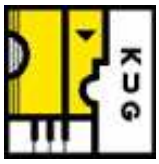
Diplomarbeit

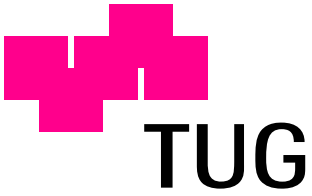
Adaptive Blind Source Separation in Multi-Channel Systems

Helmuth Ploner-Bernard

Institut für Elektronische Musik und Akustik
Vorstand: o. Univ.-Prof. Mag. art. DI Dr. techn. Robert Höldrich
Universität für Musik und Darstellende Kunst Graz

Technische Universität Graz





Begutachter: o. Univ.-Prof. Mag. art. DI Dr. techn. Robert Höldrich
Betreuer: Univ. Ass. DI Dr. techn. Alois Sontacchi

Graz, 22. November 2004

Zusammenfassung

Das Feld der blinden Quellenseparation (BSS) befasst sich damit, aus Signalen, die als Superposition von mehreren unabhängigen Quellensignalen aufzufassen sind, die ursprünglichen Quellensignale wiederherzustellen. Entscheidend ist, dass dafür lediglich die gemischten Signale zur Verfügung stehen und weder die statistischen Eigenschaften der ursprünglichen Quellensignale noch Einzelheiten über den Überlagerungsprozess bekannt sind.

Die Anwendungsgebiete der BSS umfassen neben Aufgaben im Bereich der Biomedizin, Bildanalyse und der Telekommunikation (blinde Kanalentzerrung) auch die Trennung von akustischen Signalen.

In dieser Diplomarbeit werden zunächst die für die BSS nötigen informationstheoretischen Konzepte, statistischen Schätzmethoden und Verfahren der nichtlinearen Optimierung behandelt. Darauf aufbauend wird als mächtiges Werkzeug zur blinden Quellenseparation das statistische Modell der Analyse unabhängiger Komponenten (ICA) vorgestellt, welches unter der Voraussetzung der statistischen Unabhängigkeit der Quellensignale eine Lösung des BSS-Problems ermöglicht. Ausgehend von unterschiedlichsten Ansätzen (Maximierung der Entfernung von der Gauss-Verteilung, Maximum-Likelihood-Schätzung, Minimierung der gemeinsamen Information, Diagonalisierung des Kumulantentensors) werden diverse, praktisch anwendbare Algorithmen zur adaptiven Schätzung des ICA-Modells theoretisch aufgearbeitet und vergleichend gegenübergestellt.

Abstract

The purpose of Blind Source Separation (BSS) is to recover a set of latent independent source signals from observable signals that are generated in a mixing process as superpositions of these very source signals. For this task, only the mixtures are available, whereas both the statistical properties of the original source signals and the details of the mixing process are unknown.

The field of application of BSS includes not only tasks in the area of biomedical sciences, computer vision, and telecommunications (blind channel equalization), but also the separation of acoustic signals.

This diploma thesis starts with discussing the information-theoretic concepts, the methods for parameter estimation, and some issues from nonlinear optimization theory that are needed in BSS. In this context, the statistical model of Independent Component Analysis (ICA) is introduced as a powerful tool for performing BSS, relying solely on the statistical independence of the source signals. Based on several different approaches (maximization of non-Gaussianity, maximum likelihood estimation, minimization of mutual information, diagonalization of the cumulant tensor), various implementable algorithms for adaptively estimating the ICA model are derived, compared and contrasted.

Contents

List of Figures

Part I

Fundamentals

1 Probability Theory

Since it is beyond the scope of this thesis to provide an in-depth discussion of probability theory, the basic concepts from this field are assumed to be known. However, we want to quickly review general notions regarding *multidimensional* random variables and to establish their notations. To this end, define a random vector $\boldsymbol{x}$ as a vector

$$\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^{\mathrm{T}} \tag{1.1}$$

whose components x_i are continuous-valued random variables themselves (Papoulis, 1991). Note that throughout this thesis, vectors and matrices are always denoted in boldface lowercase and uppercase symbols, respectively, be they random or deterministic.

