula of Eq. (10.12), we may write

$$\begin{aligned} h(Y) &= -E[\log f_Y(y)] \\ &= -E\left[\log \left(\frac{1}{|a|} f_X\left(\frac{y}{a}\right)\right)\right] \\ &= -E\left[\log f_X\left(\frac{y}{a}\right)\right] + \log |a| \end{aligned}$$

By putting $Y = aX$ in this relation we obtain

$$h(aX) = - \int_{-\infty}^{\infty} f_X(x) \log f_X(x) dx + \log |a|$$

from which Eq. (10.16) follows immediately.

Equation (10.16) applies to a scalar random variable. It may be generalized to the case of a random vector $\mathbf{X}$ premultiplied by matrix $\mathbf{A}$ as follows:

$$h(\mathbf{AX}) = h(\mathbf{X}) + \log|\det(\mathbf{A})| \quad (10.18)$$

where $\det(\mathbf{A})$ is the determinant of matrix $\mathbf{A}$.

10.3 MAXIMUM ENTROPY PRINCIPLE

Suppose that we are given a stochastic system with a set of known states but unknown probabilities, and that somehow we learn some *constraints* on the probability distribution of the states. The constraints can be certain ensemble average values or bounds on these values. The problem is to choose a probability model that is optimum in some sense, given this *prior knowledge* about the model. We usually find that there is an infinite number of possible models that satisfy the constraints. Which model should we choose?

The answer to this fundamental question lies in the *maximum entropy (Max Ent) principle*⁵ due to Jaynes (1957). The Max Ent principle may be stated as follows (Jaynes, 1957, 1982):

When an inference is made on the basis of incomplete information, it should be drawn from the probability distribution that maximizes the entropy, subject to constraints on the distribution.

In effect, the notion of entropy defines a kind of measure on the space of probability distributions, such that those distributions of high entropy are favored over others.

From this statement, it is apparent that the Max Ent problem is a constrained optimization problem. To illustrate the procedure for solving such a problem, consider the maximization of the differential entropy

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

over all probability density functions $f_X(x)$ of a random variable X , subject to the following constraints:

1. $f_X(x) \geq 0$, with equality outside the support of x .
2. $\int_{-\infty}^{\infty} f_X(x) dx = 1$.
3. $\int_{-\infty}^{\infty} f_X(x) g_i(x) dx = \alpha_i \quad \text{for } i = 1, 2, \dots, m$

where $g_i(x)$ is some function of x . Constraints 1 and 2 simply describe two fundamental properties of a probability density function. Constraint 3 defines the moments of X depending on how the function $g_i(x)$ is formulated. In effect, constraint 3 sums up the prior knowledge available about the random variable X . To solve this constrained optimization problem, we use the *method of Lagrange multipliers*⁶ by first formulating the objective function

$$J(f) = \int_{-\infty}^{\infty} \left[-f_X(x) \log f_X(x) + \lambda_0 f_X(x) + \sum_{i=1}^m \lambda_i g_i(x) f_X(x) \right] dx \quad (10.19)$$

where $\lambda_0, \lambda_1, \dots, \lambda_m$ are the *Lagrange multipliers*. Differentiating the integrand with respect to $f_X(x)$ and then setting the result equal to zero, we get

$$-1 - \log f_X(x) + \lambda_0 + \sum_{i=1}^m \lambda_i g_i(x) = 0$$

Solving this equation for the unknown $f_X(x)$, we get

$$f_X(x) = \exp \left(-1 + \lambda_0 + \sum_{i=1}^m \lambda_i g_i(x) \right) \quad (10.20)$$

The Lagrange multipliers in Eq. (10.20) are chosen in accordance with constraints 2 and 3. Equation (10.20) defines the maximum entropy distribution for this problem.

Example 10.2 One-dimensional Gaussian Distribution

Suppose the prior knowledge available to us is made up of the mean μ and variance σ^2 of a random variable X . By definition, we have

$$\int_{-\infty}^{\infty} (x - \mu)^2 f_X(x) dx = \sigma^2 = \text{constant}$$

Comparing this equation with constraint 3, we readily see that

$$g_1(x) = (x - \mu)^2 \\ \alpha_1 = \sigma^2$$

Hence, the use of Eq. (10.20) yields

$$f_X(x) = \exp[-1 + \lambda_0 + \lambda_1(x - \mu)^2]$$

Note that λ_1 has to be negative if the integrals of $f_X(x)$ and $(x - \mu)^2 f_X(x)$ with respect to x are to converge. Substituting this equation in equality constraints 2 and 3, and then solving for λ_0 and λ_1 , we get

$$\lambda_0 = 1 - \log(2\pi\sigma^2)$$

and

$$\lambda_1 = -\frac{1}{2\sigma^2}$$

The desired form for $f_X(x)$ is therefore described by

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (10.21)$$

which is recognized as the probability density of a *Gaussian random variable* X of mean μ and variance σ^2 . The maximum value of the differential entropy of such a random variable is given by

$$h(X) = \frac{1}{2} [1 + \log(2\pi\sigma^2)] \quad (10.22)$$

We may summarize the results of this example as follows:

1. For a given variance σ^2 , the Gaussian random variable has the largest differential entropy attainable by any random variable. That is, if X is a Gaussian random variable and Y is any other random variable with the same mean and variance, then for all Y

$$h(X) \geq h(Y)$$

with the equality holding only if X and Y are the same.

2. The entropy of a Gaussian random variable X is uniquely determined by the variance of X (i.e., it is independent of the mean of X).

Example 10.3 Multidimensional Gaussian Distribution

In this second example, we want to build on the results of Example 10.2 to evaluate the differential entropy of a *multidimensional Gaussian distribution*. Since the entropy of a Gaussian random variable X is independent of the mean of X , we may justifiably simplify the discussion in this example by considering an m -dimensional vector $\mathbf{X}$ of zero mean. Let the second-order statistics of $\mathbf{X}$ be described by the covariance matrix Σ defined as the expectation of the outer product of $\mathbf{X}$ with itself. The joint probability density function of the random vector $\mathbf{X}$ is given by (Wilks, 1962)

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{m/2} (\det(\Sigma))^{1/2}} \exp\left(-\frac{1}{2} \mathbf{x}^T \Sigma^{-1} \mathbf{x}\right) \quad (10.23)$$

where $\det(\Sigma)$ is the determinant of Σ . Equation (10.14) defines the differential entropy of $\mathbf{X}$. Therefore, substituting Eq. (10.23) in (10.14), we obtain the result

$$h(\mathbf{X}) = \frac{1}{2} [m + m \log(2\pi) + \log|\det(\Sigma)|] \quad (10.24)$$

which includes Eq. (10.22) as a special case. In light of the Max Ent principle, we may thus state that for a given covariance matrix Σ , the multivariate Gaussian distribution of Eq. (10.23) has the largest differential entropy attainable by any random vector of zero mean, and that maximum differential entropy is defined by Eq. (10.24).

10.4 MUTUAL INFORMATION

In the design of a self-organizing system, the primary objective is to develop an algorithm that is able to learn an input-output relationship of interest on the basis of input patterns alone. In this context, the notion of mutual information is of profound importance because of some highly desirable properties. To set the stage for the discussion, consider a stochastic system with input X and output Y . Both X and Y are permitted to take *discrete* values only, denoted by x and y , respectively. The entropy $H(X)$ is a measure of the prior uncertainty about X . How can we measure the uncertainty about X after observing Y ? In order to answer this question, we define the *conditional entropy* of X given Y as follows (Gray, 1990; Cover and Thomas, 1991):

$$H(X|Y) = H(X, Y) - H(Y) \quad (10.25)$$

with the property that

$$0 \leq H(X|Y) \leq H(X) \quad (10.26)$$

The conditional entropy $H(X|Y)$ represents the *amount of uncertainty remaining about the system input X after the system output Y has been observed*. The other quantity $H(X, Y)$ in Eq. (10.25) is the *joint entropy* of X and Y , which is defined by

$$H(X, Y) = - \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y)$$

where $p(x, y)$ is the *joint probability mass function* of discrete random variables X and Y , and $\mathcal{X}$ and $\mathcal{Y}$ are their respective alphabets.

Since the entropy $H(X)$ represents our uncertainty about the system input *before* observing the system output, and the conditional entropy $H(X|Y)$ represents our uncertainty about the system input *after* observing the system output, the difference $H(X) - H(X|Y)$ must represent our uncertainty about the system input that is resolved by observing the system output. This quantity is called the *mutual information* between the random variables X and Y . Denoting it by $I(X; Y)$, we may thus write⁷

$$I(X; Y) = H(X) - H(X|Y) \\ = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \left(\frac{p(x, y)}{p(x) p(y)} \right) \quad (10.27)$$

Entropy is a special case of mutual information, since we have

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

The mutual information $I(X; Y)$ between two discrete random variables X and Y has the following properties (Cover and Thomas, 1991; Gray 1990).

1. The mutual information between X and Y is symmetric; that is,

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

where the mutual information $I(Y; X)$ is a measure of the uncertainty about the system output Y that is resolved by observing the system input X , and the mutual information $I(X; Y)$ is a measure of the uncertainty about the system input that is resolved by observing the system output.

2. The mutual information between X and Y is always nonnegative; that is,

$$I(X; Y) \geq 0$$

In effect, this property states that we cannot lose information, on the average, by observing the system output Y . Moreover, the mutual information is zero if and only if the input and output of the system are statistically independent.

3. The mutual information between X and Y may be expressed in terms of the entropy of Y as

$$I(X; Y) = H(Y) - H(Y|X) \quad (10.28)$$

where $H(Y|X)$ is a *conditional entropy*. The right-hand side of Eq. (10.28) is the ensemble average of the information conveyed by the system output Y , minus the ensemble average of the information conveyed by Y given what we already know the system input X . This latter quantity, $H(Y|X)$, conveys information about the processing noise, rather than about the system input X .

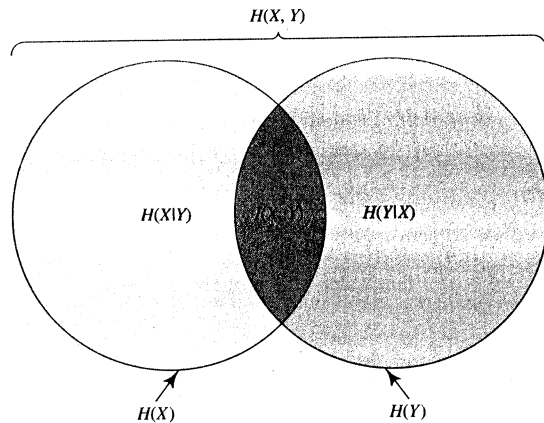


FIGURE 10.1 Illustration of the relations among the mutual information $I(X; Y)$ and the entropies $H(X)$ and $H(Y)$.

Figure 10.1 provides a visual interpretation of Eqs. (10.27) and (10.28). The entropy of the system input X is represented by the circle on the left. The entropy of the system output Y is represented by the circle on the right. The mutual information between X and Y is represented by the overlap between these two circles.

Mutual Information for Continuous Random Variables

Consider next a pair of continuous random variables X and Y . By analogy with Eq. (10.27), we define the *mutual information* between the random variables X and Y as

$$I(X; Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X,Y}(x, y) \log \left(\frac{f_{X,Y}(x, y)}{f_X(x)f_Y(y)} \right) dx dy \quad (10.29)$$

where $f_{X,Y}(x, y)$ is the joint probability density function of X and Y , and $f_X(x|y)$ is the conditional probability density function of X , given $Y = y$. Note that

$$f_{X,Y}(x, y) = f_X(x|y) f_Y(y)$$

and so we may also write

$$I(X; Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X,Y}(x, y) \log \left(\frac{f_{X,Y}(x, y)}{f_X(x)f_Y(y)} \right) dx dy$$

Also, by analogy with our previous discussion for discrete random variables, the mutual information $I(X; Y)$ between the continuous random variables X and Y has the following properties:

$$\begin{aligned} I(X; Y) &= h(X) - h(X|Y) \\ &= h(Y) - h(Y|X) \\ &= h(X) + h(Y) - h(X, Y) \end{aligned} \quad (10.30)$$

$$I(Y; X) = I(X; Y) \quad (10.31)$$

$$I(X; Y) \geq 0 \quad (10.32)$$

The parameter $h(X)$ is the differential entropy of X ; likewise for $h(Y)$. The parameter $h(X|Y)$ is the *conditional differential entropy* of X given Y ; it is defined by the double integral

$$h(X|Y) = - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{X,Y}(x, y) \log f_X(x|y) dx dy \quad (10.33)$$

The parameter $h(Y|X)$ is the conditional differential entropy of Y given X . It is defined in a manner similar to $h(X|Y)$. The parameter $h(X, Y)$ is the joint differential entropy of X and Y .

Note that Eq. (10.32) is satisfied with the equality sign only when the random variables X and Y are *statistically independent*. When this condition is satisfied, the joint probability density function of X and Y can be factored as

$$f_{X,Y}(x, y) = f_X(x)f_Y(y) \quad (10.34)$$

where $f_X(x)$ and $f_Y(y)$ are the marginal probability density functions of X and Y , respectively. Equivalently, we may write

$$f_X(x|y) = f_X(x)$$

which states that knowledge of the outcome of Y can in no way affect the distribution of X . Applying this condition to Eq. (10.29) reduces the mutual information $I(X; Y)$ between X and Y to zero.

The definition of mutual information $I(X; Y)$ given in Eq. (10.29) applies to scalar random variables X and Y . This definition may be readily extended to random vectors $\mathbf{X}$ and $\mathbf{Y}$, and we may thus write $I(\mathbf{X}; \mathbf{Y})$. Specifically, we define $I(\mathbf{X}; \mathbf{Y})$ as the multifold integral:

$$I(\mathbf{X}; \mathbf{Y}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \mathbf{y}) \log \left(\frac{f_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \mathbf{y})}{f_{\mathbf{X}}(\mathbf{x})f_{\mathbf{Y}}(\mathbf{y})} \right) d\mathbf{x} d\mathbf{y} \quad (10.35)$$

The mutual information $I(\mathbf{X}; \mathbf{Y})$ has properties that parallel those given in Eqs. (10.30) to (10.32) for scalar random variables.

10.5 KULLBACK–LEIBLER DIVERGENCE

In Eq. (10.11) we define the Kullback–Leibler divergence for discrete random variables. This definition may be extended to the general case of continuous random vectors. Let $f_{\mathbf{X}}(\mathbf{x})$ and $g_{\mathbf{X}}(\mathbf{x})$ denote two different probability density functions of an m -by-1 random vector $\mathbf{X}$. In light of Eq. (10.11), we may then define the *Kullback–Leibler divergence* between $f_{\mathbf{X}}(\mathbf{x})$ and $g_{\mathbf{X}}(\mathbf{x})$ as follows (Kullback, 1968; Shore and Johnson, 1980):

$$D_{f_{\mathbf{X}} \| g_{\mathbf{X}}} = \int_{-\infty}^{\infty} f_{\mathbf{X}}(\mathbf{x}) \log \left(\frac{f_{\mathbf{X}}(\mathbf{x})}{g_{\mathbf{X}}(\mathbf{x})} \right) d\mathbf{x} \quad (10.36)$$

The Kullback–Leibler divergence has some unique properties:

1. It always has a positive value or is zero. For the special case when $f_{\mathbf{X}}(\mathbf{x}) = g_{\mathbf{X}}(\mathbf{x})$, we have a perfect match between the two distributions, and $D_{f_{\mathbf{X}}\|g}$ is exactly zero.
2. It is invariant with respect to the following changes in the components of the vector $\mathbf{x}$:
 - Permutation of the order in which the components are arranged.
 - Amplitude scaling.
 - Monotonic nonlinear transformation.

The mutual information $I(\mathbf{X}; \mathbf{Y})$ between a pair of vectors $\mathbf{X}$ and $\mathbf{Y}$ has an interesting interpretation in terms of the Kullback–Leibler divergence. First, we note that

$$f_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \mathbf{y}) = f_{\mathbf{Y}}(\mathbf{y}|\mathbf{x})f_{\mathbf{X}}(\mathbf{x}) \quad (10.37)$$

Hence, we may rewrite Eq. (10.35) in the equivalent form

$$I(\mathbf{X}; \mathbf{Y}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \mathbf{y}) \log \left(\frac{f_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \mathbf{y})}{f_{\mathbf{X}}(\mathbf{x})f_{\mathbf{Y}}(\mathbf{y})} \right) d\mathbf{x} d\mathbf{y}$$

Comparing this formula with that of Eq. (10.36), we immediately deduce the following result:

$$I(\mathbf{X}; \mathbf{Y}) = D_{f_{\mathbf{X},\mathbf{Y}}\|f_{\mathbf{X}}f_{\mathbf{Y}}} \quad (10.38)$$

In words, the mutual information $I(\mathbf{X}; \mathbf{Y})$ between $\mathbf{X}$ and $\mathbf{Y}$ is equal to the Kullback–Leibler divergence between the joint probability density function $f_{\mathbf{X},\mathbf{Y}}(\mathbf{x}, \mathbf{y})$ and the product of the probability density functions $f_{\mathbf{X}}(\mathbf{x})$ and $f_{\mathbf{Y}}(\mathbf{y})$.

A special case of this latter result is the Kullback–Leibler divergence between the probability density function $f_{\mathbf{X}}(\mathbf{x})$ of an m -by-1 random vector $\mathbf{X}$ and the product of its m marginal probability density functions. Let $\tilde{f}_{X_i}(x_i)$ denote the i th marginal probability density function of element X_i , which is defined by

$$\tilde{f}_{X_i}(x_i) = \int_{-\infty}^{\infty} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}^{(i)}, \quad i = 1, 2, \dots, m \quad (10.39)$$

where $\mathbf{x}^{(i)}$ is the $(m-1)$ -by-1 vector left after removing the i th element from vector $\mathbf{x}$. The Kullback–Leibler divergence between $f_{\mathbf{X}}(\mathbf{x})$ and the *factorial distribution* $\prod_i \tilde{f}_{X_i}(x_i)$ is given by

$$D_{f_{\mathbf{X}}\|\tilde{f}_{\mathbf{X}}} = \int_{-\infty}^{\infty} f_{\mathbf{X}}(\mathbf{x}) \log \left(\frac{f_{\mathbf{X}}(\mathbf{x})}{\prod_{i=1}^m \tilde{f}_{X_i}(x_i)} \right) d\mathbf{x} \quad (10.40)$$

which may also be written in the expanded form

$$D_{f_{\mathbf{X}}\|\tilde{f}_{\mathbf{X}}} = \int_{-\infty}^{\infty} f_{\mathbf{X}}(\mathbf{x}) \log f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x} - \sum_{i=1}^m \int_{-\infty}^{\infty} f_{\mathbf{X}}(\mathbf{x}) \log \tilde{f}_{X_i}(x_i) d\mathbf{x} \quad (10.41)$$

The first integral on the right-hand side of Eq. (10.41) is, by definition, equal to $-h(\mathbf{X})$, where $h(\mathbf{X})$ is the differential entropy of $\mathbf{X}$. To deal with the second term we first note that

$$d\mathbf{x} = d\mathbf{x}^{(i)} dx_i$$

Hence we may write

$$\int_{-\infty}^{\infty} f_{\mathbf{X}}(\mathbf{x}) \log \tilde{f}_{X_i}(x_i) d\mathbf{x} = \int_{-\infty}^{\infty} \log \tilde{f}_{X_i}(x_i) \int_{-\infty}^{\infty} f_{\mathbf{X}}(\mathbf{x}) d\mathbf{x}^{(i)} dx_i \quad (10.42)$$

where the inner integral on the right-hand side is with respect to the $(m-1)$ -by-1 vector $\mathbf{x}^{(i)}$ and the outer integral is with respect to the scalar x_i . But from Eq. (10.39) we see that the inner integral is in fact equal to the marginal probability density function $\tilde{f}_{X_i}(x_i)$. Accordingly, we may rewrite Eq. (10.42) in the equivalent form

$$\begin{aligned} \int_{-\infty}^{\infty} f_{\mathbf{X}}(\mathbf{x}) \log \tilde{f}_{X_i}(x_i) d\mathbf{x} &= \int_{-\infty}^{\infty} \tilde{f}_{X_i}(x_i) \log \tilde{f}_{X_i}(x_i) dx_i \\ &= -\tilde{h}(X_i), \quad i = 1, 2, \dots, m \end{aligned} \quad (10.43)$$

where $\tilde{h}(X_i)$ is the i th *marginal entropy* (i.e., the differential entropy based on the marginal probability density function $\tilde{f}_{X_i}(x_i)$). Finally, using Eq. (10.43) in (10.41) and noting that the first integral in Eq. (10.41) is equal to $-h(\mathbf{X})$, we may simplify the Kullback–Leibler divergence of Eq. (10.41) to

$$D_{f_{\mathbf{X}}\|\tilde{f}_{\mathbf{X}}} = -h(\mathbf{X}) + \sum_{i=1}^m \tilde{h}(X_i) \quad (10.44)$$

This formula will be of particular use in our study of the blind source separation problem later in the chapter.