1.1 Probability Density Functions

1.1.1 Joint Probability Density Function

For a random vector as in Eq. (1.1), we denote its *joint* (or, *multivariate*) *probability density function* (abbreviated p. d. f.) by (Papoulis, 1991)

$$p_{\boldsymbol{x}}(\boldsymbol{x}) = p_{\boldsymbol{x}}(x_1, \ldots, x_N). \tag{1.2}$$

For notational convenience, the subscript of the p. d. f. is often dropped. In general, no ambiguities should arise from that.

1.1.2 Marginal Density Function

Integrating the joint probability density function of a random vector $\boldsymbol{x}$ over one or more of the random variables yields the joint p. d. f. of the remaining random variables (Papoulis, 1991). In particular, by integrating over all random variables except one, e. g. x_i, we get the *marginal p. d. f.* of that single random variable

$p_{x_i}(x_i)$ (Papoulis, 1991):

$$p_{x_i}(x_i) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} p(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N) \mathrm{d}x_1 \cdots \mathrm{d}x_{i-1} \mathrm{d}x_{i+1} \cdots \mathrm{d}x_N. \tag{1.3}$$

1.1.3 Probability Density Function of a Transformation

Let us suppose that a random vector $\boldsymbol{y} = \begin{bmatrix} y_1 & \cdots & y_N \end{bmatrix}^{\mathrm{T}}$ is obtained from another random vector $\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^{\mathrm{T}}$ through a transformation described by N functions

$$g_1(\boldsymbol{x}), \ldots, g_N(\boldsymbol{x}) \tag{1.4}$$

as follows:

$$y_1 = g_1(\boldsymbol{x}), \ldots, y_N = g_N(\boldsymbol{x}). \tag{1.5}$$

Then, the p. d. f. $p_{\boldsymbol{y}}(\boldsymbol{y})$ of the transformed random vector can be found by solving the system

$$\begin{cases} g_1(\boldsymbol{x}) = y_1 \\ \quad \vdots \\ g_N(\boldsymbol{x}) = y_N \end{cases} \tag{1.6a}$$

or, with the functions $g_i(\boldsymbol{x})$ compiled into the vector function $\boldsymbol{g}(\boldsymbol{x})$,

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{y} \tag{1.6b}$$

for a specific set of numbers $y_1, \ldots, y_N$ (Papoulis, 1991). This yields

$$p_{\boldsymbol{y}}(\boldsymbol{y}) = \sum_j \frac{p_{\boldsymbol{x}}(\boldsymbol{x}_j)}{|\det \boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x}_j)|}, \tag{1.7}$$

where the summation is carried out over the set of solutions of the system in Eqs. (1.6), denoted $\boldsymbol{x}_j$, and $\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x}_j)$ is the *Jacobian matrix* of the vector function $\boldsymbol{g}(\boldsymbol{x})$, evaluated at the solution $\boldsymbol{x}_j$. The Jacobian matrix $\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x})$ of a vector function

$\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} g_1(\boldsymbol{x}) & \cdots & g_N(\boldsymbol{x}) \end{bmatrix}^{\mathrm{T}}$ is given by

$$\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x}) = \frac{\partial \boldsymbol{g}(\boldsymbol{x})}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial g_1(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial g_1(\boldsymbol{x})}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_N(\boldsymbol{x})}{\partial x_1} & \cdots & \frac{\partial g_N(\boldsymbol{x})}{\partial x_N} \end{bmatrix}. \tag{1.8}$$

Note that in Eq. (1.7), one term is added for every solution of the system in Eqs. (1.6), and the p. d. f. of $\boldsymbol{y}$ is zero if there does not exist any.