Pythagorean Decomposition

Next we consider the Kullback–Leibler divergence between the probability density functions $f_{\mathbf{X}}(\mathbf{x})$ and $f_{\mathbf{U}}(\mathbf{x})$. The m -by-1 random vector $\mathbf{U}$ consists of independent variables, as shown by

$$f_{\mathbf{U}}(\mathbf{x}) = \prod_{i=1}^m f_{U_i}(x_i)$$

and the m -by-1 random vector $\mathbf{X}$ is defined in terms of $\mathbf{U}$ by

$$\mathbf{X} = \mathbf{A}\mathbf{U}$$

where $\mathbf{A}$ is a nondiagonal matrix. Let $\tilde{f}_{X_i}(x_i)$ denote the marginal probability density function of each X_i that is derived from $f_{\mathbf{X}}(\mathbf{x})$. Then the Kullback–Leibler divergence between $f_{\mathbf{X}}(\mathbf{x})$ and $f_{\mathbf{U}}(\mathbf{x})$ admits the following *Pythagorean decomposition*:

$$D_{f_{\mathbf{X}}\|f_{\mathbf{U}}} = D_{f_{\mathbf{X}}\|\tilde{f}_{\mathbf{X}}} + D_{\tilde{f}_{\mathbf{X}}\|f_{\mathbf{U}}} \quad (10.45)$$

We refer to this classic relation as a Pythagorean decomposition because it has an information-geometric interpretation (Amari, 1985). A proof of the decomposition is given in note 8.

6 MUTUAL INFORMATION AS AN OBJECTIVE FUNCTION TO BE OPTIMIZED

Now that we have developed an adequate understanding of Shannon's information theory, we are ready to discuss its role in the study of self-organizing systems.

To proceed with the discussion, consider a neural system with multiple inputs and outputs. The primary objective here is for the system to be self-organizing, designed for a specific task (e.g., modeling, extraction of statistically salient features, or signal separation). This requirement can be satisfied by choosing the mutual information between certain variables of the system as the *objective function* to be optimized. This particular choice is justified by the following considerations:

- The mutual information has some unique properties as discussed in Section 10.4.
- It can be determined without the need for a teacher so the provision for self-organization is naturally met.

The problem thus becomes one of adjusting the free parameters (i.e., synaptic weights) of the system so as to optimize the mutual information.

Depending on the application of interest, we may identify four different scenarios as illustrated in Fig. 10.2, which can arise in practice. These scenarios are described as follows:

- In scenario 1 depicted in Fig. 10.2a, the input vector $\mathbf{X}$ is composed of the elements $X_1, X_2, \dots, X_m$, and the output vector $\mathbf{Y}$ is composed of the elements $Y_1, Y_2, \dots, Y_l$. The requirement is to *maximize the information conveyed to the system output $\mathbf{Y}$ about the system input $\mathbf{X}$* .

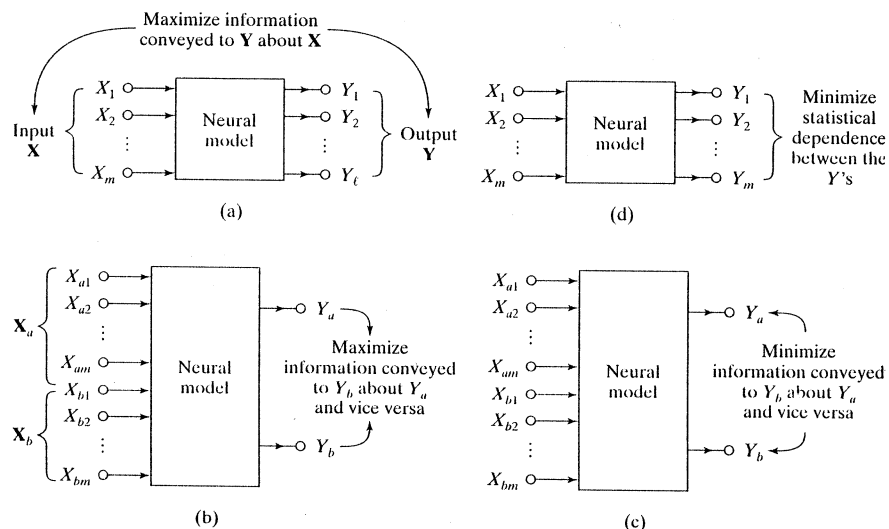


FIGURE 10.2 Four basic scenarios that lend themselves to the application of Infomax and its three variants.

- In scenario 2 depicted in Fig. 10.2b, a pair of input vectors $\mathbf{X}_a$ and $\mathbf{X}_b$ is derived from adjacent but nonoverlapping regions of an image. The inputs $\mathbf{X}_a$ and $\mathbf{X}_b$ produce scalar outputs Y_a and Y_b , respectively. The requirement is to *maximize the information conveyed to Y_a about Y_b* , and vice versa.
- In scenario 3 depicted in Fig. 10.2c, the input vectors $\mathbf{X}_a$ and $\mathbf{X}_b$ are derived from a corresponding pair of regions belonging to two separate images. The outputs produced by these two input vectors are denoted by Y_a and Y_b , respectively. The objective is to *minimize the information conveyed to Y_a about Y_b* .
- In scenario 4 depicted in Fig. 10.2d, the input vector $\mathbf{X}$ and output vector $\mathbf{Y}$ are defined in a manner similar to those in Fig. 10.2a but with equal dimensionality (i.e., $l = m$). The objective here is for the *statistical dependence between the components of the output vector $\mathbf{Y}$ to be minimized*.

In all of these situations, mutual information plays a central role. However, the way in which it is formulated depends on the particular situation being considered. In the rest of the chapter, the issues involved in these scenarios and their practical implications are discussed in the same order just presented.

10.7 MAXIMUM MUTUAL INFORMATION PRINCIPLE

The idea of designing a neural processor to maximize the mutual information $I(\mathbf{Y}; \mathbf{X})$ is appealing as the basis for statistical signal processing. This method of optimization is embodied in the *maximum mutual information (Infomax) principle* due to Linsker (1987, 1988a, 1989a), which may be stated formally as follows:

The transformation of a random vector $\mathbf{X}$ observed in the input layer of a neural system to a random vector $\mathbf{Y}$ produced in the output layer of the system should be so chosen that the activities of the neurons in the output layer jointly maximize information about the activities in the input layer. The objective function to be maximized is the mutual information $I(\mathbf{Y}; \mathbf{X})$ between the vectors $\mathbf{X}$ and $\mathbf{Y}$.

The Infomax principle provides a mathematical framework for self-organization of the signal transmission system described in Fig. 10.2a that is independent of the rule used for its implementation. Also, this principle may be viewed as the neural network counterpart of the concept of *channel capacity*, which defines the Shannon limit on the rate of information transmission through a communication channel.

In the sequel, we illustrate application of the Infomax principle with two examples involving a single noisy neuron. In one example the noise appears at the output, and in the other example it appears at the input.

Example 10.4 A Single Neuron Corrupted by Processing Noise

Consider the simple case of a linear neuron that receives its inputs from a set of m source nodes. Let the output of this neuron in the presence of *processing noise* be expressed as

$$Y = \left(\sum_{i=1}^m w_i X_i \right) + N \quad (10.46)$$

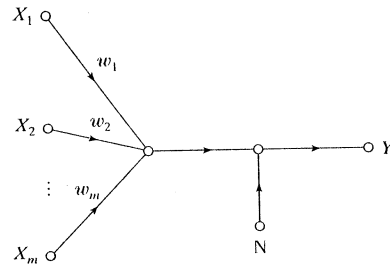


FIGURE 10.3 Signal-flow graph of a noisy neuron.

where w_i is the i th synaptic weight and N is the processing noise, as modeled in Fig. 10.3. It is assumed that:

- The output Y of the neuron is a Gaussian random variable with variance σ_Y^2 .
- The processing noise N is also a Gaussian random variable with zero mean and variance σ_N^2 .
- The processing noise is uncorrelated with any of the input components. That is,

$$E[NX_i] = 0 \quad \text{for all } i$$

The Gaussianity of the output Y can be satisfied in one of two ways. The inputs $X_1, X_2, \dots, X_m$ are all Gaussian-distributed. Then with the additive noise N assumed to be Gaussian too, the Gaussianity of Y is assured by virtue of the fact that it is the weighted sum of a number of Gaussian-distributed random variables. Alternatively, the inputs $X_1, X_2, \dots, X_m$ are identically and independently distributed, in which case the distribution of their weighted sum approaches a Gaussian distribution for large m by the central limit theorem.

To proceed with the analysis, we first note from the second line of Eq. (10.30) that the mutual information $I(Y; \mathbf{X})$ between the output Y of the neuron and the input vector $\mathbf{X}$ is

$$I(Y; \mathbf{X}) = h(Y) - h(Y|\mathbf{X}) \quad (10.47)$$

In view of Eq. (10.46), we note that the probability density function of Y , given the input vector $\mathbf{X}$, is the same as the probability density function of a constant plus a Gaussian-distributed random variable. Accordingly, the conditional entropy $h(Y|\mathbf{X})$ is the "information" that the output neuron conveys about the processing noise N rather than about the signal vector $\mathbf{X}$. We may thus set

$$h(Y|\mathbf{X}) = h(N)$$

and therefore rewrite Eq. (10.47) simply as

$$I(Y; \mathbf{X}) = h(Y) - h(N) \quad (10.48)$$

By applying Eq. (10.22) for the differential entropy of a Gaussian random variable to the problem at hand, we obtain

$$h(Y) = \frac{1}{2} [1 + \log(2\pi\sigma_Y^2)] \quad (10.49)$$

and

$$h(N) = \frac{1}{2} [1 + \log(2\pi\sigma_N^2)] \quad (10.50)$$

After simplification, the use of Eqs. (10.49) and (10.50) in (10.48) yields

$$I(Y; \mathbf{X}) = \frac{1}{2} \log \left(\frac{\sigma_Y^2}{\sigma_N^2} \right) \quad (10.51)$$

where σ_Y^2 depends on σ_N^2 .

The ratio σ_Y^2/σ_N^2 may be viewed as a *signal-to-noise ratio*. Imposing the constraint that the noise variance σ_N^2 is fixed, we see from Eq. (10.51) that the mutual information $I(Y; \mathbf{X})$ is maximized by maximizing the variance σ_Y^2 of the neuron output Y . We may therefore state that under certain conditions, maximizing the output variance of a neuron maximizes the mutual information between the output signal of that neuron and its inputs (Linsker, 1988a).

Example 10.5 A Single Neuron Corrupted by Additive Input Noise

Suppose that the noise corrupting the behavior of a linear neuron originates at the input ends of the synapses as shown in the model of Fig. 10.4. According to this second noise model, we have

$$Y = \sum_{i=1}^m w_i(X_i + N_i) \quad (10.52)$$

where each noise N_i is assumed to be an independent Gaussian random variable with zero mean and variance $\sigma_{N_i}^2$. We may rewrite Eq. (10.52) in a form similar to that of Eq. (10.46), as shown by

$$Y = \left(\sum_{i=1}^m w_i X_i \right) + N'$$

where N' is a composite noise component, defined by

$$N' = \sum_{i=1}^m w_i N_i$$

The noise N' has a Gaussian distribution with zero mean and a variance equal to the sum of the variances of its independent noise components; that is,

$$\sigma_{N'}^2 = \sum_{i=1}^m w_i^2 \sigma_{N_i}^2$$

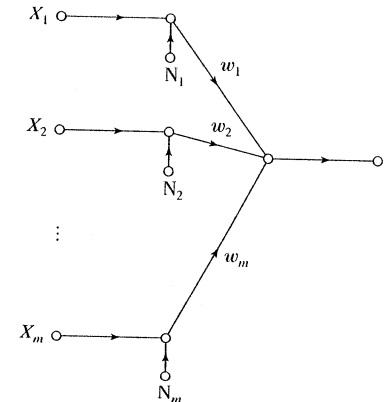


FIGURE 10.4 Another noise model.

As before, we assume that the output Y of the neuron has a Gaussian distribution with variance σ_Y^2 . The mutual information $I(Y; \mathbf{X})$ between Y and $\mathbf{X}$ is still given by Eq. (10.47). This time, however, the conditional entropy $h(Y|\mathbf{X})$ is defined by

$$\begin{aligned} h(Y|\mathbf{X}) &= h(N') \\ &= \frac{1}{2} (1 + 2\pi\sigma_N^2) \\ &= \frac{1}{2} \left[1 + 2\pi\sigma_N^2 \sum_{i=1}^m w_i^2 \right] \end{aligned} \quad (10.53)$$

Thus, using Eqs. (10.49) and (10.53) in (10.47), and then simplifying terms, we get (Linsker, 1988a)

$$I(Y; \mathbf{X}) = \frac{1}{2} \log \left(\frac{\sigma_Y^2}{\sigma_N^2 \sum_{i=1}^m w_i^2} \right) \quad (10.54)$$

Under the constraint that the noise variance σ_N^2 is maintained constant, the mutual information $I(Y; \mathbf{X})$ is now maximized by maximizing the ratio $\sigma_Y^2 / \sum_{i=1}^m w_i^2$, where σ_Y^2 is a function of w_i . ■

What can we deduce from Examples 10.4 and 10.5? First, we see from the material presented in these two examples that the result of applying the Infomax principle is problem-dependent. The equivalence between maximizing the mutual information $I(Y; \mathbf{X})$ and output variance that applies to the model of Fig. 10.3, for a prescribed noise variance σ_N^2 , does not carry over to the model of Fig. 10.4. It is only when we impose the constraint $\sum_i w_i^2 = 1$ on the model of Fig. 10.4 that both models behave in a similar manner.

In general, the determination of the mutual information $I(\mathbf{Y}; \mathbf{X})$ between input vector $\mathbf{X}$ and output vector $\mathbf{Y}$ is a difficult task. In Examples 10.4 and 10.5, we made the analysis mathematically tractable by assuming that the noise distributions in a system with one or more sources of noise are *multivariate Gaussian*. This assumption needs to be justified.

In adopting a Gaussian noise model, we are in essence invoking a “surrogate” mutual information computed on the premise that the output vector $\mathbf{Y}$ of a neuron has a multivariate Gaussian distribution with the same mean vector and covariance matrix as the actual distribution. In Linsker (1993), the Kullback–Leibler divergence is used to provide a principled justification for the use of such surrogate mutual information, under the condition that the network has stored information about the mean vector and covariance matrix of the output vector $\mathbf{Y}$, but not about higher-order statistics.

Finally, the analysis presented in both Examples 10.4 and 10.5 was carried out in the context of a single neuron. This was done purposely with a specific point in mind: For the Infomax principle to be mathematically tractable, the optimization should be performed at the local neuronal level. Such optimization is consistent with the essence of self-organization.

Example 10.6

In Examples 10.4 and 10.5 we considered noisy neurons. In this example we consider a noiseless network that transforms a random vector $\mathbf{X}$ of arbitrary distribution to a new random vector $\mathbf{Y}$ of different distribution. Recognizing that $I(\mathbf{X}; \mathbf{Y}) = I(\mathbf{Y}; \mathbf{X})$ and extending Eq. (10.28) to the

situation described here, we may express the mutual information between the input vector $\mathbf{X}$ and output vector $\mathbf{Y}$ as follows:

$$I(\mathbf{Y}; \mathbf{X}) = H(\mathbf{Y}) - H(\mathbf{Y}|\mathbf{X})$$

where $H(\mathbf{Y})$ is the entropy of $\mathbf{Y}$ and $H(\mathbf{Y}|\mathbf{X})$ is the conditional entropy of $\mathbf{Y}$ given $\mathbf{X}$. With the mapping from $\mathbf{X}$ to $\mathbf{Y}$ assumed to be noiseless, the conditional entropy $H(\mathbf{Y}|\mathbf{X})$ attains its lowest possible value: it diverges to $-\infty$. This result is due to the differential nature of the entropy of a continuous random variable that was discussed in Section 10.2. However, this difficulty is of no consequence when we consider the *gradient* of the mutual information $I(\mathbf{Y}; \mathbf{X})$ with respect to a weight matrix $\mathbf{W}$ that parameterizes the mapping network. Specifically, we may write:

$$\frac{\partial I(\mathbf{Y}; \mathbf{X})}{\partial \mathbf{W}} = \frac{\partial H(\mathbf{Y})}{\partial \mathbf{W}} \quad (10.55)$$

because the conditional entropy $H(\mathbf{Y}|\mathbf{X})$ is independent of $\mathbf{W}$. Equation (10.55) shows that for a noiseless mapping network, maximizing the entropy of the network output $\mathbf{Y}$ is equivalent to maximizing the mutual information between $\mathbf{Y}$ and the network input $\mathbf{X}$, with both maximizations being performed with respect to the weight matrix $\mathbf{W}$ of the mapping network (Bell and Sejnowski, 1995). ■

10.8 INFOMAX AND REDUNDANCY REDUCTION

In Shannon's framework of information theory, order and structure represent *redundancy*, which diminishes uncertainty that is resolved by the receipt of information. The more order and structure we have in the underlying process, the less information we receive by observing that process. Consider for example, the highly structured and redundant sequence of examples *aaaaaa*. On receiving the first example *a*, we can immediately say that the remaining five examples are all the same. The information conveyed by such a sequence of examples is limited to that contained in a single example. In other words, the more redundant a sequence of examples is, the less information content is received from the environment.

From the definition of mutual information $I(\mathbf{Y}; \mathbf{X})$, we know it is a measure of the uncertainty about the output $\mathbf{Y}$ of a system that is resolved by observing the system input $\mathbf{X}$. The Infomax principle operates by maximizing the mutual information $I(\mathbf{Y}; \mathbf{X})$, as a result of which we are more certain about the system output $\mathbf{Y}$ by observing the system input $\mathbf{X}$. In light of the previously mentioned relationship between information and redundancy, we may therefore say that the Infomax principle leads to a reduction in redundancy in the output $\mathbf{Y}$ compared to that in the input $\mathbf{X}$.

The presence of noise is a factor that prompts the use of redundancy and the related method of diversity (Linsker, 1988a). When the additive noise in the input signal is high, we may use redundancy to combat the degrading effects of noise. In such an environment more of the (correlated) components of the input signal are combined by the processor to provide an accurate representation of the input. Also when the output noise (i.e., processor noise) is high, more of the output components are directed by the processor to provide redundant information. The number of independent properties observed at the output of the processor is thereby reduced, but the representation accuracy of each property is increased. We may thus state that *a high level of noise favors redundancy of representation*. When, however, the *noise level is low, diversity of*

representation is favored over redundancy. By diversity we mean two or more outputs with different properties being produced by the processor. Problem 10.6 discusses the redundancy/diversity tradeoff from the perspective of Infomax. It is noteworthy that the redundancy/diversity tradeoff is somewhat analogous (though different from) the bias/variance tradeoff discussed in Chapter 2.

Modeling of a Perceptual System

Since the early days of information theory, it has been suggested that the redundancy of sensory messages (stimuli) is important for understanding perception (Attneave, 1954; Barlow, 1959). Indeed, the redundancy of sensory messages provides the *knowledge* that enables the brain to build up its “cognitive maps” or “working models” of the environment around it (Barlow, 1989). Regularities in the sensory messages must somehow be recoded by the brain for it to know what usually happens. However, *redundancy reduction* is the more specific form of *Barlow's hypothesis*. This hypothesis says that the purpose of early processing is to transform the highly redundant sensory input into a more efficient *factorial code*. In other words, the neuronal outputs become *statistically independent* when conditioned on the input.