Example 1.1 (Probability Density Function of a Transformation)
In this example, we consider the one-to-one mapping

$$y = g(x) = \frac{\ln\left(3 - 2\left|x\right|\right)}{\ln 3}, \qquad -1 \leq x \leq 1.$$

As one can easily verify, this equation possesses the two solutions

$$x_1 = g^{-1}(y) = +\frac{3 - 3^y}{2}, \qquad 0 \leq y < 1, \quad 0 < x_1 \leq 1$$

and

$$x_2 = g^{-1}(y) = -\frac{3 - 3^y}{2}, \qquad 0 \leq y < 1, \quad -1 \leq x_2 < 0.$$

In this case, the Jacobian matrix consists of only a single entry $J_g(x)$, for which we get

$$J_g(x) = \frac{\mathrm{d}y}{\mathrm{d}x} = \frac{\mathrm{d}g(x)}{\mathrm{d}x} = \frac{1}{\ln 3} \cdot \frac{2\,\mathrm{sign}\,(x)}{2\left|x\right| - 3}, \qquad x \neq 0.$$

From Eq. (1.7), we can derive the p. d. f. $p_y(y)$ of the random variable y as a function of the random variable x:

$$\begin{aligned} p_y(y) &= \left.\frac{p_x(x)}{|J_g(x)|}\right|_{x=+\frac{3-3^y}{2}} + \left.\frac{p_x(x)}{|J_g(x)|}\right|_{x=-\frac{3-3^y}{2}}, \qquad 0 \leq y < 1 \\ &= \frac{\ln 3}{2} \cdot 3^y \left[p_x\left(\frac{3 - 3^y}{2}\right) + p_x\left(-\frac{3 - 3^y}{2}\right)\right], \qquad 0 \leq y < 1. \end{aligned}$$

If x is distributed symmetrically, this expression can be simplified to

$$p_y(y) = 3^y \ln 3 \; p_x\left(\frac{3 - 3^y}{2}\right), \qquad 0 \leq y < 1.$$

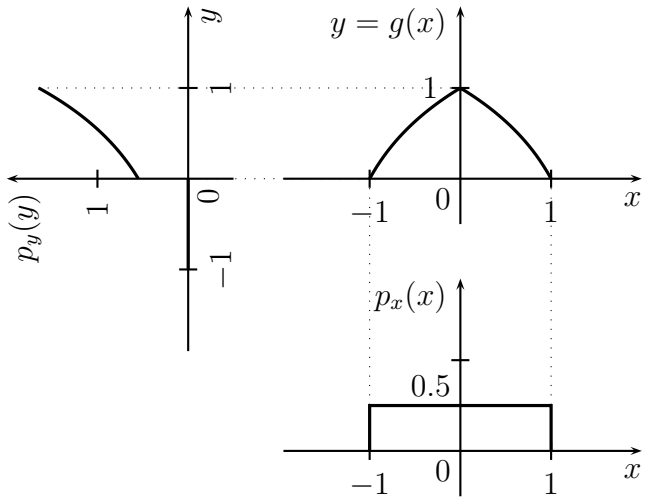


Figure 1.1: The p. d. f. of the uniformly distributed random variable x is transformed by the function $y = g(x)$ to yield a new random variable y.

Fig. 1.1 illustrates how the p. d. f. of a uniformly distributed random variable x is transformed by the function from this example. ■

Example 1.2 (Probability Density Function of a Linear Transformation) Consider next the linear transformation of a random vector $\boldsymbol{x}$ as described by

$$\begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}$$

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}.$$

If the transformation matrix $\boldsymbol{A}$ is invertible, solving this system for $\boldsymbol{x}$ gives the unique solution

$$\boldsymbol{x} = \boldsymbol{A}^{-1}\boldsymbol{y}.$$

The Jacobian matrix $\boldsymbol{J_g}(\boldsymbol{x})$ of the transformation is exactly the transformation matrix $\boldsymbol{A}$ itself:

$$\boldsymbol{J_g}(\boldsymbol{x}) = \boldsymbol{J_g} = \boldsymbol{A}.$$

Note that in this case, the Jacobian matrix $\boldsymbol{J_g}(\boldsymbol{x})$ is constant, i. e. it is not a

function of the random variables. Therefore, Eq. (1.7) yields

$$p_{\boldsymbol{y}}(\boldsymbol{y}) = \frac{p_{\boldsymbol{x}}(\boldsymbol{A}^{-1}\boldsymbol{y})}{|\det \boldsymbol{A}|} \tag{1.9}$$