Inspired by Barlow's hypothesis, Atick and Redlich (1990) postulated the *principle of minimum redundancy* as the basis for an information-theoretic model of the perceptual system shown in Fig. 10.5. The model consists of three components: *input channel*, *recoding system*, and *output channel*. The output of the input channel is described by

$$\mathbf{X} = \mathbf{S} + \mathbf{N}_1$$

where $\mathbf{S}$ is an ideal signal received by the input channel and $\mathbf{N}_1$ is assumed to be the source of all noise in the input. The signal $\mathbf{X}$ is subsequently transformed (recoded) by a linear matrix operator $\mathbf{A}$. It is then transmitted through the optic nerve, or output channel, producing the output $\mathbf{Y}$, as shown by

$$\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{N}_2$$

where $\mathbf{N}_2$ denotes the postencoding intrinsic noise. In the approach taken by Atick and Redlich, it is observed that light signals arriving at the retina contain useful sensory information in a highly redundant form. Moreover, it is hypothesized that the purpose of retinal signal processing is to reduce or eliminate the redundant bits of data due to

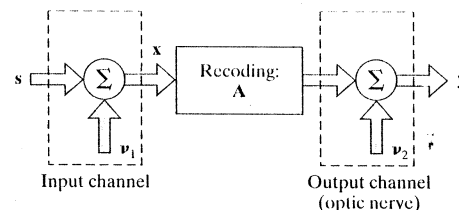


FIGURE 10.5 Model of a perceptual system. The signal vector $\mathbf{s}$ and noise vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ are values of the random vectors $\mathbf{S}$, $\mathbf{N}_1$, and $\mathbf{N}_2$, respectively.

both correlations and noise, before sending the signal along the optic nerve. To quantify this notion, a *redundancy measure* is defined by

$$R = 1 - \frac{I(\mathbf{Y}; \mathbf{S})}{C(\mathbf{Y})} \quad (10.56)$$

where $I(\mathbf{Y}; \mathbf{S})$ is the mutual information between $\mathbf{Y}$ and $\mathbf{S}$, and $C(\mathbf{Y})$ is the channel capacity of the optic nerve (output channel). Equation (10.56) is justified on the grounds that the information the brain is interested in is the ideal signal $\mathbf{S}$, while the physical channel through which this information must pass is in reality the optic nerve. It is assumed that there is no dimensionality reduction in the input-output mapping performed by the perceptual system, which means that $C(\mathbf{Y}) > I(\mathbf{Y}; \mathbf{S})$. The requirement is to find an input-output mapping (i.e., matrix $\mathbf{A}$) that minimizes the redundancy measure R , subject to the constraint of no information loss, as shown by

$$I(\mathbf{Y}; \mathbf{X}) = I(\mathbf{X}; \mathbf{X}) - \epsilon$$

where ϵ is some small positive parameter. The *channel capacity* $C(\mathbf{Y})$ is defined as the maximum rate of information flow possible through the optic nerve, ranging over all probability distributions of inputs applied to it, and keeping the average input power fixed.

When the signal vector $\mathbf{S}$ and the output vector $\mathbf{Y}$ have the same dimensionality and there is noise in the system, the principle of minimum redundancy and the Infomax principle are mathematically equivalent, provided that a similar constraint is imposed on the computational capability of the output neurons in both cases. To be specific, suppose that the channel capacity is measured in terms of the dynamic range of the output of each neuron in the model of Fig. 10.5. Then, according to the principle of minimum redundancy, the quantity to be minimized is

$$1 - \frac{I(\mathbf{Y}; \mathbf{S})}{C(\mathbf{Y})}$$

for a given permissible information loss, and therefore for a given $I(\mathbf{Y}; \mathbf{S})$. Thus the quantity to be minimized is essentially

$$F_1(\mathbf{Y}; \mathbf{S}) = C(\mathbf{Y}) - \lambda I(\mathbf{Y}; \mathbf{S}) \quad (10.57)$$

On the other hand, according to the Infomax principle the quantity to be maximized in the model of Fig. 10.5 is

$$F_2(\mathbf{Y}; \mathbf{S}) = I(\mathbf{Y}; \mathbf{S}) + \lambda C(\mathbf{Y}) \quad (10.58)$$

Although the functions $F_1(\mathbf{Y}; \mathbf{S})$ and $F_2(\mathbf{Y}; \mathbf{S})$ are different, their optimizations yield identical results: They are both formulations of the method of Lagrange multipliers, with the roles of $I(\mathbf{Y}; \mathbf{S})$ and $C(\mathbf{Y})$ being simply interchanged.

The important point to note from this discussion is that despite the difference in formulations, these two information-theoretic principles lead to similar results. In summary, maximization of the mutual information between the output and input of a neural system does indeed lead to redundancy reduction.⁹

0.9 SPATIALLY COHERENT FEATURES

The Infomax principle, as postulated in Section 10.6, applies to a situation where the mutual information $I(\mathbf{Y}; \mathbf{X})$ between the output vector $\mathbf{Y}$ of a neural system and the input vector $\mathbf{X}$ is the objective function to be maximized, as illustrated in Fig. 10.2a. With appropriate changes in terminology, we may extend this principle to deal with the unsupervised processing of the image of a natural scene (Becker and Hinton, 1992). An unprocessed pixel of such an image contains a wealth of information about the scene of interest, albeit in complex form. In particular, the intensity of each pixel is affected by such intrinsic parameters as depth, reflectance, and surface orientation, as well as background noise and illumination. The goal is to design a self-organizing system that is capable of learning to encode this complex information in a simpler form. To be more specific, the objective is to extract higher-order features that exhibit *simple coherence across space* in such a way that the representation of information in one spatially localized region of the image makes it easy to produce the representation of information in neighboring regions; a region refers to a collection of pixels in the image. The situation described herein pertains to the scenario illustrated in Fig. 10.2b.

We may thus formulate the *first* variant of the Infomax principle¹⁰ as follows (Becker, 1996; Becker and Hinton, 1992):

The transformation of a pair of vectors $\mathbf{X}_a$ and $\mathbf{X}_b$ (representing adjacent, nonoverlapping regions of an image by a neural system) should be so chosen that the scalar output Y_a of the system due to the input $\mathbf{X}_a$ maximizes information about the second scalar output Y_b due to $\mathbf{X}_b$. The objective function to be maximized is the mutual information $I(Y_a; Y_b)$ between the outputs Y_a and Y_b .

We refer to this principle as a variant of the Infomax principle in the sense that it is not equivalent to Infomax or derived from it, but certainly functions in a similar spirit.

To be specific, consider Fig. 10.6 that shows two neural networks (modules) a and b receiving inputs $\mathbf{X}_a$ and $\mathbf{X}_b$ from adjacent, nonoverlapping regions of an image. The scalars Y_a and Y_b denote the outputs of these two modules due to the respective input vectors $\mathbf{X}_a$ and $\mathbf{X}_b$. Let S denote a signal component common to both Y_a and Y_b , which

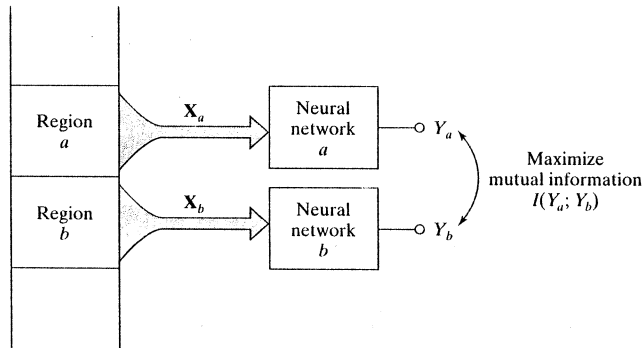


FIGURE 10.6 Processing of two neighboring regions of an image in accordance with the first variant of Infomax.

is representative of the spatial coherence across the two pertinent regions of the original image. We may express Y_a and Y_b as noisy versions of the common signal S , as shown by

$$Y_a = S + N_a \quad (10.59)$$

and

$$Y_b = S + N_b \quad (10.60)$$

The N_a and N_b are additive noise components, assumed to be statistically independent, zero-mean Gaussian-distributed random variables. The signal component S is also assumed to be Gaussian with a distribution of its own. According to Eqs. (10.59) and (10.60), the two modules a and b in Fig. 10.6 make consistent assumptions about each other.

Using the last line of Eq. (10.30), the mutual information between Y_a and Y_b is defined by

$$I(Y_a; Y_b) = h(Y_a) + h(Y_b) - h(Y_a, Y_b) \quad (10.61)$$

According to the formula of Eq. (10.22) for the differential entropy of a Gaussian random variable, the differential entropy $h(Y_a)$ of Y_a is given by

$$h(Y_a) = \frac{1}{2} \left[1 + \log(2\pi\sigma_a^2) \right] \quad (10.62)$$

where σ_a^2 is the variance of Y_a . Similarly, the differential entropy of Y_b is given by

$$h(Y_b) = \frac{1}{2} \left[1 + \log(2\pi\sigma_b^2) \right] \quad (10.63)$$

where σ_b^2 is the variance of Y_b . As for the joint differential entropy $h(Y_a, Y_b)$, we use the formula of Eq. (10.24) to write

$$h(Y_a, Y_b) = 1 + \log(2\pi) + \frac{1}{2} \log|\det(\Sigma)| \quad (10.64)$$

The 2-by-2 matrix Σ is the covariance matrix of Y_a and Y_b ; it is defined by

$$\Sigma = \begin{bmatrix} \sigma_a^2 & \rho_{ab}\sigma_a\sigma_b \\ \rho_{ab}\sigma_a\sigma_b & \sigma_b^2 \end{bmatrix} \quad (10.65)$$

where ρ_{ab} is the *correlation coefficient* of Y_a and Y_b ; that is,

$$\rho_{ab} = \frac{E[(Y_a - E[Y_a])(Y_b - E[Y_b])]}{\sigma_a\sigma_b} \quad (10.66)$$

Hence the determinant of Σ is

$$\det(\Sigma) = \sigma_a^2\sigma_b^2(1 - \rho_{ab}^2) \quad (10.67)$$

and so we may rewrite Eq. (10.64) as

$$h(Y_a, Y_b) = 1 + \log(2\pi) + \frac{1}{2} \log[\sigma_a^2\sigma_b^2(1 - \rho_{ab}^2)] \quad (10.68)$$

By substituting Eqs. (10.62), (10.63), and (10.68) in (10.61), and then simplifying terms, we get

$$I(Y_a; Y_b) = -\frac{1}{2} \log(1 - \rho_{ab}^2) \quad (10.69)$$

From Eq. (10.69) we immediately deduce that maximizing the mutual information $I(Y_a; Y_b)$ is equivalent to maximizing the correlation coefficient ρ_{ab} , which is intuitively satisfying. Note that, by definition, $|\rho_{ab}| \leq 1$.

Maximizing the mutual information $I(Y_a; Y_b)$ may be viewed as the nonlinear generalization of canonical correlation in statistics (Becker and Hinton, 1992). Given two input vectors (stimuli) $\mathbf{X}_a$ and $\mathbf{X}_b$ (not necessarily of the same dimensionality), and two corresponding weight vectors, $\mathbf{w}_a$ and $\mathbf{w}_b$, the objective of *canonical correlation analysis* is to find linear combinations $Y_a = \mathbf{w}_a^T \mathbf{X}_a$ and $Y_b = \mathbf{w}_b^T \mathbf{X}_b$ that have maximum correlation between them (Anderson, 1984). Maximizing $I(Y_a; Y_b)$ is a nonlinear generalization of canonical correlation by virtue of the nonlinearity built into the design of the neural modules in Fig. 10.6.

In Becker and Hinton (1992), it is demonstrated that by maximizing the mutual information $I(Y_a; Y_b)$ it is possible to extract stereo disparity (depth) from random dot stereograms. This is a difficult feature extraction problem that could not be solved by a one-layer or linear neural network.

0.10 SPATIALLY INCOHERENT FEATURES

The unsupervised processing of an image considered in the previous section deals with the extraction of spatially coherent features from an image. We now consider the opposite of the situation described therein. To be specific, consider Fig. 10.2c, where the objective is to enhance the *spatial differences* between a pair of corresponding regions derived from two separate images. Whereas the mutual information between the outputs of the modules is maximized in Fig. 10.2b, we do the exact opposite in Fig. 10.2c.

We may thus state the *second* variant of the Infomax principle as follows (Ukrainec and Haykin, 1992, 1996):

The transformation of a pair of input vectors, $\mathbf{X}_a$ and $\mathbf{X}_b$, representing data derived from corresponding regions in a pair of separate images, by a neural system should be so chosen that the scalar output Y_a of the system due to the input $\mathbf{X}_a$ minimizes information about the second scalar output Y_b due to $\mathbf{X}_b$. The objective function to be minimized is the mutual information $I(Y_a; Y_b)$ between the outputs Y_a and Y_b .

Here again we refer to this principle as a variant of the Infomax principle in that it is not equivalent to Infomax or derived from it, but certainly functions in the spirit of it.¹¹

The second variant of the Infomax principle finds application in *radar polarimetry*, for example, where a surveillance radar system produces a pair (or more) of images of an environment of interest by transmitting on one polarization and receiving the backscatter from the environment on the same or different polarization. The polarization can be vertical or horizontal. For example, we may have a pair of radar images, one image representing like polarization (horizontal–horizontal, say), and the other image representing cross-polarization (horizontal on transmit and vertical on receive). Such

an application is described in Ukrainec and Haykin (1992, 1996), which pertains to the *enhancement of a polarization target* in a dual-polarized radar system. The sample radar scene used in the study is described as follows. An incoherent radar transmits in a horizontally polarized fashion, and receives radar returns on both horizontal and vertical polarization channels. The target of interest is a *cooperative, polarization-twisting reflector* designed to rotate the incident polarization through 90 degrees. In the normal operation of a radar system, the detection of such a target is made difficult by imperfections in the system as well as reflections from unwanted polarimetric targets on the ground (i.e., radar clutter). We perceive that a nonlinear mapping is needed to account for the non-Gaussian distribution common to radar returns. The target enhancement problem is cast as a variational problem involving the minimization of a quadratic cost functional with constraints. The net result is a processed cross-polarized image that exhibits a significant improvement in target visibility, far more pronounced than that attainable through the use of a linear technique such as principal components analysis. The model used by Ukrainec and Haykin assumes Gaussian statistics for the transformed data, since a model-free estimate of the probability density function is a computationally challenging task. The mutual information between two Gaussian variables Y_a and Y_b is defined by Eq. (10.61). To learn the synaptic weights of the two modules, a variational approach is taken. The requirement is to suppress the radar clutter that is common to the horizontally polarized and vertically polarized radar images. To satisfy this requirement, the mutual information $I(Y_a; Y_b)$ is minimized, subject to a constraint imposed on the synaptic weights as shown by

$$P = (\text{tr}[\mathbf{W}^T \mathbf{W}] - 1)^2 \quad (10.70)$$

where $\mathbf{W}$ is the overall weight matrix of the network, and $\text{tr}[\cdot]$ is the trace of the enclosed matrix. A stationary point is reached when we have

$$\nabla_{\mathbf{W}} I(Y_a; Y_b) + \lambda \nabla_{\mathbf{W}} P = 0 \quad (10.71)$$

where λ is the Lagrange multiplier. A quasi-Newton optimization routine was used to find the minimum; quasi-Newton's methods are discussed in Chapter 4.

Figure 10.7 shows the architecture of the neural network used in Ukrainec and Haykin (1992, 1996). A Gaussian radial-basis function (RBF) network was chosen for each of the two modules because it has the advantage of providing a set of fixed basis-functions (i.e., a nonadaptive hidden layer). The input data are expanded onto the basis functions and then combined using layers of *linear* weights; the dashed lines shown in Fig. 10.7 represent the cross-coupling connections between the two modules. The centers of the Gaussian functions were chosen at evenly spaced intervals to cover the entire input domain, and their widths were chosen using a heuristic. Figure 10.8a shows the raw horizontally polarized and vertically polarized (both on receive) radar images of a parklike setting on the shore of Lake Ontario. The range coordinate is along the horizontal axis of each image, increasing from left to right; the azimuth coordinate is along the vertical axis, increasing down the image. Figure 10.8b shows the combined image obtained by minimizing the mutual information between the horizontally and vertically polarized radar images, as just described. The bright spot clearly visible in this image corresponds to the radar return from a cooperative, polarization-twisting reflector placed along the lake shore. The clutter-suppression performance of

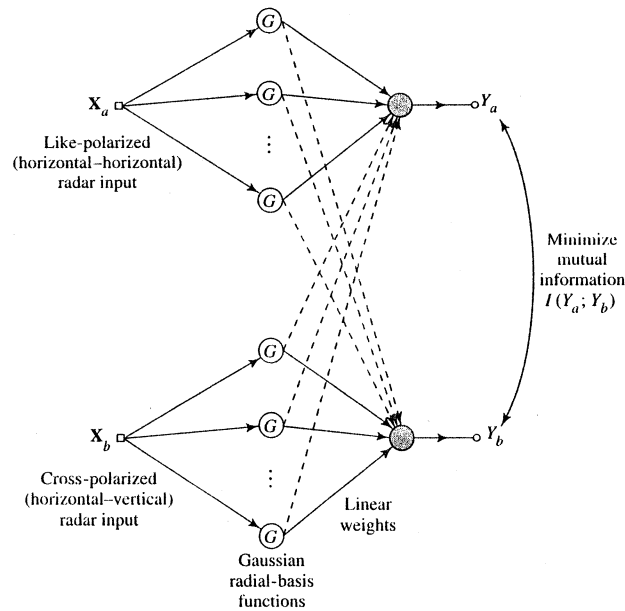


FIGURE 10.7 Block diagram of a neural processor, the goal of which is to suppress background clutter using a pair of polarimetric, noncoherent radar inputs; clutter suppression is attained by minimizing the mutual information between the outputs of the two modules.

the information-theoretic model described here exceeds that of commonly used projections using principal components analysis (Ukrainec and Haykin, 1992, 1996).¹²

10.11 INDEPENDENT COMPONENTS ANALYSIS

We now turn our attention to the last scenario described in Fig. 10.2d. To add more specificity to the signal processing problem stated therein, consider the block diagram in Fig. 10.9. The operation starts with a random source vector $\mathbf{U}(n)$ defined by

$$\mathbf{U} = [U_1, U_2, \dots, U_m]^T$$

where the m components are supplied by a set of *independent sources*. Temporal sequences are considered here; henceforth the argument n denotes discrete time. The vector $\mathbf{U}$ is applied to a linear system whose input-output characterization is defined by a nonsingular m -by- m matrix $\mathbf{A}$, called the *mixing matrix*. The result is an m -by-1 observation vector $\mathbf{X}(n)$ related to $\mathbf{U}(n)$ as follows (see Fig. 10.10a)

$$\mathbf{X} = \mathbf{A}\mathbf{U} \quad (10.72)$$

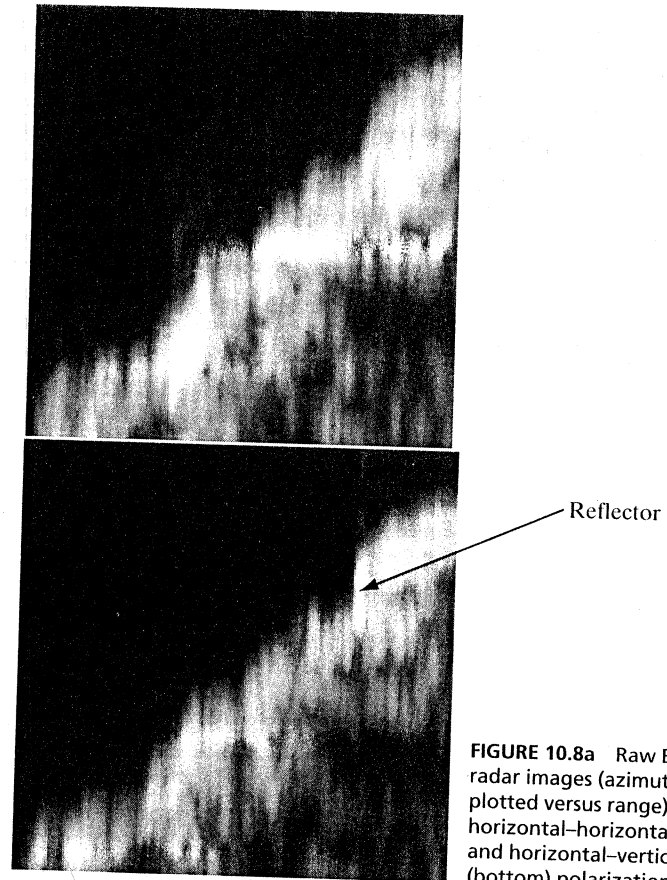


FIGURE 10.8a Raw B-scan radar images (azimuth plotted versus range) for horizontal-horizontal (top) and horizontal-vertical (bottom) polarizations.

where

$$\mathbf{X} = [X_1, X_2, \dots, X_m]^T$$

The source vector $\mathbf{U}$ and the mixing matrix $\mathbf{A}$ are both unknown. The only thing available to us is the observation vector $\mathbf{X}$. Given $\mathbf{X}$, the problem is to find a *demixing matrix* $\mathbf{W}$ such that the original source vector $\mathbf{U}$ can be recovered from the output vector $\mathbf{Y}$ defined by (see Fig. 10.10b)

$$\mathbf{Y} = \mathbf{W}\mathbf{X} \quad (10.73)$$

where

$$\mathbf{Y} = [Y_1, Y_2, \dots, Y_m]^T$$

FIGURE 10.8b Composite image computed by minimizing the mutual information between the two polarized radar images of Fig. 10.8a.

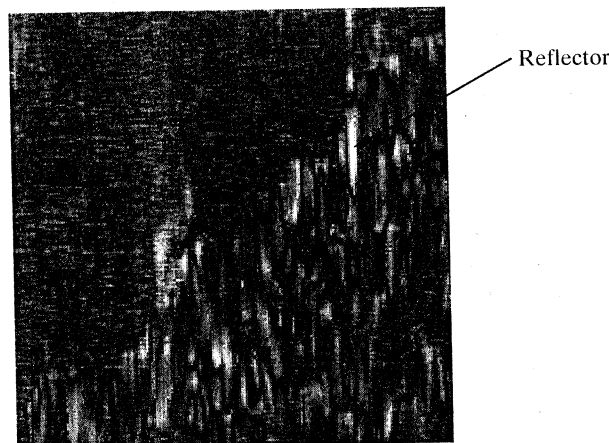


FIGURE 10.9 Block diagram of processor for the blind source separation problem. The vectors $\mathbf{u}$, $\mathbf{x}$, and $\mathbf{y}$ are values of the respective random vectors $\mathbf{U}$, $\mathbf{X}$, and $\mathbf{Y}$.

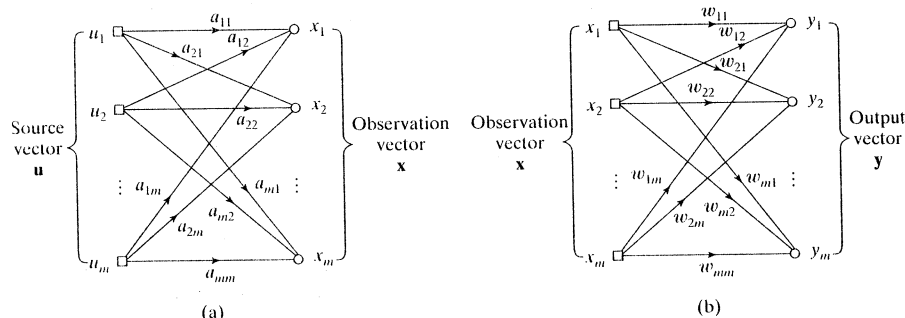
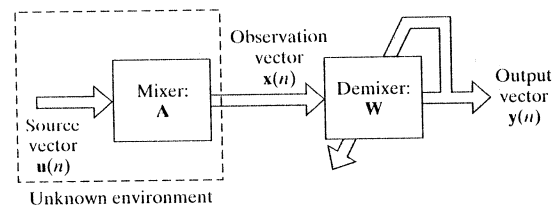


FIGURE 10.10 Detailed description of (a) mixing matrix and (b) demixing matrix.

It is ordinarily assumed that the source signals $U_1, U_2, \dots, U_m$ are zero-mean signals, which in turn means that the observables $X_1, X_2, \dots, X_m$ are also zero-mean signals. The same is true for the demixer outputs $Y_1, Y_2, \dots, Y_m$.

We may thus state the *blind source separation problem* as follows:

Given N independent realizations of the observation vector $\mathbf{X}$, find an estimate of the inverse of the mixing matrix $\mathbf{A}$.

Source separation exploits primarily *spatial diversity* in that the different sensors providing realizations of the vector $\mathbf{X}$ carry different mixtures of the sources. Spectral diversity, if it exists, can also be exploited, but the fundamental approach to source separation is essentially *spatial*: looking for structure across the sensors, not across time (Cardoso, 1998a).

The solution to the blind source separation problem is feasible, except for an arbitrary scaling of each signal component and permutation of indices. In other words, it is possible to find a demixing matrix $\mathbf{W}$ whose individual rows are a rescaling and permutation of those of the mixing matrix $\mathbf{A}$. That is, the solution may be expressed in the form

$$\mathbf{Y} = \mathbf{W}\mathbf{X} = \mathbf{W}\mathbf{A}\mathbf{U} \rightarrow \mathbf{D}\mathbf{P}\mathbf{U}$$

where $\mathbf{D}$ is a nonsingular diagonal matrix and $\mathbf{P}$ is a permutation matrix.

The problem described herein is commonly referred to as the blind (signal) source separation problem,¹³ where the term “blind” is used to signify the fact that the only information used to recover the original signal sources is contained in a realization of the observation vector $\mathbf{X}$, denoted by $\mathbf{x}$. The underlying principle involved in its solution is called *independent components analysis* (ICA) (Comon, 1994), which may be viewed as an extension of principal components analysis (PCA). Whereas PCA can only impose independence up to the second order while constraining the direction vectors to be orthogonal, ICA imposes *statistical independence* on the individual components of the output vector $\mathbf{Y}$ and has no orthogonality constraint. Note also that in practice, an algorithmic implementation of independent components analysis can only go for “as statistically independent as possible.”

The need for blind source separation arises in many diverse applications that include the following:

- **Speech separation.** In this application, the vector $\mathbf{x}$ consists of several speech signals that have been linearly mixed together, and the requirement is to separate them (Bell and Sejnowski, 1995). A difficult form of this situation, for example, arises in a teleconferencing environment.
- **Array antenna processing.** In this second application, the vector $\mathbf{x}$ represents the output of a radar array antenna produced by several incident narrowband signals originating from sources of unknown directions (Cardoso and Souloumnia, 1993; Swindlehurst et al., 1997). Here again the requirement is to separate the source signals. (By a narrowband signal we mean a band-pass signal whose bandwidth is small compared to the carrier frequency.)
- **Multisensor biomedical records.** In this third application, the vector $\mathbf{x}$ consists of recordings made by a multitude of sensors used to monitor biological signals of interest. For example, the requirement may be that of separating the heartbeat of a fetus from that of the mother (Cardoso, 1998b).
- **Financial market data analysis.** In this application the vector $\mathbf{x}$ consists of a set of different stock market data and the requirement is to extract the underlying set of dominant independent components (Back and Weigend, 1998).

In these applications, the blind source separation problem may be compounded by the possible presence of unknown propagation delays, extensive filtering imposed on the sources by their environments, and unavoidable contamination of the observation vector $\mathbf{x}$ by noise. These impairments mean that (unfortunately) the idealized form of instantaneous mixing of signals described in Eq. (10.72) is very rarely encountered in real-world situations. In what follows, however, we will ignore these impairments in order to develop insight into the fundamental aspects of the blind source separation problem.

Criterion for Statistical Independence

With statistical independence as the property desired from the components of the output vector $\mathbf{Y}$ for blind source separation, what practical measure can we use for it? One obvious possibility is to choose the mutual information $I(Y_i; Y_j)$ between the random variables Y_i and Y_j constituting any two components of the output vector $\mathbf{Y}$. When, in the ideal case, $I(Y_i; Y_j)$ is zero, the components Y_i and Y_j are statistically independent. This would therefore suggest minimizing the mutual information between every pair of the random variables constituting the output vector $\mathbf{Y}$. This objective is equivalent to minimizing the Kullback–Leibler divergence between the following two distributions: (1) the probability density function $f_{\mathbf{Y}}(\mathbf{y}, \mathbf{W})$ parameterized by $\mathbf{W}$, and (2) the corresponding factorial distribution defined by

$$\tilde{f}_{\mathbf{Y}}(\mathbf{y}, \mathbf{W}) = \prod_{i=1}^m \tilde{f}_{Y_i}(y_i, \mathbf{W}) \quad (10.74)$$

where $\tilde{f}_{Y_i}(y_i, \mathbf{W})$ is the marginal probability density function of Y_i . In effect, Eq. (10.74) may be viewed as a *constraint* imposed on the learning algorithm, forcing it to contrast $f_{\mathbf{Y}}(\mathbf{y}, \mathbf{W})$ against the factorial distribution $\tilde{f}_{\mathbf{Y}}(\mathbf{y}, \mathbf{W})$. We may thus state the third variant to the Infomax principle for independent components analysis as (Comon, 1994):

Given an m -by-1 vector $\mathbf{X}$ representing a linear combination of m independent source signals, the transformation of the observation vector $\mathbf{X}$ by a neural system into a new vector $\mathbf{Y}$ should be carried out in such a way that the Kullback–Leibler divergence between the parameterized probability denoting function $f_{\mathbf{Y}}(\mathbf{y}, \mathbf{W})$ and the corresponding factorial distribution $\tilde{f}_{\mathbf{Y}}(\mathbf{y}, \mathbf{W})$ is minimized with respect to the unknown parameter matrix $\mathbf{W}$.

The Kullback–Leibler divergence for the problem described herein is considered in Section 10.5. The formula we are seeking is given in Eq. (10.44). Adapting that formula to our present situation, we may express the Kullback–Leibler divergence between the probability density functions $f_{\mathbf{Y}}(\mathbf{y}, \mathbf{W})$ and $\tilde{f}_{\mathbf{Y}}(\mathbf{y}, \mathbf{W})$ as follows:

$$D_{f\|\tilde{f}}(\mathbf{W}) = -h(\mathbf{Y}) + \sum_{i=1}^m \tilde{h}(Y_i) \quad (10.75)$$

where $h(\mathbf{Y})$ is the entropy of random vector $\mathbf{Y}$ at the output of the demixer and $\tilde{h}(Y_i)$ is the marginal entropy of the i th element of $\mathbf{Y}$. The Kullback–Leibler divergence $D_{f\|\tilde{f}}$ is the objective (contrast) function that we focus on henceforth for solving the blind source separation problem.

Determination of the Differential Entropy $h(\mathbf{Y})$

The output vector $\mathbf{Y}$ is related to the input vector $\mathbf{X}$ by Eq. (10.73), where $\mathbf{W}$ is the demixing matrix. In light of Eq. (10.18) we may express the differential entropy of $\mathbf{Y}$ as:

$$\begin{aligned} h(\mathbf{Y}) &= h(\mathbf{W}\mathbf{X}) \\ &= h(\mathbf{X}) + \log|\det(\mathbf{W})| \end{aligned} \quad (10.76)$$

where $\det(\mathbf{W})$ is the determinant of $\mathbf{W}$.

Determination of the Marginal Entropy $\tilde{h}(Y_i)$

To determine the Kullback–Leibler divergence $D_{f\|\tilde{f}}$, we also need to know the marginal entropy $\tilde{h}(Y_i)$. To determine $\tilde{h}(Y_i)$ we require knowledge of the marginal distribution of Y_i , which in turn requires integrating out the effects of all the components of the random vector $\mathbf{Y}$ except for the i th component. For a vector $\mathbf{Y}$ of high dimensionality it is usually more difficult to calculate $\tilde{h}(Y_i)$ than $h(\mathbf{Y})$. We may overcome this difficulty by deriving an approximate formula for $\tilde{h}(Y_i)$ in terms of the higher-order moments of the random variable Y_i . This is accomplished by properly truncating one of two expansions:

- Edgeworth series (Comon, 1991)
- Gram–Charlier series (Amari et al., 1996)

In this chapter we follow the latter approach. An exposition of the Gram–Charlier series is presented in note 14. A brief description of the Edgeworth series is also presented in that note.

To be specific, the Gram–Charlier expansion of the parameterized marginal probability density function $\tilde{f}_{Y_i}(y_i, \mathbf{W})$ is described by

$$\tilde{f}_{Y_i}(y_i, \mathbf{W}) = \alpha(y_i) \left[1 + \sum_{k=3}^{\infty} c_k H_k(y_i) \right] \quad (10.77)$$

where the various terms are defined as follows:

1. The multiplying factor $\alpha(y_i)$ is the probability density function of a normalized Gaussian random variable with zero mean and unit variance; that is,

$$\alpha(y_i) = \frac{1}{\sqrt{2\pi}} e^{-y_i^2/2}$$

2. The $H_k(y_i)$ are *Hermite polynomials*.
3. The coefficients of the expansion, $\{c_k : k = 3, 4, \dots\}$, are defined in terms of the cumulants of the random variable Y_i .

The natural order of the terms in Eq. (10.77) is *not* the best for the Gram–Charlier series. Rather, the terms listed here in the parentheses should be grouped together (Helstrom, 1968):

$$k = (0), (3), (4, 6), (5, 7, 9), \dots$$

For the blind-source separation problem, the approximation of the marginal probability density function $\tilde{f}_{Y_i}(y_i)$ by truncating the Gram–Charlier series at $k = (4, 6)$ is considered to be adequate. We may thus write

$$\tilde{f}_{Y_i}(y_i) \approx \alpha(y_i) \left(1 + \frac{\kappa_{i,3}}{3!} H_3(y_i) + \frac{\kappa_{i,2}^2}{4!} H_4(y_i) + \frac{(\kappa_{i,6} + 10\kappa_{i,3}^2)}{6!} H_6(y_i) \right) \quad (10.78)$$

where $\kappa_{i,k}$ is the k th order cumulant of Y_i . Let $m_{i,k}$ denote the k th-order moment of Y_i defined by

$$\begin{aligned} m_{i,k} &= E[Y_i^k] \\ &= E\left[\left(\sum_{k=1}^m w_{ik} X_i\right)^k\right] \end{aligned} \quad (10.79)$$

where X_i is the i th element of observation vector $\mathbf{X}$ and w_{ik} is the ik -th element of weight matrix $\mathbf{W}$. Earlier we justified the zero-mean assumption of Y_i for all i . Accordingly, we have $\sigma_i^2 = m_{i,2}$ (i.e., the variance and the mean-square value are equal) and so relate the cumulants of Y_i to its moments as follows:

$$\kappa_{i,3} = m_{i,3} \quad (10.80)$$

$$\kappa_{i,4} = m_{i,4} - 3m_{i,2}^2 \quad (10.81)$$

$$\kappa_{i,6} = m_{i,6} - 10m_{i,3}^2 - 15m_{i,2}m_{i,4} + 30m_{i,2}^3 \quad (10.82)$$

The algorithm of $\tilde{f}_{Y_i}(y_i)$, using the approximation of Eq. (10.78), is given by

$$\log \tilde{f}_{Y_i}(y_i) \approx \log \alpha(y_i) + \log \left(1 + \frac{\kappa_{i,3}}{3!} H_3(y_i) + \frac{\kappa_{i,2}^2}{4!} H_4(y_i) + \frac{(\kappa_{i,6} + 10\kappa_{i,3}^2)}{6!} H_6(y_i) \right) \quad (10.83)$$

To proceed further, we use the expansion of a logarithm:

$$\log(1 + y) \approx y - \frac{y^2}{2} \quad (10.84)$$

where all the terms of order three and higher are ignored.

From our previous discussion we recall that the formula for the marginal entropy of Y_i is (see Eq. (10.43))

$$\tilde{h}(Y_i) = - \int_{-\infty}^{\infty} f_{Y_i}(y_i) \log f_{Y_i}(y_i) dy_i, \quad i = 1, 2, \dots, m$$

where m is the number of sources. By making use of the approximations described in Eqs. (10.78), (10.83) and (10.84), and invoking certain integrals that involve the normalized Gaussian density $\alpha(y)$ and various Hermite polynomials $H_k(y)$, we obtain the following approximate formula for the marginal entropy (Madhuranath and Haykin, 1998):

$$\begin{aligned} \tilde{h}(Y_i) &= \frac{1}{2} \log(2\pi e) - \frac{\kappa_{i,3}^2}{12} - \frac{\kappa_{i,4}^2}{48} - \frac{(\kappa_{i,6} + 10\kappa_{i,3}^2)^2}{1440} \\ &\quad + \frac{3}{8} \kappa_{i,3}^2 \kappa_{i,4} + \frac{\kappa_{i,3}^2(\kappa_{i,6} + 10\kappa_{i,3}^2)}{24} + \frac{\kappa_{i,4}^2(\kappa_{i,6} + 10\kappa_{i,3}^2)}{24} \\ &\quad + \frac{\kappa_{i,4}(\kappa_{i,6} + 10\kappa_{i,3}^2)^2}{64} + \frac{\kappa_{i,4}^3}{16} + \frac{(\kappa_{i,6} + 10\kappa_{i,3}^2)^3}{432} \end{aligned} \quad (10.85)$$

By substituting Eqs. (10.76) and (10.85) in (10.75), we get the Kullback–Leibler divergence for the problem at hand:

$$\begin{aligned} D_{f\|f}(\mathbf{W}) &\approx -h(\mathbf{X}) - \log|\det(\mathbf{W})| + \frac{m}{2} \log(2\pi e) \\ &\quad - \sum_{i=1}^m \left(\frac{\kappa_{i,3}^2}{12} + \frac{\kappa_{i,4}^2}{48} + \frac{(\kappa_{i,6} + 10\kappa_{i,3}^2)^2}{1440} - \frac{3}{8} \kappa_{i,3}^2 \kappa_{i,4} \right. \\ &\quad - \frac{\kappa_{i,3}^2(\kappa_{i,6} + 10\kappa_{i,3}^2)}{24} - \frac{\kappa_{i,4}^2(\kappa_{i,6} + 10\kappa_{i,3}^2)}{24} \\ &\quad \left. - \frac{\kappa_{i,4}(\kappa_{i,6} + 10\kappa_{i,3}^2)^2}{64} - \frac{\kappa_{i,4}^3}{16} - \frac{(\kappa_{i,6} + 10\kappa_{i,3}^2)^3}{432} \right) \end{aligned} \quad (10.86)$$

where the cumulants are all functions of the weight matrix $\mathbf{W}$.

Activation Function

To evaluate the Kullback–Leibler divergence described in Eq. (10.86), we need an adaptive procedure for the computation of the higher-order cumulants for the observation vector $\mathbf{x}$. The question is: How do we proceed with this computation, bearing in mind the way in which the approximate formula of Eq. (10.86) is derived? Recall that the derivation is based on the Gram–Charlier expansion, assuming that the random variable Y_i has zero mean and unit variance. We previously justified the zero-mean assumption on the grounds that, to begin with, the source signals typically have zero mean. As for the unit-variance assumption, we may deal with it by taking one of two approaches:

1. Constrained approach. In this approach, the unit-variance assumption is imposed on the computation of the higher-order cumulants $\kappa_{i,3}$, $\kappa_{i,4}$, and $\kappa_{i,6}$ for all i (Amari et al., 1996). Unfortunately, there is no guarantee that the variance of Y_i , namely σ_i^2 , remains constant, let alone equal to 1, during the computation. From the defining equations (10.81) and (10.82), we note that both $\kappa_{i,4}$ and $\kappa_{i,6}$ depend on $\sigma_i^2 = m_{i,2}$. The result of assuming $\sigma_i^2 = 1$ is that the estimates derived for $\kappa_{i,4}$ and $\kappa_{i,6}$ are highly biased and therefore erroneous relative to the estimate of $\kappa_{i,3}$.

2. Unconstrained approach. In this alternative approach, the variance σ_i^2 is treated as an unknown time-varying parameter, which is how it actually is in practice (Madhuranath and Haykin, 1998). The effect of deviation in the value of σ_i^2 from 1 is viewed as a scaling variation in the value of random variable Y_i . Most importantly, the estimates derived for $\kappa_{i,4}$ and $\kappa_{i,6}$ account for the variation of σ_i^2 with time. A proper relationship between the estimates of all three higher-order cumulants in Eq. (10.86) is thereby maintained.

An experimental study of blind source separation reported in Madhuranath and Haykin (1998) shows that the unconstrained approach yields a superior performance compared to the constrained approach. In what follows, we follow the unconstrained approach.

To develop a learning algorithm for computing $\mathbf{W}$, we need to differentiate Eq. (10.86) with respect to $\mathbf{W}$ and thereby formulate an activation function for the algorithm.

Let A_{ik} denote the ik -th cofactor of matrix $\mathbf{W}$. Using Laplace's expansion of $\det(\mathbf{W})$ by the i th row, we may then write (Wylie and Barrett, 1982)

$$\det(\mathbf{W}) = \sum_{k=1}^m w_{ik} A_{ik}, \quad i = 1, 2, \dots, m \quad (10.87)$$

where w_{ik} is the ik -th element of matrix $\mathbf{W}$. Hence, differentiating the logarithm of $\det(\mathbf{W})$ with respect to w_{ik} , we get

$$\begin{aligned} \frac{\partial}{\partial w_{ik}} \log(\det(\mathbf{W})) &= \frac{1}{\det(\mathbf{W})} \frac{\partial}{\partial w_{ik}} \det(\mathbf{W}) \\ &= \frac{A_{ik}}{\det(\mathbf{W})} \\ &= (\mathbf{W}^{-T})_{ik} \end{aligned} \quad (10.88)$$

where $\mathbf{W}^{-T}$ is the inverse of the transposed matrix $\mathbf{W}^T$. The partial derivatives of the other terms (that depend on $\mathbf{W}$) in Eq. (10.86) with respect to w_{ik} are (see Eqs. (10.80) through (10.82))

$$\begin{aligned} \frac{\partial \kappa_{i,3}}{\partial w_{ik}} &= 3 E[Y_i^2 X_k] \\ \frac{\partial \kappa_{i,4}}{\partial w_{ik}} &= 4 E[Y_i^3 X_k] - 12 m_{i,2} E[Y_i X_k] \\ \frac{\partial}{\partial w_{ik}} (\kappa_{i,6} + 10 \kappa_{i,3}^2) &= 6 E[Y_i^5 X_k] - 30 m_{i,4} E[Y_i X_k] \\ &\quad - 60 m_{i,2} E[Y_i^3 X_k] + 180 m_{i,2}^2 E[Y_i X_k] \end{aligned}$$

In deriving an adaptive algorithm the usual approach is to replace expectations with their instantaneous values. Hence, by doing this replacement in these three equations, we get the following approximate results:

$$\frac{\partial \kappa_{i,3}}{\partial w_{ik}} \approx 3 y_i^2 x_k \quad (10.89)$$

$$\frac{\partial \kappa_{i,4}}{\partial w_{ik}} \approx -8 y_i^3 x_k \quad (10.90)$$

$$\frac{\partial}{\partial w_{ik}} (\kappa_{i,6} + 10 \kappa_{i,3}^2) \approx 96 y_i^5 x_k \quad (10.91)$$

Substituting Eqs. (10.88) through (10.91) in the expression for the derivative of Eq. (10.86) with respect to w_{ik} yields

$$\frac{\partial}{\partial w_{ik}} D_{f\|f}(\mathbf{W}) \approx -(\mathbf{W}^{-T})_{ik} + \varphi(y_i) x_k \quad (10.92)$$

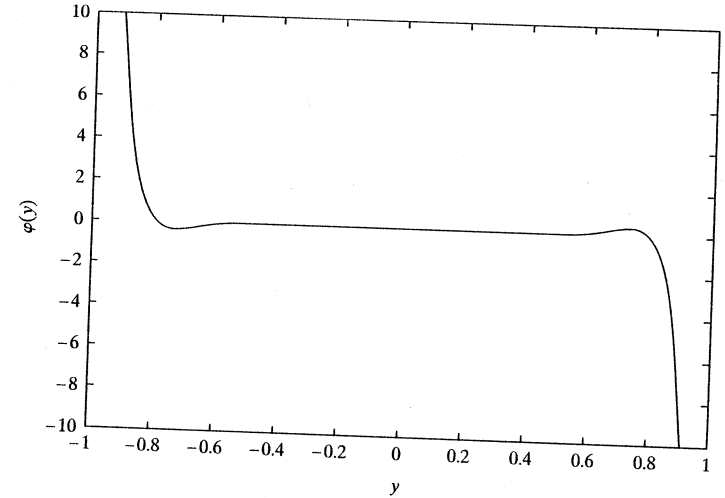


FIGURE 10.11 Activation function $\varphi(y)$ of Eq. (10.93).

where $\varphi(y_i)$ is the nonmonotonic activation function of the learning algorithm, defined by (Madhuranath and Haykin, 1998)

$$\varphi(y_i) = \frac{1}{2} y_i^5 + \frac{2}{3} y_i^7 + \frac{15}{2} y_i^9 + \frac{2}{15} y_i^{11} - \frac{112}{3} y_i^{13} + 128 y_i^{15} - \frac{512}{3} y_i^{17} \quad (10.93)$$

Figure 10.11 plots the activation function $\varphi(y_i)$ versus y_i for $-1 < y_i < 1$. This covers the range of values of the demixer output y_i to which the operation of the learning algorithm is ordinarily confined. It is noteworthy that the slope of the activation function is positive in the interval $(-0.734, 0.734)$; this is a requirement for stability of the algorithm as discussed later in the section.

Learning Algorithm for ICA

The objective of the learning algorithm is to minimize the Kullback–Leibler divergence between the probability density function of $\mathbf{Y}$ and the factorial distribution of Y_i for $i = 1, 2, \dots, m$. This minimization may be implemented using the method of gradient descent whereby the adjustment applied to weight w_{ik} is defined by

$$\begin{aligned} \Delta w_{ik} &= -\eta \frac{\partial}{\partial w_{ik}} D_{f\|f} \\ &= \eta ((\mathbf{W}^{-T})_{ik} - \varphi(y_i) x_k) \end{aligned} \quad (10.94)$$

where η is a learning-rate parameter.

Extending the formula of Eq. (10.94) to the entire weight matrix $\mathbf{W}$ of the demixer, we may express the adjustment $\Delta \mathbf{W}$ applied to $\mathbf{W}$ as follows:

$$\Delta \mathbf{W} = \eta (\mathbf{W}^{-T} - \varphi(\mathbf{y}) \mathbf{x}^T) \quad (10.95)$$

where $\mathbf{x}^T$ is the transpose of the m -by-1 observation vector $\mathbf{x}$, and

$$\boldsymbol{\varphi}(\mathbf{y}) = [\varphi(y_1), \varphi(y_2), \dots, \varphi(y_m)]^T \quad (10.96)$$

The formula for $\Delta \mathbf{W}$ given in Eq. (10.95) can be rewritten by noting that

$$\mathbf{y}^T = \mathbf{x}^T \mathbf{W}^T$$

and thereby reformulating it in the equivalent form

$$\begin{aligned} \Delta \mathbf{W} &= \eta [\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y}) \mathbf{x}^T \mathbf{W}^T] \mathbf{W}^{-T} \\ &= \eta [\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y}) \mathbf{y}^T] \mathbf{W}^{-T} \end{aligned} \quad (10.97)$$

where $\mathbf{I}$ is the identity matrix. The update rule for adapting the demixing matrix takes the form

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \eta(n) [\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y}(n)) \mathbf{y}^T(n)] \mathbf{W}^{-T}(n) \quad (10.98)$$

where the parameters are all shown in their time varying forms.

Equivariant Property

The purpose of a blind source separation algorithm is to update the demixing matrix $\mathbf{W}(n)$ such that the output vector

$$\mathbf{y}(n) = \mathbf{W}(n) \mathbf{x}(n) = \mathbf{W}(n) \mathbf{A} \mathbf{u}(n)$$

is as close as possible to the original source vector $\mathbf{u}(n)$ in some statistical sense. To be more specific, consider a global system characterized by a matrix $\mathbf{C}(n)$ that is obtained by multiplying the mixing matrix $\mathbf{A}$ and demixing matrix $\mathbf{W}(n)$ as shown by

$$\mathbf{C}(n) = \mathbf{W}(n) \mathbf{A} \quad (10.99)$$

Ideally, this global system would satisfy two conditions:

1. The algorithm responsible for adjusting $\mathbf{C}(n)$ converges to an optimum value equal to the permutation matrix.
2. The algorithm is itself described by

$$\mathbf{C}(n+1) = \mathbf{C}(n) + \eta(n) \mathbf{G}(\mathbf{C}(n) \mathbf{u}(n)) \mathbf{C}(n) \quad (10.100)$$

where $\mathbf{G}(\mathbf{C}(n) \mathbf{u}(n))$ is a vector-valued function of $\mathbf{C}(n) \mathbf{u}(n)$. The performance of the algorithm is completely characterized by the system matrix $\mathbf{C}(n)$ and *not* by the individual values of the mixing matrix $\mathbf{A}$ and demixing matrix $\mathbf{W}(n)$. Such an adaptive system is said to be *equivariant* (Cardoso and Laheld, 1996).

The adaptive algorithm of Eq. (10.98) is certainly capable of approximately satisfying the first condition. However, as it stands, it cannot satisfy the second condition. To appreciate this point, we may rewrite Eq. (10.98) in the equivalent form

$$\mathbf{C}(n+1) = \mathbf{C}(n) + \eta(n) \mathbf{G}(\mathbf{C}(n) \mathbf{u}(n)) \mathbf{W}^{-T}(n) \mathbf{A} \quad (10.101)$$

where

$$\mathbf{G}(\mathbf{C}(n) \mathbf{u}(n)) = \mathbf{I} - \boldsymbol{\varphi}(\mathbf{C}(n) \mathbf{u}(n)) (\mathbf{C}(n) \mathbf{u}(n))^T \quad (10.102)$$

The algorithm of Eq. (10.98) falls short of the equivariant condition described in Eq. (10.100) in that the vector-valued function $\mathbf{G}(\mathbf{C}(n) \mathbf{u}(n))$ is postmultiplied by

$\mathbf{W}^{-T}(n) \mathbf{A}$ which, in general, is different from $\mathbf{C}(n)$. We may rectify this situation by interposing the matrix product $\mathbf{W}^T(n) \mathbf{W}(n)$ between them. The term $\mathbf{W}^T \mathbf{W}$, being made up of the product of matrix $\mathbf{W}$ and its transpose, is always positive definite. This is the reason why multiplication by $\mathbf{W}^T \mathbf{W}$ does not change the sign of the minima of the learning algorithm.

The important question is: What is the implication of this modification that is made in order to achieve the equivariant condition? The answer lies in how the gradient descent in parameter space is formulated. Ideally, we should use the *natural gradient*¹⁵ of the objective function $D_{f\|\hat{\mathbf{f}}}(\mathbf{W})$, defined in terms of the usual gradient $\nabla D_{f\|\hat{\mathbf{f}}}$ as:

$$\nabla^* D_{f\|\hat{\mathbf{f}}}(\mathbf{W}) = (\nabla D_{f\|\hat{\mathbf{f}}}(\mathbf{W})) \mathbf{W}^T \mathbf{W} \quad (10.103)$$

The usual gradient matrix $\nabla D_{f\|\hat{\mathbf{f}}}$ is itself defined by Eq. (10.92). In an implicit sense, the gradient $\nabla D_{f\|\hat{\mathbf{f}}}(\mathbf{W})$ is the optimum direction for descent only when the parameter space $\mathcal{W} = \{\mathbf{W}\}$ is Euclidean with an orthonormal coordinate system. In a typical situation involving neural networks, however, the parameter space $\mathcal{W}$ has a coordinate system that is nonorthonormal. The natural gradient $\nabla^* D_{f\|\hat{\mathbf{f}}}(\mathbf{W})$ will provide the *steepest descent* in this latter situation, hence the preference for using it instead of the usual gradient in formulating the stochastic algorithm for blind source separation. For the natural gradient space to be definable, two conditions must be satisfied:

1. The parameter space $\mathcal{W}$ is *Riemannian*.¹⁶ The Riemannian structure is a differentiable manifold with a positive definite metric $\mathbf{W}$.
2. The matrix $\mathbf{W}$ is nonsingular (i.e., invertible).

Both of these conditions are satisfied for the problem at hand.

By modifying the algorithm of Eq. (10.98) in this manner we may write

$$\begin{aligned} \mathbf{W}(n+1) &= \mathbf{W}(n) + \eta(n) [\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y}(n)) \mathbf{y}^T] (\mathbf{W}(n) \mathbf{W}^T(n)) \mathbf{W}^{-T}(n) \\ &= \mathbf{W}(n) + \eta(n) [\mathbf{I} - \boldsymbol{\varphi}(\mathbf{y}(n)) \mathbf{y}^T(n)] \mathbf{W}(n) \end{aligned} \quad (10.104)$$

which leads to blind source separation with the equivariance property. Figure 10.12 shows a signal-flow graph representation of Eq. (10.104).

For the adaptive algorithm described in Eq. (10.104) to produce a correct solution to the blind source separation problem described in Fig. 10.9, the following two requirements must be satisfied for all components of the output vector $\mathbf{Y}$:

- The Gram–Charlier expansion used to compute the nonlinearity $\boldsymbol{\varphi}(\cdot)$ includes a sufficient number of terms to produce a good approximation to the marginal entropy $h(Y_i)$; for example, this requirement is satisfied by the activation function of Eq. (10.93).
- The learning rate η is small enough for the estimates of the cumulants of Y_i to be reliable.

Stability Considerations

A discussion of the blind source separation problem is incomplete without consideration of the stability of the adaptive learning algorithm described in Eq. (10.104). In Amari et al. (1997), a general stability analysis of this algorithm is presented for an arbitrary activation function $\boldsymbol{\varphi}(\cdot)$. The analysis is performed in the sense of asymptotic

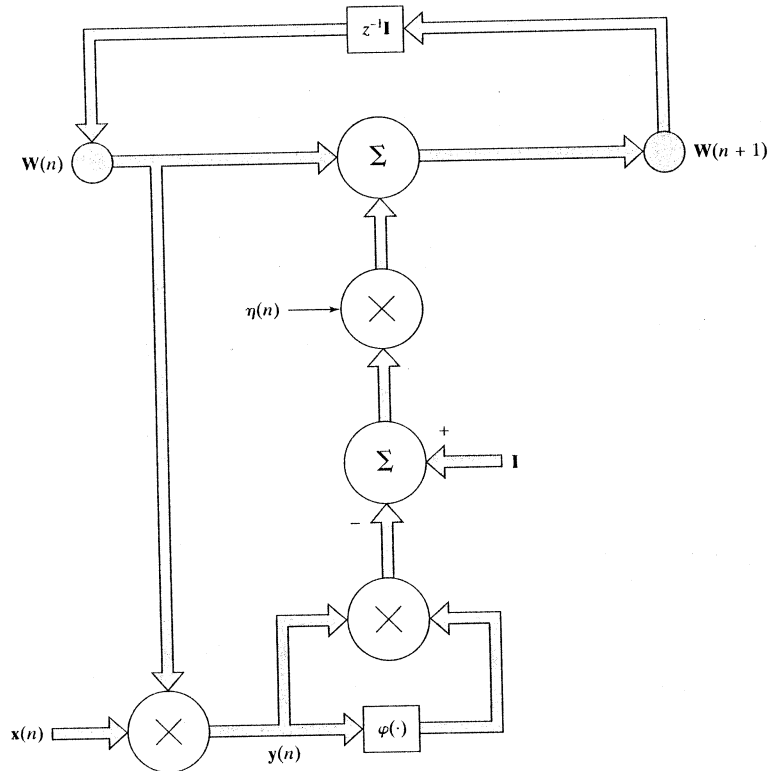


FIGURE 10.12 Signal-flow graph of the blind source separation learning algorithm described in Eq. (10.104).

convergence of the algorithm to the desired equilibrium point where a successful separation of sources is guaranteed.

Equation (10.104) is a discrete-time description of the blind source separation algorithm based on the natural gradient. For the purpose of stability analysis, the algorithm is reformulated in continuous time as follows:

$$\dot{\mathbf{W}}(t) = \eta(t)[\mathbf{I} - \varphi(\mathbf{y}(t))\mathbf{y}^T(t)]\mathbf{W}(t) \quad (10.105)$$

where t denotes continuous time, and $\dot{\mathbf{W}}(t) = \partial \mathbf{W}(t)/\partial t$. The learning-rate parameter $\eta(t)$ is positive for all time t . Let

$$\sigma_i^2 = E[y_i^2] \quad (10.106)$$

$$k_i = E\left[\frac{\partial \varphi(y_i)}{\partial y_i}\right] \quad (10.107)$$

$$q_i = E\left[y_i^2 \frac{\partial \varphi(y_i)}{\partial y_i}\right] \quad (10.108)$$

Then, according to Amari et al. (1997), the separating solution is a stable equilibrium point of the adaptive algorithm of Eq. (10.104) for an arbitrary activation function $\varphi(\cdot)$ if and only if the following conditions are satisfied

$$q_i + 1 > 0 \quad (10.109)$$

$$k_i > 0 \quad (10.110)$$

$$\sigma_i^2 \sigma_j^2 k_i k_j > 1 \quad (10.111)$$

for all (i, j) with $i \neq j$. Equations (10.109) to (10.111) are the necessary and sufficient conditions for stability of the adaptive algorithm of Eq. (10.104).

Convergence Considerations

Given that we have satisfied the stability requirements of Eqs. (10.109) through (10.111), what can we say about the convergence behavior of the learning algorithm of Eq. (10.104) based on the activation function of Eq. (10.93)? In light of an experimental study reported in Madhuranath and Haykin (1998), roughly speaking, we may say that there are two phases to the convergence process:

- In phase I, the variance $\sigma_i^2(n)$ of random variable Y_i at the demixer output goes through a period of adjustment, whereafter it reaches a fairly stable value. During this phase, the cumulants $\kappa_{i,3}$, $\kappa_{i,4}$, and $\kappa_{i,6}$ remain essentially unchanged.
- In phase II, the cumulants $\kappa_{i,3}$, $\kappa_{i,4}$, and $\kappa_{i,6}$ go through a period of adjustments of their own, whereafter they attain fairly stable values. At that point, we may say that the algorithm has converged.

It thus appears that an evaluation of the variance and higher-order cumulants of the demixer outputs (i.e., separated source signals) provides the basis of a sensible procedure for studying the convergence behavior of the learning algorithm of Eq. (10.104). It is also of interest to note that it is only during phase II that the algorithm is conforming to the Gram–Charlier expansion.

10.12 COMPUTER EXPERIMENT

Consider the system described in Fig. 10.9 involving the following three independent sources:

$$u_1(n) = 0.1 \sin(400n) \cos(30n)$$

$$u_2(n) = 0.01 \operatorname{sgn}(\sin(500n + 9 \cos(40n)))$$

$$u_3(n) = \text{noise uniformly distributed in the range } [-1, 1]$$

The mixing matrix $\mathbf{A}$ is

$$\mathbf{A} = \begin{bmatrix} 0.56 & 0.79 & -0.37 \\ -0.75 & 0.65 & 0.86 \\ 0.17 & 0.32 & -0.48 \end{bmatrix}$$

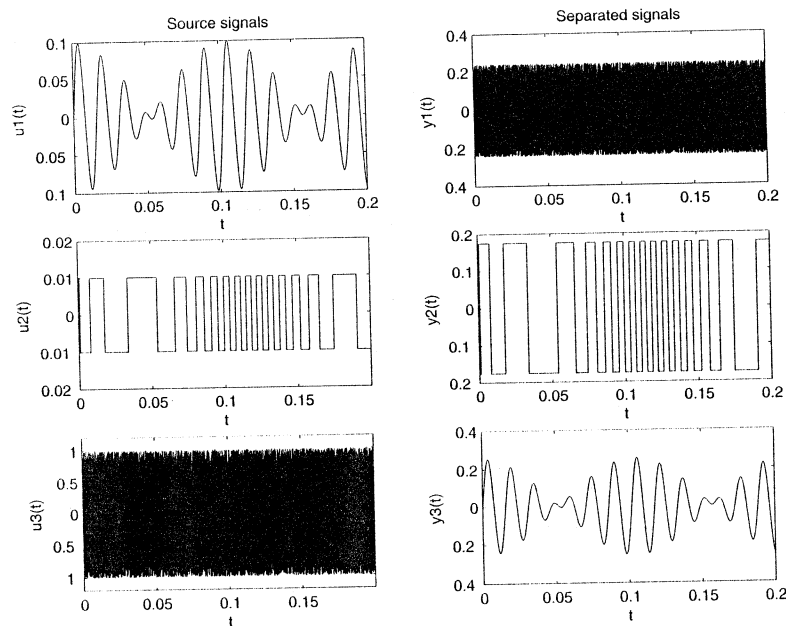


FIGURE 10.13 Waveforms on left-hand side: original source signals. Waveforms on right-hand side: separated source signals.

The waveforms of the source signals are displayed in the left-hand side of Fig. 10.13.

For the demixing, we used the batch version of the update rule described in Eq. (10.104); see Problem 10.14. Batch processing was chosen for the primary reason of improved convergence. The algorithm was implemented using the following conditions:

- **Initialization.** To initialize the algorithm, the weights in the demixing matrix $\mathbf{W}$ were picked from a random number generator with a uniform distribution inside the range $[0.0, 0.05]$.
- **Learning rate.** The learning-rate parameter was fixed at $\eta = 0.1$.
- **Signal duration.** The time series produced at the mixer output had a sampling period of 10^{-4} s and contained $N = 65,000$ samples.

The right-hand side of Fig. 10.13 displays the waveforms of the signals produced at the output of the demixer in Fig. 10.9 after 300 iterations. Except for rescaling and permutation of the unknown source outputs, there are no discernible differences between the two sets of waveforms shown in the left-hand and right-hand sides of Fig. 10.13. For the results presented here, the actual weight matrix used for initialization of the algorithm was

$$\mathbf{W}(0) = \begin{bmatrix} 0.0109 & 0.0340 & 0.0260 \\ 0.0024 & 0.0467 & 0.0415 \\ 0.0339 & 0.0192 & 0.0017 \end{bmatrix}$$

The algorithm converged to the final weight matrix

$$\mathbf{W} = \begin{bmatrix} 0.2222 & 0.0294 & -0.6213 \\ -10.1932 & -9.8141 & -9.7259 \\ 4.1191 & -1.7879 & -6.3765 \end{bmatrix}$$

The corresponding value of the matrix product $\mathbf{WA}$ is

$$\mathbf{WA} = \begin{bmatrix} -0.0032 & -0.0041 & 0.2413 \\ -0.0010 & -17.5441 & -0.0002 \\ 2.5636 & 0.0515 & -0.0009 \end{bmatrix}$$

Rearranging terms in this matrix product so that the output signals appear in the same order as the original source signals, we may write

$$\mathbf{WA} = \begin{bmatrix} 2.5636 & 0.0515 & -0.0009 \\ -0.0010 & -17.5441 & -0.0002 \\ -0.0032 & -0.0041 & 0.2413 \end{bmatrix}$$

The first, second, and third rows of the matrix product correspond to the amplitude-modulated signal, clipped frequency-modulated signal, and noise, respectively. The diagonal elements of $\mathbf{WA}$ define the factors by which the output waveforms on the right-hand side of Fig. 10.13 have been scaled with respect to the original source waveforms on the left-hand side of the figure.

For a quantitative evaluation of the demixer's performance, we may use a global rejection index defined by (Amari et al., 1996)

$$\mathcal{J} = \sum_{i=1}^m \left(\sum_{j=1}^m \frac{|p_{ij}|}{\max_k |p_{ik}|} - 1 \right) + \sum_{j=1}^m \left(\sum_{i=1}^m \frac{|p_{ij}|}{\max_k |p_{kj}|} - 1 \right)$$

where $\mathbf{P} = \{p_{ij}\} = \mathbf{WA}$. The performance index $\mathcal{J}$ is a measure of the *diagonality* of matrix $\mathbf{P}$. If the matrix $\mathbf{P}$ is perfectly diagonal, $\mathcal{J} = 0$. For a matrix $\mathbf{P}$ whose elements are not concentrated on the principal diagonal, the performance index $\mathcal{J}$ will be high.

For the waveforms displayed in Fig. 10.13, $\mathcal{J} = 0.0606$.

10.13 MAXIMUM LIKELIHOOD ESTIMATION

The method of independent components analysis (i.e., the third variant of the Infomax principle) described in the previous section is just one of many methods that have been proposed in the literature for blind source separation. In an information-theoretic context, however, there are only two other methods for performing the task of source separation in an unsupervised manner: maximum likelihood and maximum entropy. In this section we discuss maximum likelihood.

Maximum likelihood is a well-established procedure for statistical estimation with some nice properties; see note 5 of Chapter 7. In this procedure we first formulate a log-likelihood function and then optimize it with respect to the parameter vector of the probabilistic model under consideration. From the discussion presented in Chapter 7, we recall that the likelihood function is the probability density function of a data set in a

given model, but viewed as a function of the unknown parameters of the model. Referring to Fig. 10.9, let $f_U(\cdot)$ denote the probability density function of the random source vector $\mathbf{U}$. Then the probability density function of the observation vector $\mathbf{X} = \mathbf{A}\mathbf{U}$ at the output of the mixer is defined by (Papoulis, 1984)

$$f_X(\mathbf{x}, \mathbf{A}) = |\det(\mathbf{A})|^{-1} f_U(\mathbf{A}^{-1}\mathbf{x}) \quad (10.112)$$

where $\det(\mathbf{A})$ is the determinant of mixing matrix $\mathbf{A}$. Let $\mathcal{T} = \{\mathbf{x}_k\}_{k=1}^N$ denote a set of N independent realizations of the random vector $\mathbf{X}$. We may then write

$$f_X(\mathcal{T}, \mathbf{A}) = \prod_{k=1}^N f_X(\mathbf{x}_k, \mathbf{A}) \quad (10.113)$$

We find it more convenient to work with the *normalized* (divided by the sample size N) version of the log-likelihood function, as shown by

$$\begin{aligned} \frac{1}{N} \log f_X(\mathcal{T}, \mathbf{A}) &= \frac{1}{N} \sum_{k=1}^N \log f_X(\mathbf{x}_k, \mathbf{A}) \\ &= \frac{1}{N} \sum_{k=1}^N \log f_U(\mathbf{A}^{-1}\mathbf{x}_k) - \log|\det(\mathbf{A})| \end{aligned}$$

Let $\mathbf{y} = \mathbf{A}^{-1}\mathbf{x}$ be a realization of the random vector $\mathbf{Y}$ at the demixer output, and thus write

$$\frac{1}{N} \log f_X(\mathcal{T}, \mathbf{A}) = \frac{1}{N} \sum_{k=1}^N \log f_U(\mathbf{y}_k) - \log|\det(\mathbf{A})| \quad (10.114)$$

Let $\mathbf{A}^{-1} = \mathbf{W}$ and let $f_Y(\mathbf{y}, \mathbf{W})$ denote the probability density function of $\mathbf{Y}$ parameterized by $\mathbf{W}$. Then recognizing that the summation in Eq. (10.114) is the sample average of $\log f_U(\mathbf{y}_k)$, we find from the law of large numbers that, with probability 1, as the sample size N approaches infinity:

$$\begin{aligned} L(\mathbf{W}) &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=1}^N \log f_U(\mathbf{y}_k) + \log|\det(\mathbf{W})| \\ &= E[\log f_U(\mathbf{y}_k)] + \log|\det(\mathbf{W})| \\ &= \int_{-\infty}^{\infty} f_Y(\mathbf{y}, \mathbf{W}) \log f_U(\mathbf{y}) d\mathbf{y} + \log|\det(\mathbf{W})| \end{aligned} \quad (10.115)$$

where the expectation in the second line is with respect to $\mathbf{Y}$. The quantity $L(\mathbf{W})$ is the desired log-likelihood function. By writing

$$f_U(\mathbf{y}) = \left(\frac{f_U(\mathbf{y})}{f_Y(\mathbf{y}, \mathbf{W})} \right) f_Y(\mathbf{y}, \mathbf{W})$$

we may express $L(\mathbf{W})$ in the equivalent form

$$\begin{aligned} L(\mathbf{W}) &= \int_{-\infty}^{\infty} f_Y(\mathbf{y}, \mathbf{W}) \log \left(\frac{f_U(\mathbf{y})}{f_Y(\mathbf{y}, \mathbf{W})} \right) d\mathbf{y} + \int_{-\infty}^{\infty} f_Y(\mathbf{y}, \mathbf{W}) \log f_Y(\mathbf{y}, \mathbf{W}) d\mathbf{y} + \log|\det(\mathbf{W})| \\ &= -D_{f_Y \| f_U} - h(\mathbf{Y}, \mathbf{W}) + \log|\det(\mathbf{W})| \end{aligned} \quad (10.116)$$

where $h(\mathbf{Y}, \mathbf{W})$ is the differential entropy of random vector $\mathbf{Y}$ parameterized by $\mathbf{W}$, and $D_{f_Y \| f_U}$ is the Kullback–Leibler divergence between $f_Y(\mathbf{y}, \mathbf{W})$ and $f_U(\mathbf{y})$. Using Eq. (10.76) in (10.116), we may simplify the expression for the log-likelihood function $L(\mathbf{W})$ as follows (Cardoso, 1998a):

$$L(\mathbf{W}) = -D_{f_Y \| f_U} - h(\mathbf{X}) \quad (10.117)$$

where $h(\mathbf{X})$ is the differential entropy of the random vector $\mathbf{X}$ at the demixer input. The only quantity in Eq. (10.117) that depends on the weight vector $\mathbf{W}$ of the demixer is the Kullback–Leibler divergence $D_{f_Y \| f_U}$. We therefore conclude from Eq. (10.117) that maximizing the log-likelihood function $L(\mathbf{W})$ is equivalent to minimizing the Kullback–Leibler divergence $D_{f_Y \| f_U}$, that is, matching the probability distribution of the demixer output $\mathbf{Y}$ to that of the original source vector $\mathbf{U}$, which is intuitively satisfying.

Relationship between Maximum Likelihood and Independent Components Analysis

Applying the Pythagorean decomposition described in Eq. (10.45) to the problem at hand, we may express the Kullback–Leibler divergence $D_{f_Y \| f_U}$ for maximum likelihood as follows:

$$D_{f_Y \| f_U} = D_{f_Y \| \tilde{f}_Y} + D_{\tilde{f}_Y \| f_U} \quad (10.118)$$

The first Kullback–Leibler divergence $D_{f_Y \| \tilde{f}_Y}$ on the right-hand side of Eq. (10.118) is a measure of *structural mismatch* that characterizes the method of independent components analysis. The second Kullback–Leibler divergence $D_{\tilde{f}_Y \| f_U}$ is a measure of *marginal mismatch* between the marginal distribution of the demixer output $\mathbf{Y}$ and the distribution of the original source vector $\mathbf{U}$. We may thus express the “global” distribution matching criterion for maximum likelihood as follows (Amari, 1997; Cardoso, 1998a):

$$\left(\begin{array}{c} \text{Total} \\ \text{mismatch} \end{array} \right) = \left(\begin{array}{c} \text{Structural} \\ \text{mismatch} \end{array} \right) + \left(\begin{array}{c} \text{Marginal} \\ \text{mismatch} \end{array} \right) \quad (10.119)$$

“Structural mismatch” refers to the structure of a distribution pertaining to a set of independent variables, whereas “marginal mismatch” refers to the mismatch between the individual marginal distributions.

Under the ideal condition $\mathbf{W} = \mathbf{A}^{-1}$ (i.e., perfect blind source separation), both the structural mismatch and marginal mismatch vanish. At that point, maximum likelihood and independent components analysis yield exactly the same solution. The idealized relationship between maximum likelihood and independent components analysis is depicted in Fig. 10.14 (Cardoso, 1996; Amari, 1997). In this figure, $\mathcal{S}$ is the set of *all* probability density functions $f_Y(\mathbf{y})$ of the random vector $\mathbf{Y}$ at the demixer output; $\mathcal{F}$ is the set of all independent probability distributions, that is, those of the product form. Both $\mathcal{S}$ and $\mathcal{F}$ are of infinite dimension. The set $\mathcal{D} = \{f_Y(\mathbf{y}, \mathbf{W})\}$ is the finite set of probability distributions measured at the demixer output. The set $\mathcal{D}$ is m^2 -dimensional, where m is the dimension of $\mathbf{Y}$, and the weight matrix $\mathbf{W}$ is a coordinate system in it. From Fig. 10.14 we clearly see that both $D_{f_Y \| \tilde{f}_Y}$ and $D_{\tilde{f}_Y \| f_U}$ are minimized at $\mathbf{W} = \mathbf{A}^{-1}$. It is

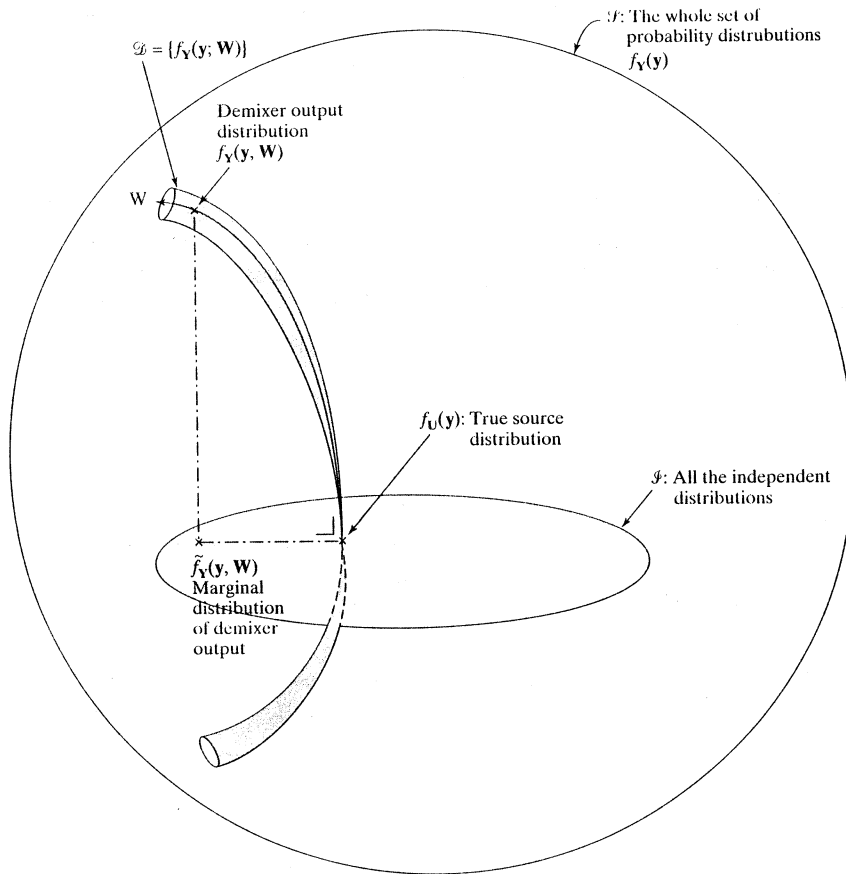


FIGURE 10.14 Illustration of the relationship between maximum likelihood and independent components analysis for blind source separation. Maximum likelihood minimizes $D_{f_Y \| f_U}$, whereas independent components analysis minimizes $D_{f_Y \| \tilde{f}_Y}$.

interesting to prove that the sets $\mathcal{D}$ and $\mathcal{F}$ are indeed orthogonal at their intersection point defined by the true probability density function $f_U(y)$.

A blind source separation algorithm based on maximum likelihood must include a provision for estimating the underlying source distributions when they are unknown, which is typically the case. The parameters for this estimation can be adapted just as we adapt the demixing weight matrix $\mathbf{W}$. In other words, we should perform a *joint estimation* of the mixing matrix and (some characteristics) of the source distributions (Cardoso, 1997, 1998a); an elegant and well developed approach for this joint estimation is presented in Pham et al. (1992, 1997).

10.14 MAXIMUM ENTROPY METHOD

The *maximum entropy method* for blind source separation is due to Bell and Sejnowski (1995). Figure 10.15 shows the block diagram of the system based on this method. As before, the demixer operates on the observation vector $\mathbf{X}$ to produce an output $\mathbf{Y} = \mathbf{W}\mathbf{X}$ that is an estimate of the original source vector $\mathbf{U}$. The vector $\mathbf{Y}$ is transformed into a vector $\mathbf{Z}$ by passing it through a component-wise nonlinearity denoted by $\mathbf{G}(\cdot)$, which is monotonic and invertible. Thus, unlike $\mathbf{Y}$, the vector $\mathbf{Z}$ is assured of a bounded differential entropy $h(\mathbf{Z})$ for an arbitrarily large demixer. For a prescribed nonlinearity $\mathbf{G}(\cdot)$, the maximum entropy method produces an estimate of the original source vector $\mathbf{U}$ by maximizing the entropy $h(\mathbf{Z})$ with respect to $\mathbf{W}$. In light of Eq. (10.55) derived in Example 10.6, we see that the maximum entropy method is closely related to the Infomax principle.¹⁷

The nonlinearity $\mathbf{G}$ is a *diagonal map* described by

$$\mathbf{G} : \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \rightarrow \begin{bmatrix} g_1(y_1) \\ g_2(y_2) \\ \vdots \\ g_m(y_m) \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix} \quad (10.120)$$

We may thus write

$$\begin{aligned} \mathbf{Z} &= \mathbf{G}(\mathbf{Y}) \\ &= \mathbf{G}(\mathbf{W}\mathbf{A}\mathbf{U}) \end{aligned} \quad (10.121)$$

Since the nonlinearity $\mathbf{G}(\cdot)$ is invertible, we may express the original source vector $\mathbf{U}$ in terms of the demixer output vector $\mathbf{Z}$ as

$$\begin{aligned} \mathbf{U} &= \mathbf{A}^{-1}\mathbf{W}^{-1}\mathbf{G}^{-1}(\mathbf{Z}) \\ &= \Psi(\mathbf{Z}) \end{aligned} \quad (10.122)$$

where $\mathbf{G}^{-1}$ is the *inverse nonlinearity*:

$$\mathbf{G}^{-1} : \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix} \rightarrow \begin{bmatrix} g_1^{-1}(z_1) \\ g_2^{-1}(z_2) \\ \vdots \\ g_m^{-1}(z_m) \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \quad (10.123)$$

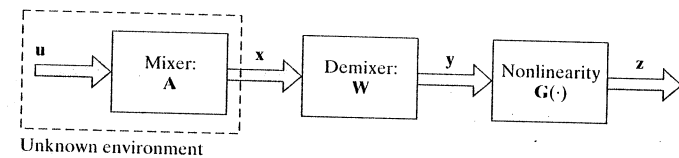


FIGURE 10.15 Block diagram of the maximum entropy method for blind source separation. The vectors $\mathbf{u}$, $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ are values of the random vectors $\mathbf{U}$, $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$, respectively.

The probability density function of the output vector $\mathbf{Z}$ is defined in terms of that of the source vector $\mathbf{U}$ by (Papoulis, 1984)

$$f_{\mathbf{Z}}(\mathbf{z}) = \frac{f_{\mathbf{U}}(\mathbf{u})}{|\det(\mathbf{J}(\mathbf{u}))|} \Big|_{\mathbf{u}=\Psi(\mathbf{z})} \quad (10.124)$$

where $\det(\mathbf{J}(\mathbf{u}))$ is the determinant of the Jacobian matrix $\mathbf{J}(\mathbf{u})$. The ij -th element of this latter matrix is defined by

$$J_{ij} = \frac{\partial z_i}{\partial u_j} \quad (10.125)$$

Hence the entropy of the random vector $\mathbf{Z}$ at the output of the nonlinearity $\mathbf{G}$ is

$$\begin{aligned} h(\mathbf{Z}) &= -E[\log f_{\mathbf{Z}}(\mathbf{z})] \\ &= -E\left[\log\left(\frac{f_{\mathbf{U}}(\mathbf{u})}{|\det(\mathbf{J}(\mathbf{u}))|}\right)\right]_{\mathbf{u}=\Psi(\mathbf{z})} \\ &= -D_{f_{\mathbf{U}}||\det\mathbf{J}} \text{ evaluated at } \mathbf{u} = \Psi(\mathbf{z}) \end{aligned} \quad (10.126)$$

We thus see that maximizing the entropy $h(\mathbf{Z})$ is equivalent to minimizing the Kullback-Leibler divergence between $f_{\mathbf{U}}(\mathbf{u})$ and a probability density function of $\mathbf{U}$ defined by $|\det(\mathbf{J}(\mathbf{u}))|$.

Suppose now the random variable Z_i (i.e., the i th element of $\mathbf{Z}$) is *uniformly distributed* inside the interval $[0, 1]$ for all i . According to Example 10.1, the entropy $h(\mathbf{Z})$ is then equal to zero. Correspondingly, we find from Eq. (10.126) that

$$f_{\mathbf{U}}(\mathbf{u}) = |\det(\mathbf{J}(\mathbf{u}))| \quad (10.127)$$

Under the ideal condition $\mathbf{W} = \mathbf{A}^{-1}$, this relationship reduces to

$$f_{U_i}(u_i) = \frac{\partial z_i}{\partial y_i} \Big|_{z_i=g(u_i)} \quad \text{for all } i \quad (10.128)$$

Conversely, we can say that if Eq. (10.128) is satisfied, then maximizing $h(\mathbf{Z})$ yields $\mathbf{W} = \mathbf{A}^{-1}$ and blind source separation is thereby achieved.

We may now summarize the results obtained on the maximum entropy method for blind source separation as follows (Bell and Sejnowski, 1995):

Let the nonlinearity at the demixer output in Fig. 10.15 be defined in terms of the original source distribution as

$$\begin{aligned} z_i &= g_i(y_i) \\ &= \int_{-\infty}^{z_i} f_{U_i}(u_i) du_i \quad \text{for } i = 1, 2, \dots, m \end{aligned} \quad (10.129)$$

Maximizing the entropy of the random vector $\mathbf{Z}$ at the output of the nonlinearity $\mathbf{G}$ is then equivalent to $\mathbf{W} = \mathbf{A}^{-1}$, which yields perfect blind source separation.

The maximum entropy and maximum likelihood methods for blind source separation are indeed equivalent under the condition that the random variable Z_i is uniformly distributed inside the interval $[0, 1]$ for all i (Cardoso, 1997). To prove this relationship we first use the chain rule of calculus to rewrite Eq. (10.125) in the equivalent form

$$\begin{aligned} J_{ij} &= \sum_{k=1}^m \frac{\partial z_i}{\partial y_i} \frac{\partial y_i}{\partial x_k} \frac{\partial x_k}{\partial u_j} \\ &= \sum_{k=1}^m \frac{\partial z_i}{\partial y_i} w_{ik} a_{kj} \end{aligned} \quad (10.130)$$

The Jacobian matrix $\mathbf{J}$ may therefore be expressed as

$$\mathbf{J} = \mathbf{D}\mathbf{W}\mathbf{A}$$

where $\mathbf{D}$ is the diagonal matrix

$$\mathbf{D} = \text{diag}\left(\frac{\partial z_1}{\partial y_1}, \frac{\partial z_2}{\partial y_2}, \dots, \frac{\partial z_m}{\partial y_m}\right)$$

Hence,

$$|\det(\mathbf{J})| = |\det(\mathbf{W}\mathbf{A})| \prod_{i=1}^m \frac{\partial z_i}{\partial y_i} \quad (10.131)$$

An estimate of the probability density function $f_{\mathbf{U}}(\mathbf{u})$ parameterized by the weight matrix $\mathbf{W}$ and the nonlinearity $\mathbf{G}$, in light of Eq. (10.131), may be written formally as (Roth and Baram, 1996)

$$f_{\mathbf{U}}(\mathbf{u}|\mathbf{W}, \mathbf{G}) = |\det(\mathbf{W}\mathbf{A})| \prod_{i=1}^m \frac{\partial g_i(y_i)}{\partial y_i} \quad (10.132)$$

We thus see that under this condition, maximizing the log-likelihood function $\log f_{\mathbf{U}}(\mathbf{u}|\mathbf{W}, \mathbf{G})$ is equivalent to maximizing the entropy $h(\mathbf{Z})$ for blind source separation. That is, the methods of maximum entropy and maximum likelihood are equivalent.

Learning Algorithm for Blind Source Separation

Referring to the second line of Eq. (10.126), we note that since the source distribution is typically fixed, maximizing the entropy $h(\mathbf{z})$ requires maximizing the expectation of the denominator term $\log|\det(\mathbf{J}(\mathbf{u}))|$ with respect to the weight matrix $\mathbf{W}$. With an adaptive algorithm for doing this computation as our goal, we may thus consider the instantaneous objective function

$$\Phi = \log|\det(\mathbf{J})| \quad (10.133)$$

Substituting Eq. (10.131) into (10.133) yields

$$\Phi = \log|\det(\mathbf{A})| + \log|\det(\mathbf{W})| + \sum_{i=1}^m \log\left(\frac{\partial z_i}{\partial y_i}\right) \quad (10.134)$$

Hence, differentiating Φ with respect to the weight matrix $\mathbf{W}$ of the demixer we get (see Problem 10.16)

$$\frac{\partial \Phi}{\partial \mathbf{W}} = \mathbf{W}^{-T} + \sum_{i=1}^m \frac{\partial}{\partial \mathbf{W}} \log \left(\frac{\partial z_i}{\partial y_i} \right) \quad (10.135)$$

To proceed further with this formula we need to specify the nonlinearity fed by the demixer output. A simple form of nonlinearity that may be used here is the logistic function

$$z_i = g(y_i) = \frac{1}{1 + e^{-y_i}}, \quad i = 1, 2, \dots, m \quad (10.136)$$

Figure 10.16 presents plots of this nonlinearity and its inverse. This figure shows that the logistic function satisfies the basic requirements of monotonicity and invertibility for blind source separation. Substituting Eq. (10.136) into (10.135) yields

$$\frac{\partial \Phi}{\partial \mathbf{W}} = \mathbf{W}^{-T} + (\mathbf{I} - 2\mathbf{z})\mathbf{x}^T$$

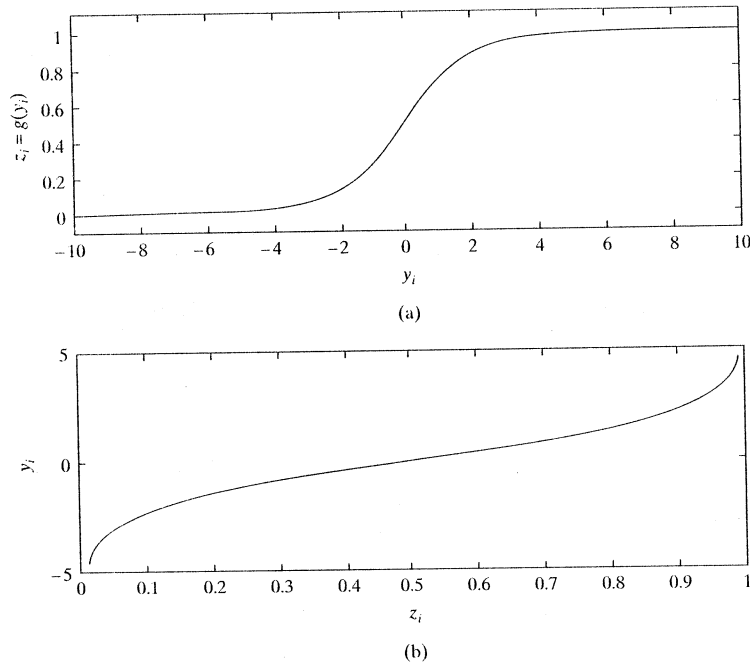


FIGURE 10.16 (a) Logistic function: $z_i = g(y_i) = \frac{1}{1 + e^{-y_i}}$. (b) Inverse of logistic function: $y_i = g^{-1}(z_i)$.

where $\mathbf{x}$ is the received signal vector, $\mathbf{z}$ is the nonlinear transformed output vector of the demixer, and $\mathbf{1}$ is a corresponding vector of ones.

The objective of the learning algorithm is to maximize the entropy $h(\mathbf{Z})$. Accordingly, invoking the method of steepest ascent, the change applied to the weight matrix $\mathbf{W}$ is (Bell and Sejnowski, 1995)

$$\begin{aligned} \Delta \mathbf{W} &= \eta \frac{\partial \Phi}{\partial \mathbf{W}} \\ &= \eta (\mathbf{W}^{-T} + (\mathbf{I} - 2\mathbf{z})\mathbf{x}^T) \end{aligned} \quad (10.137)$$

where η is the learning-rate parameter. As with the independent components analysis we may eliminate the need for inverting the transposed weight matrix $\mathbf{W}^T$ by using the natural gradient, which is equivalent to multiplying Eq. (10.137) by the matrix product $\mathbf{W}^T \mathbf{W}$. This optimal rescaling yields the desired formula for weight change as

$$\begin{aligned} \Delta \mathbf{W} &= \eta (\mathbf{W}^{-T} + (\mathbf{I} - 2\mathbf{z})\mathbf{x}^T) \mathbf{W}^T \mathbf{W} \\ &= \eta (\mathbf{I} + (\mathbf{I} - 2\mathbf{z})(\mathbf{W}\mathbf{x})^T) \mathbf{W} \\ &= \eta (\mathbf{I} + (\mathbf{I} - 2\mathbf{z})\mathbf{y}^T) \mathbf{W} \end{aligned} \quad (10.138)$$

where the vector $\mathbf{y}$ is the demixer output. The learning algorithm for computing the weight matrix $\mathbf{W}$ is therefore

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \eta (\mathbf{I} + (\mathbf{I} - 2\mathbf{z}(n))\mathbf{y}^T(n)) \mathbf{W}(n) \quad (10.139)$$

The algorithm is initiated with $\mathbf{W}(0)$ selected from a uniformly distributed set of small numbers.

Theoretical considerations and experimental investigations have shown that the learning algorithm of Eq. (10.139) is limited to separating sources with super-Gaussian distributions (Bell and Sejnowski, 1995); for the definition of super-Gaussian distributions, see note 18. This limitation is a direct consequence of using the logistic function for the nonlinearity at the back end of the system in Fig. 10.15. In particular, the logistic function imposes prior knowledge, namely a super-Gaussian shape, on the source distribution. There is nothing in the maximum entropy method, however, that restricts its use to the logistic function any more than the maximum likelihood method is restricted to some fixed prior. The application of the maximum entropy method may be broadened to a wider spectrum of source distributions by modifying the learning algorithm of Eq. (10.138) so as to provide for joint estimation of the underlying source distribution and the mixing matrix. This requirement is of a similar nature to that discussed for maximum likelihood in the previous section.

10.15 SUMMARY AND DISCUSSION

In this chapter we establish mutual information, rooted in Shannon's information theory, as a basic statistical tool for self-organization. The mutual information between an input process and an output process has some unique properties that suggest its adoption as the objective function to be optimized for self-organized learning. Indeed, some important principles for self organization have emerged from the discussion presented in this chapter:

- *Maximum mutual information (Infomax) principle* (Linsker, 1988). This principle, in its basic form, is well suited for the development of self-organized models and feature maps.
- The *first variant* of Infomax, due to Becker and Hinton (1992), is well suited for image processing where the objective is the discovery of properties of a noisy sensory input exhibiting coherence across both space and time.
- The *second variant* of Infomax, due to Ukrainec and Haykin (1992), finds applications in dual image processing where the objective is to maximize the spatial differentiation between the corresponding regions of two separate images (views) of an environment of interest.
- The third variant of Infomax for *independent components analysis* is due to Comon (1994), although its roots go back to Barlow's hypothesis (Barlow, 1985, 1989). Nevertheless, in Comon (1994) a rigorous formulation of independent components analysis was presented for the first time.
- Maximum entropy method due to Bell and Sejnowski (1995), which is also related to the Infomax principle. Maximum entropy is equivalent to maximum likelihood (Cardoso, 1997).

Independent components analysis and the maximum entropy method provide two alternative methods for blind source separation, with each one of them offering attributes of its own. A blind source separation algorithm based on the maximum entropy method is simple to implement, whereas a corresponding algorithm based on independent components analysis is more elaborate in derivation but may have broader applicability.

A neurobiological motivation that is often cited for blind source separation is the cocktail party phenomenon. This phenomenon refers to the remarkable human ability to selectively tune to and follow an auditory input of interest in a noisy environment. As explained in Chapter 2, the underlying neurobiological model involved in the solution to this very difficult signal processing problem is much more complicated than what is entailed in the idealized model described in Fig. 10.9. The neurobiological model involves both temporal and spatial forms of processing, which are needed in order to cope with unknown delays, reverberation, and noise. Now that we have a reasonably firm understanding of the basic issues involved in the neural solution to the standard blind source separation problem, it is perhaps time that we move on and tackle real-life problems on a scale comparable to the cocktail party phenomenon.

Another open research area worthy of detailed attention is that of *blind deconvolution*. Deconvolution is a signal processing operation that ideally unravels the effects of convolution performed by a linear time-invariant system operating on the input signal. More specifically, in ordinary deconvolution the output signal and the system are both known and the requirement is to reconstruct what the input signal must have been. In blind deconvolution, or in more precise terms, unsupervised deconvolution, *only* the output signal is known and there may also be information on the source statistics; the requirement is to find the input signal, the system, or both. Clearly, blind deconvolution is a more difficult signal processing task than ordinary deconvolution. Although blind deconvolution has indeed received a great deal of attention in the literature (Haykin, 1994a), our understanding of an information-theoretic approach to blind deconvolution that parallels the blind source separation problem is at an early stage of development (Douglas and Haykin, 1997). Moreover, a cost effective solution

to the blind equalization of a hostile channel such as the mobile communications channel is just as challenging in its own right as the cocktail party problem.

In summary, *blind adaptation*, be it in the context of source separation or deconvolution, has a long way to go before it can reach a mature state of development comparable to that of supervised learning.

NOTES AND REFERENCES

1. For detailed treatment of information theory, see the book by Cover and Thomas (1991); see also Gray (1990). For a collection of papers on the development of information theory (including the 1948 classic paper by Shannon), see Slepian (1973). Shannon's paper is also reproduced, with minor revisions, in the books by Shannon and Weaver (1949) and Sloane and Wyner (1993).
For a brief review of the important principles of information theory with neural processing in mind, see Atick (1992). For a treatment of information theory from a biology perspective, see Yockey (1992).
2. Linsker's maximum mutual information principle for self-organization is not to be confused with the information-content preservation rule for decision making, a rule of thumb that is briefly discussed in Chapter 7.
3. For a review of the literature on the relation between information theory and perception, see Linsker (1990c) and Atick (1992).
4. The term "entropy," in an information-theoretic context, derives its name from analogy with entropy in thermodynamics; the latter quantity is defined by (see Chapter 11)

$$H = -k_B \sum_{\alpha} p_{\alpha} \log p_{\alpha}$$

where k_B is Boltzmann's constant, and p_{α} is the probability that the system is in state α . Except for the factor k_B , the formula for entropy H in thermodynamics has exactly the same mathematical form as the definition of entropy given in Eq. (10.8).

5. In Shore and Johnson (1980), it is proved that the maximum entropy principle is correct in the following sense:

Given prior knowledge in the form of constraints, there is only one distribution satisfying these constraints that can be chosen by a procedure that satisfies the "consistency axioms;" this unique distribution is defined by maximizing entropy.

The consistency axioms are fourfold:

- I. Uniqueness: The result should be unique.
- II. Invariance: The choice of coordinates should not affect the result.
- III. System independence: It should not matter whether independent information about independent systems is accounted for separately in terms of different densities or together in terms of a joint density.
- IV. Subset independence: It should not matter whether an independent subset of system states is treated in terms of a separate conditional density or in terms of the full system density.

In Shore and Johnson (1980), it is shown that the relative entropy or the Kullback-Leibler divergence also satisfies the consistency axioms.

6. For a discussion of the method of Lagrange multipliers, see the book by Dorny (1975).

7. The term $I(X; Y)$ was originally referred to as the *rate of information transmission* by Shannon (1948). Today, however, this term is commonly referred to as the mutual information between the random variables X and Y .
8. To prove the decomposition of Eq. (10.45), we may proceed as follows. By definition we have

$$\begin{aligned}
 D_{f_X \| f_U} &= \int_{-\infty}^{\infty} f_X(\mathbf{x}) \log \left(\frac{f_X(\mathbf{x})}{f_U(\mathbf{x})} \right) d\mathbf{x} \\
 &= \int_{-\infty}^{\infty} f_X(\mathbf{x}) \log \left(\frac{f_X(\mathbf{x})}{\tilde{f}_X(\mathbf{x})} \cdot \frac{\tilde{f}_X(\mathbf{x})}{f_U(\mathbf{x})} \right) d\mathbf{x} \\
 &= \int_{-\infty}^{\infty} f_X(\mathbf{x}) \log \left(\frac{f_X(\mathbf{x})}{\tilde{f}_X(\mathbf{x})} \right) d\mathbf{x} + \int_{-\infty}^{\infty} f_X(\mathbf{x}) \log \left(\frac{\tilde{f}_X(\mathbf{x})}{f_U(\mathbf{x})} \right) d\mathbf{x} \\
 &= D_{f_X \| \tilde{f}_X} + \int_{-\infty}^{\infty} f_X(\mathbf{x}) \log \left(\frac{\tilde{f}_X(\mathbf{x})}{f_U(\mathbf{x})} \right) d\mathbf{x}
 \end{aligned} \tag{1}$$

From the definitions of $\tilde{f}_X(\mathbf{x})$ and $f_U(\mathbf{u})$, we see that

$$\begin{aligned}
 \log \left(\frac{\tilde{f}_X(\mathbf{x})}{f_U(\mathbf{x})} \right) &= \log \left(\frac{\prod_{i=1}^m \tilde{f}_{X_i}(x_i)}{\prod_{i=1}^m f_{U_i}(x_i)} \right) \\
 &= \sum_{i=1}^m \log \left(\frac{\tilde{f}_{X_i}(x_i)}{f_{U_i}(x_i)} \right)
 \end{aligned}$$

Let B denote the integral in the last line of Eq. (1). We may then write

$$\begin{aligned}
 B &= \int_{-\infty}^{\infty} f_X(\mathbf{x}) \log \left(\frac{\tilde{f}_X(\mathbf{x})}{f_U(\mathbf{x})} \right) d\mathbf{x} \\
 &= \int_{-\infty}^{\infty} f_X(\mathbf{x}) \log \left(\frac{\prod_{i=1}^m \tilde{f}_{X_i}(x_i)}{\prod_{i=1}^m f_{U_i}(x_i)} \right) d\mathbf{x} \\
 &= \sum_{i=1}^m \int_{-\infty}^{\infty} \left(\log \left(\frac{\tilde{f}_{X_i}(x_i)}{f_{U_i}(x_i)} \right) \int_{-\infty}^{\infty} f_X(\mathbf{x}) d\mathbf{x}^{(i)} \right) dx_i \\
 &= \sum_{i=1}^m \int_{-\infty}^{\infty} \log \left(\frac{\tilde{f}_{X_i}(x_i)}{f_{U_i}(x_i)} \right) \tilde{f}_{X_i}(x_i) dx_i
 \end{aligned} \tag{2}$$

where, in the last line, we have made use of the defining equation (10.39). The integral in Eq. (2) is the Kullback–Leibler divergence $D_{\tilde{f}_{X_i} \| f_{U_i}}$ for $i = 1, 2, \dots, m$. To put the expression for B in its final form, we note that the area under $\tilde{f}_{X_i}(x_i)$ is unity, and therefore write

$$\begin{aligned}
 B &= \sum_{i=1}^m \int_{-\infty}^{\infty} \prod_{j=1}^m \tilde{f}_{X_j}(x_j) \left(\log \left(\frac{\tilde{f}_{X_i}(x_i)}{f_{U_i}(x_i)} \right) \right) dx_i^{(i)} \\
 &= \int_{-\infty}^{\infty} \tilde{f}_X(\mathbf{x}) \log \left(\frac{\prod_{i=1}^m \tilde{f}_{X_i}(x_i)}{\prod_{i=1}^m f_{U_i}(x_i)} \right) d\mathbf{x} \\
 &= D_{\tilde{f}_X \| f_U}
 \end{aligned} \tag{3}$$

where in the first line we have used the definition $d\mathbf{x} = dx_i d\mathbf{x}^{(i)}$ as described in Section 10.5. Thus, substituting Eq. (3) in (1), we obtain the desired decomposition:

$$D_{f_X \| f_U} = D_{f_X \| \tilde{f}_X} + D_{\tilde{f}_X \| f_U}$$

9. Nadal and Parga (1994, 1997) also discuss the relationship between Infomax and redundancy reduction, reaching a similar conclusion that maximization of the mutual information between the input vector and output vector of a neural system leads to data reduction. Haft and van Hemmen (1998) discuss the implementation of Infomax filters for the retina. It is shown that redundancy is essential to the attainment of noise robustness of an internal representation of the environment as it is produced by a sensory system such as the retina.
10. Becker and Hinton (1992) use the acronym $I_{\max}$ to refer to the first variant of the Infomax principle.
11. In Uttley (1970) a *negative information pathway* is considered by optimizing the negative of the mutual information between the signals at the input and the output of the pathway. It is shown that such a system adapts to become a discriminator of the more frequent pattern occurring in the set of input signals during adaptation. The model is called “informon,” which is loosely related to the second variant of the Infomax principle.
12. The system described in Ukrainec and Haykin (1996) includes a postdetection processor that uses *a priori* information about the reflector location along the water-land boundary of the waterway. A *fuzzy processor* combines primary detection performance with the output from a vision-based edge detector to effectively remove false alarms, thereby resulting in a further improvement in system performance.
13. Blind source-separation may be traced back to the seminal paper by Héroult, Jutten, and Ans (1985). For a historical account of the blind source separation problem, see Nadal and Parga (1997); this paper also emphasizes the neurobiological aspects of the problem. For an insightful overview of blind source separation, with emphasis on the underlying signal processing principles, see Cardoso (1998a).
14. **Approximation of Probability Density Function**
 (a) *Gram–Charlier Expansion*
 Let $\varphi_Y(\omega)$ denote the *characteristic function* of a random variable Y having the probability density function $f_Y(y)$. By definition we have

$$\varphi_Y(\omega) = \int_{-\infty}^{\infty} f_Y(y) e^{j\omega y} dy \tag{1}$$

where $j = \sqrt{-1}$ and ω is real. In words, the characteristic function $\varphi_Y(\omega)$ is the Fourier transform of the probability density function $f_Y(y)$, except for a sign change in the exponent. In general, the characteristic function $\varphi_Y(\omega)$ is a complex number whose real and imaginary parts are finite for all ω . If the k th moment of the random variable Y exists, then $\varphi_Y(\omega)$ can be expanded in a power series in a neighborhood of $\omega = 0$ as follows:

$$\varphi_Y(\omega) = 1 + \sum_{k=1}^{\infty} \frac{(j\omega)^k}{k!} m_k \tag{2}$$

where m_k is the k th order *moment* of the random variable Y ; it is defined by

$$\begin{aligned}
 m_k &= E[Y^k] \\
 &= \int_{-\infty}^{\infty} y^k f_Y(y) dy
 \end{aligned} \tag{3}$$

Equation (2) is derived simply by substituting the expansion of the exponential function $e^{j\omega y}$ in Eq. (1), interchanging the order of summation and integration, and then invoking the definition of Eq. (3). If the characteristic function $\varphi_Y(\omega)$ can be expanded as in Eq. (2), then we may also expand the logarithm of $\varphi_Y(\omega)$ as follows (Wilks, 1962):

$$\log \varphi_Y(\omega) = \sum_{n=1}^{\infty} \frac{\kappa_n}{n!} (j\omega)^n \quad (4)$$

where κ_n is called the n th order *cumulant* or *semi-invariant* of the random variable Y . Equation (4) is derived by expanding the logarithm of $\varphi_Y(\omega)$ in a Taylor series in $j\omega$ about $\omega = 0$.

To simplify matters, henceforth we make two assumptions:

1. The random variable Y has zero mean, that is, $\mu = 0$.
2. The variance of Y is normalized to unity, that is, $\sigma^2 = 1$.

Correspondingly, we have $\kappa_1 = 0$, $\kappa_2 = 1$, and the expansion in Eq. (4) becomes

$$\log \varphi_Y(\omega) = \frac{1}{2} (j\omega)^2 + \sum_{n=3}^{\infty} \frac{\kappa_n}{n!} (j\omega)^n \quad (5)$$

Now, let

$$r(\omega) = \sum_{n=3}^{\infty} \frac{\kappa_n}{n!} (j\omega)^n$$

We may then rewrite Eq. (5) as

$$\log \varphi_Y(\omega) = \frac{1}{2} (j\omega)^2 + r(\omega)$$

That is, the characteristic function $\varphi_Y(\omega)$ may be expressed as the product of two exponential terms:

$$\varphi_Y(\omega) = \exp\left(-\frac{\omega^2}{2}\right) \cdot \exp(r(\omega)) \quad (6)$$

By using the power series expansion for the exponential term $\exp(r(\omega))$, we have

$$\exp(r(\omega)) = 1 + \sum_{l=1}^{\infty} \frac{r^l(\omega)}{l!} \quad (7)$$

By substituting Eq. (7) in (6) and collecting terms with like powers of $(j\omega)$ in the resulting double summation, we get new coefficients of the expansion of $\varphi_Y(\omega)$ such as those shown here:

$$c_1 = 0$$

$$c_2 = 0$$

$$c_3 = \frac{\kappa_3}{6}$$

$$c_4 = \frac{\kappa_4}{24}$$

$$c_5 = \frac{\kappa_5}{120}$$

$$c_6 = \frac{1}{720} (\kappa_6 + 10 \kappa_3^2)$$

$$c_7 = \frac{1}{5040} (\kappa_7 + 35 \kappa_4 \kappa_3)$$

$$c_8 = \frac{1}{40320} (\kappa_8 + 56 \kappa_5 \kappa_3 + 35 \kappa_4^2)$$

and so on. We are now in a position to take the inverse Fourier transform of $\varphi_Y(\omega)$ to obtain an expansion for the probability density function $f_Y(y)$. In particular, we may write

$$f_Y(y) = \alpha(y) \left(1 + \sum_{k=3}^{\infty} c_k H_k(y) \right) \quad (8)$$

where $\alpha(y)$ is the probability density function of a *normalized Gaussian random variable* of zero mean and unit variance:

$$\alpha(y) = \frac{1}{\sqrt{2\pi}} e^{-y^2/2} \quad (9)$$

The expansion of Eq. (8) is known as the *Gram-Charlier series* of a probability density function in terms of the Gaussian function and its derivatives (Stuart and Ord, 1994). An expansion of this type has intuitive appeal. In particular, if the random variable Y consists of the sum of a number of independently and identically distributed random variables, then as the number of those variables increases, the central limit theorem tells us that the random variable Y is asymptotically Gaussian. The first term of the Gram-Charlier series is indeed Gaussian, which means that for such a sum the remainder of the series approaches zero as the number of variables in the sum increases.

The *Hermite polynomial* $H_k(y)$ appearing in Eq. (8) is defined in terms of the k th derivative of $\alpha(y)$ by

$$\alpha^{(k)}(y) = (-1)^k \alpha(y) H_k(y) \quad (10)$$

Some typical Hermite polynomials are

$$H_0(y) = 1$$

$$H_1(y) = y$$

$$H_2(y) = y^2 - 1$$

$$H_3(y) = y^3 - 3y$$

$$H_4(y) = y^4 - 6y^2 + 3$$

$$H_5(y) = y^5 - 10y^3 + 15y$$

$$H_6(y) = y^6 - 15y^4 + 45y^2 - 15$$

A recursive relation for these polynomials is

$$H_{k+1}(y) = y H_k(y) - k H_{k-1}(y) \quad (11)$$

A particularly useful property of the Hermite polynomials is that $H_k(y)$ and the m th derivative of the Gaussian function $\alpha(y)$ are *biorthogonal*, as shown by

$$\int_{-\infty}^{\infty} H_k(y) \alpha^{(m)}(y) dy = (-1)^m m! \delta_{km}, \quad (k, m) = 0, 1, \dots \quad (12)$$

The δ_{km} is the Kronecker delta, which is equal to unity if $k = m$ and zero otherwise.

It is important to note that the natural order of the terms is *not* the best for the Gram–Charlier series. Rather, the terms listed here in parentheses should be grouped together (Helstrom, 1968)

$$k = (0), (3), (4, 6), (5, 7, 9) \quad (13)$$

The elements of these groups are usually of the same order of magnitude. If we retain terms through $k = 4$, for example, we should also include the term $k = 6$.

(b) Edgeworth Expansion

As before, let $\alpha(y)$ denote the probability density function of a random variable normalized to zero mean and unit variance. The Edgeworth expansion of the probability density function of a random variable Y about the Gaussian approximate $\alpha(y)$ is given by (Comon, 1994; Stuart and Ord, 1994)

$$\begin{aligned} \frac{f_Y(y)}{\alpha(y)} = & 1 + \frac{\kappa_3}{3!} H_3(y) + \frac{\kappa_4}{4!} H_4(y) + \frac{10\kappa_3^2}{6!} H_6(y) + \frac{\kappa_5}{5!} H_5(y) \\ & + \frac{35\kappa_3\kappa_4}{7!} H_7(y) + \frac{280\kappa_3^3}{9!} H_9(y) + \frac{\kappa_6}{6!} H_6(y) + \frac{56\kappa_3\kappa_5}{8!} H_8(y) \\ & + \frac{35\kappa_4^2}{8!} H_8(y) + \frac{2100\kappa_3^2\kappa_4}{10!} H_{10}(y) + \frac{15400\kappa_3^4}{12!} H_{12}(y) + \dots \end{aligned} \quad (14)$$

where κ_i denotes the cumulant of order i of the standardized scalar random variable Y , and H_i denotes the Hermite polynomial of order i . Equation (14) is called the *Edgeworth series*.

The key feature of the Edgeworth expansion is that its coefficients decrease uniformly. On the other hand, the terms in the Gram–Charlier expansion of Eq. (8) do not tend uniformly to zero from the viewpoint of numerical errors; that is, in general, no term is negligible compared to a preceding term. It is for this reason that the recommended procedure for truncating the Gram–Charlier expansion is to follow the grouping of terms described in Eq. (13).

15. The idea of using $\nabla^*D = (\nabla D)W^T W$ instead of the usual gradient ∇D for solving the source separation problem is described in Cardoso and Laheld (1996). Therein, ∇^*D is referred to as the *relative gradient*. This gradient is exactly the same as the *natural gradient*, the definition of which follows from an information-geometric perspective (Amari, 1998; Amari et al. 1996). A similar algorithm was described earlier in Cichocki and Moszczyński (1992) and Cichocki (et al., 1994).
16. In the Riemannian space of dimension n , for example, the squared norm of a vector $\mathbf{a}$ is defined by

$$\|\mathbf{a}\|^2 = \sum_{i=1}^n \sum_{j=1}^n g_{ij} a_i a_j$$

where the g_{ij} are functions of the coordinates $x_1, x_2, \dots, x_n$ of the Riemannian space, $g_{ij} = g_{ji}$, and the right-hand side of this expression is always positive. This expression is a generalization of the Euclidean formula for a squared norm:

$$\|\mathbf{a}\|^2 = \sum_{i=1}^n a_i^2$$

For a discussion of the Riemannian structure, see Amari (1987), and Murray and Rice (1993).

17. Bell and Sejnowski (1995) refer to their method of blind source separation as Infomax in light of Eq. (10.55) defining the relationship between the entropy $H(\mathbf{Y})$ and mutual information $I(\mathbf{Y}; \mathbf{X})$. However, the preferred terminology is “maximum entropy method” as it involves maximization of the entropy $h(\mathbf{Z})$, where $\mathbf{Z} = \mathbf{G}(\mathbf{Y})$. A note of caution: The maximum entropy method for blind source separation due to Bell and Sejnowski is *not* to be confused with the maximum entropy method (MEM) due to Burg (1975) for spectrum analysis.
18. A random variable X is said to be *sub-Gaussian* (Benveniste et al., 1987) if:
 - it is uniformly distributed, or
 - its probability density function $f_X(x)$ is expressible in the form $\exp(-g(x))$ where $g(x)$ is an even function that is differentiable, except possibly at the origin and $g(x)$ and $g'(x)/x$ are strictly increasing for $0 < x < \infty$.
 For example, we may have $g(x) = |x|^\beta$ with $\beta > 2$.

If, however, $g'(x)/x$ is strictly decreasing for $0 < x < \infty$, and the remaining properties mentioned hold, the random variable X is said to be *super-Gaussian* (Benveniste et al., 1987). For example, we may have $g(x) = |x|^\beta$ with $\beta < 2$.

Sometimes (perhaps in an abusive way) the sign of the kurtosis of a random variable is used as an indicator of its sub-Gaussianity or super-Gaussianity. The *kurtosis* of random variable X is defined by

$$K_4(x) = \frac{E[X^4]}{(E[X^2])^2} - 3$$

On this basis, the random variable X is said to be sub-Gaussian or super-Gaussian if the kurtosis $K_4(x)$ is negative or positive, respectively.

PROBLEMS

MaxEnt Principle

- 10.1 The support of a random variable X (i.e., the range of values for which it is nonzero) is defined by $[a, b]$; there is no other constraint imposed on this random variable. What is the maximum entropy distribution for this random variable? Justify your answer.

Mutual Information

- 10.2 Derive the properties of the mutual information $I(X; Y)$ between two continuous-valued random values X and Y as described in Section 10.4.
- 10.3 Consider a random input vector $\mathbf{X}$ made up of a primary component $\mathbf{X}_1$ and a contextual component $\mathbf{X}_2$. Define

$$Y_i = \mathbf{a}_i^T \mathbf{X}_1$$

$$Z_i = \mathbf{b}_i^T \mathbf{X}_2$$

How is the mutual information between $\mathbf{X}_1$ and $\mathbf{X}_2$ related to the mutual information between Y_i and Z_i ? Assume that the probability model of $\mathbf{X}$ is defined by the multivariate Gaussian distribution

$$f_{\mathbf{X}}(\mathbf{x}) = \frac{1}{(2\pi)^{m/2} (\det \Sigma)^{1/2}} \exp((\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}))$$

where $\boldsymbol{\mu}$ is the mean vector of $\mathbf{X}$ and Σ is its covariance matrix.

10.4 In this problem we explore the use of relative entropy or the Kullback–Leibler divergence to derive a supervised learning algorithm for multilayer perceptrons (Hopfield, 1987b; Baum and Wilczek, 1988). To be specific, consider a multilayer perceptron consisting of an input layer, a hidden layer, and an output layer. Given a case or example α presented to the input, the output of neuron k in the output layer is assigned a probabilistic interpretation:

$$y_{k|\alpha} = p_{k|\alpha}$$

Correspondingly, let $q_{k|\alpha}$ denote the actual (true) value of the conditional probability that the proposition k is true, given the input case α . The relative entropy for the multilayer perceptron is defined by

$$D_{p\|q} = \sum_{\alpha} p_{\alpha} \sum_k \left(q_{k|\alpha} \log \left(\frac{q_{k|\alpha}}{p_{k|\alpha}} \right) + (1 - q_{k|\alpha}) \log \left(\frac{1 - q_{k|\alpha}}{1 - p_{k|\alpha}} \right) \right)$$

where p_{α} is the *a priori* probability of occurrence of case α .

Using $D_{p\|q}$ as the cost function to be optimized, derive a learning rule for training the multilayer perceptron.

Infomax Principle

10.5 Consider two channels whose outputs are represented by the random variables X and Y . The requirement is to maximize the mutual information between X and Y . Show that this requirement is achieved by satisfying two conditions:

- (a) The probability of occurrence of X or that of Y is 0.5.
- (b) The joint probability distribution of X and Y is concentrated in a small region of the probability space.

10.6 Consider the noise model of Fig. P10.6, which shows m source nodes in the input layer of a two-neuron network. Both neurons are linear. The inputs are denoted by $X_1, X_2, \dots, X_m$, and the resulting outputs are denoted by Y_1 and Y_2 . You may make the following assumptions:

- The additive noise components N_1 and N_2 at the outputs of the network are Gaussian distributed, with zero mean and common variance σ_N^2 . They are also uncorrelated with each other.
- Each noise source is uncorrelated with the input signals.
- The output signals Y_1 and Y_2 are both Gaussian random variables with zero mean.

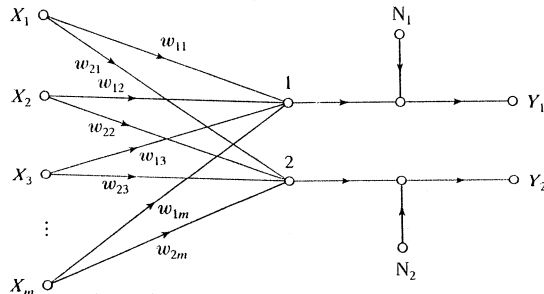


FIGURE P10.6

- (a) Determine the mutual information $I(\mathbf{Y}; \mathbf{X})$ between the output vector $\mathbf{Y} = [Y_1, Y_2]^T$ and the input vector $\mathbf{X} = [X_1, X_2, \dots, X_m]^T$.
- (b) Using the result derived in part (a), investigate the redundancy/diversity tradeoff under the following conditions (Linsker, 1988a):
 - (i) Large noise variance, represented by σ_N^2 being large compared to the variances of Y_1 and Y_2 .
 - (ii) Low noise variance, represented by σ_N^2 being small compared to the variances of Y_1 and Y_2 .

10.7 In the variant of the Infomax principle described in Section 10.9, due to Becker and Hinton (1992), the objective is to maximize the mutual information $I(Y_a; Y_b)$ between the outputs Y_a and Y_b of a noisy neural system due to the input vectors $\mathbf{X}_a$ and $\mathbf{X}_b$. In another approach discussed in Becker and Hinton (1992), a different objective is set: maximize the mutual information $I\left(\frac{Y_a + Y_b}{2}, S\right)$ between the average of the outputs Y_a and Y_b and the underlying signal component S common to these two outputs.

Using the noisy model described in Eqs. (10.59) and (10.60), do the following:

- (a) Show that

$$I\left(\frac{Y_a + Y_b}{2}, S\right) = \frac{\text{var}[Y_a + Y_b]}{\text{var}[N_a + N_b]}$$

where N_a and N_b are the noise components in Y_a and Y_b , respectively.

- (b) Demonstrate the interpretation of this mutual information as a signal-plus-noise to noise ratio.

Independent Components Analysis

10.8 Make a detailed comparison between principal components analysis (discussed in Chapter 8) and independent components analysis (discussed in this chapter).

10.9 Independent components analysis may be used as a preprocessing step for approximate data analysis before detection and classification (Comon, 1994). Discuss the property of independent components analysis that can be exploited for this application.

10.10 *Darmois' theorem* states that the sum of independent variables can be Gaussian distributed only if these variables are themselves Gaussian distributed (Darmois, 1953). Use independent components analysis to prove this theorem.

10.11 In practice, an algorithmic implementation of independent components analysis can only go for "as statistically independent as possible." Contrast the solution to the blind source separation problem using such an algorithm to the solution obtained using a decorrelation method. Assume that the covariance matrix of the observation vector is nonsingular.

10.12 Referring to the scheme described in Fig. 10.9, show that minimizing the mutual information between any two components of the demixer output $\mathbf{Y}$ is equivalent to minimizing the Kullback–Leibler divergence between the parameterized probability density function $f_Y(\mathbf{y}, \mathbf{W})$ and the corresponding factorial distribution $\tilde{f}_Y(\mathbf{y}, \mathbf{W})$.

10.13 The adaptive algorithm for blind source separation described in Eq. (10.104) has two important properties: (1) the equivariant property, and (2) the property that the weight matrix $\mathbf{W}$ is maintained nonsingular. Property (1) is discussed in some detail in the latter part of Section 10.11. In this problem we consider the second property.

Provided that the initial value $\mathbf{W}(0)$ used in starting the algorithm of Eq. (10.104) satisfies the condition $|\det(\mathbf{W}(0))| \neq 0$, show that

$$|\det(\mathbf{W}(n))| \neq 0 \quad \text{for all } n$$

This is the necessary and sufficient condition for ensuring that $\mathbf{W}(n)$ is nonsingular for all n .

10.14 In this problem we formulate the batch version of the blind source-separation algorithm described in Eq. (10.104). Specifically we write

$$\Delta \mathbf{W} = \eta \left(\mathbf{I} - \frac{1}{N} \Phi(\mathbf{Y}) \mathbf{Y}^T \right) \mathbf{W}$$

where

$$\mathbf{Y} = \begin{bmatrix} y_1(1) & y_1(2) & \cdots & y_1(N) \\ y_2(1) & y_2(2) & \cdots & y_2(N) \\ \vdots & \vdots & \ddots & \vdots \\ y_m(1) & y_m(2) & \cdots & y_m(N) \end{bmatrix}$$

and

$$\Phi(\mathbf{Y}) = \begin{bmatrix} \varphi(y_1(1)) & \varphi(y_1(2)) & \cdots & \varphi(y_1(N)) \\ \varphi(y_2(1)) & \varphi(y_2(2)) & \cdots & \varphi(y_2(N)) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(y_m(1)) & \varphi(y_m(2)) & \cdots & \varphi(y_m(N)) \end{bmatrix}$$

where N is the number of available data points. Justify the formulation of the adjustment $\Delta \mathbf{W}$ applied to the weight matrix $\mathbf{W}$ as described.

Maximum Entropy Method

10.15 Consider Fig. 10.15 in which we have

$$\mathbf{Y} = \mathbf{W}\mathbf{X}$$

where

$$\mathbf{Y} = [Y_1, Y_2, \dots, Y_m]^T$$

$$\mathbf{X} = [X_1, X_2, \dots, X_m]^T$$

and $\mathbf{W}$ is an m -by- m weight matrix. Let

$$\mathbf{Z} = [Z_1, Z_2, \dots, Z_m]^T$$

where

$$Z_k = \varphi(Y_k), \quad k = 1, 2, \dots, m$$

(a) Show that the joint entropy of $\mathbf{Z}$ is related to the Kullback–Leibler divergence $D_{f\|f}$ as follows:

$$h(\mathbf{Z}) = -D_{f\|\tilde{f}} + D_{f\|q}$$

where $D_{f\|q}$ is the Kullback–Leibler divergence between (a) the probability density function of the statistically independent (i.e., factorized) version of the output vector $\mathbf{Y}$ and (b) a “probability density function” defined by $\prod_{i=1}^m q(y_i)$.

(b) How is the formula for $h(\mathbf{Z})$ modified for the case when $q(Y_i)$ is equal to the probability density function of the original source output U_i for all i ?

10.16 (a) Starting with Eq. (10.134), derive the result given in Eq. (10.135).

(b) For the logistic function described in Eq. (10.136), show that the use of Eq. (10.135) yields the formula given in Eq. (10.137).

CHAPTER 11

Stochastic Machines and Their Approximates Rooted in Statistical Mechanics

11.1 INTRODUCTION

For our last class of unsupervised (self-organized) learning systems, we turn to statistical mechanics as the source of ideas. The subject of *statistical mechanics* encompasses the formal study of macroscopic equilibrium properties of large systems of elements that are subject to the microscopic laws of mechanics. The main aim of statistical mechanics is to derive the thermodynamic properties of macroscopic bodies starting from the motion of microscopic elements such as atoms and electrons (Landau and Lifshitz, 1980; Parisi, 1988). The number of degrees of freedom encountered here is enormous, making the use of probabilistic methods mandatory. As with Shannon’s information theory, the concept of entropy plays a vital role in the study of statistical mechanics: The more ordered the system, or the more concentrated the underlying probability distribution, the smaller the entropy. By the same token, we can say that the more disordered the system, or the more uniform the underlying probability distribution, the larger the entropy. In 1957, Jaynes showed that entropy can be used not only as the starting point of formulating statistical inference as described in the previous chapter, but also for generating the Gibbs distribution that is basic to the study of statistical mechanics.

Interest in the use of statistical mechanics as a basis for the study of neural networks goes back to the early works of Cragg and Temperley (1954) and Cowan (1968). The *Boltzmann machine* (Hinton and Sejnowski, 1983, 1986; Ackley et al., 1985) is perhaps the first multilayer learning machine inspired by statistical mechanics. The machine is named in recognition of the formal equivalence between Boltzmann’s original work on statistical thermodynamics and the network’s own dynamic behavior. Basically, the Boltzmann machine is a device for modeling the underlying probability distribution of a given data set, from which conditional distributions for use in tasks such as pattern completion and pattern classification can be derived. Unfortunately,