for the p. d. f. of the output of the transformation. ■

1.2 Expected Value

The *expected value* of $\boldsymbol{g}(x_1, \ldots, x_N)$, also called the *mean* or the *mathematical expectation* of $\boldsymbol{g}(x_1, \ldots, x_N)$, is defined by the integral (Papoulis, 1991, Hyvärinen et al., 2001):

$$\mathcal{E}\{\boldsymbol{g}(\boldsymbol{x})\} = \int_{-\infty}^{\infty} p_{\boldsymbol{x}}(\boldsymbol{x})\boldsymbol{g}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} \tag{1.10a}$$

$$\mathcal{E}\{\boldsymbol{g}(\boldsymbol{x})\} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} p_{\boldsymbol{x}}(x_1, \ldots, x_N)\boldsymbol{g}(x_1, \ldots, x_N)\mathrm{d}x_1 \cdots \mathrm{d}x_N. \tag{1.10b}$$

Here, $\boldsymbol{g}(\boldsymbol{x})$ denotes any scalar, vector, or matrix derived from the random vector $\boldsymbol{x}$, we have to perform the integration separately for each entry, and the expected value $\mathcal{E}\{\boldsymbol{g}(\boldsymbol{x})\}$ is exactly the same size as $\boldsymbol{g}(\boldsymbol{x})$.

From the definition in Eqs. (1.10), we can easily deduce the *linearity* of the expectation operator (Papoulis, 1991) as formalized by

$$\mathcal{E}\left\{\sum_{k=1}^{N} a_k \boldsymbol{g}_k(\boldsymbol{x})\right\} = \sum_{k=1}^{N} a_k \mathcal{E}\{\boldsymbol{g}_k(\boldsymbol{x})\} \tag{1.11a}$$

$$\mathcal{E}\{a_1\boldsymbol{g}_1(\boldsymbol{x}) + \cdots + a_N\boldsymbol{g}_N(\boldsymbol{x})\} = a_1\mathcal{E}\{\boldsymbol{g}_1(\boldsymbol{x})\} + \cdots + a_N\mathcal{E}\{\boldsymbol{g}_N(\boldsymbol{x})\}, \tag{1.11b}$$

where the coefficients a_k are constant, i. e. nonrandom. However, in general (cf. Section 1.3.1) there is no such relation for the expectation of products of random variables (Papoulis, 1991):

$$\mathcal{E}\left\{\prod_{k=1}^{N} a_k \boldsymbol{g}_k(\boldsymbol{x})\right\} \neq \prod_{k=1}^{N} a_k \mathcal{E}\{\boldsymbol{g}_k(\boldsymbol{x})\}. \tag{1.12}$$

A useful application of the linearity of the expected value can be found in linear algebra. More specifically, for constant matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of suitable sizes, it holds that (Hyvärinen et al., 2001)

$$\mathcal{E}\{\boldsymbol{A}\boldsymbol{x}\} = \boldsymbol{A}\mathcal{E}\{\boldsymbol{x}\}, \qquad \mathcal{E}\{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{B}\} = \mathcal{E}\{\boldsymbol{x}^{\mathrm{T}}\}\boldsymbol{B}. \tag{1.13}$$

1.2.1 Mean Vector

The expected value of $\boldsymbol{x}$ is called the *mean vector* of $\boldsymbol{x}$ and denoted by $\boldsymbol{\eta_x}$ (Hyvärinen et al., 2001).

1.2.2 Estimating Expected Values from Data Samples

According to Hyvärinen et al. (2001), a straightforward *estimator* of the expected value in Eq. (1.10) from K samples $\boldsymbol{x}[k], k = 1, \ldots, K$ of the random vector $\boldsymbol{x}$ is given by the formula

$$\mathcal{E}\{\boldsymbol{g}(\boldsymbol{x})\} \approx \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{g}(\boldsymbol{x}[k]). \tag{1.14}$$

Applications of Eq. (1.14) include the estimation of the mean vector, the correlation matrix, and the covariance matrix of random vectors.

1.3 Dependence between Random Variables

1.3.1 Covariance, Variance and Correlation

The *covariance* C_{ij} of two random variables x_i and x_j is defined as (Papoulis, 1991)

$$C_{ij} = \mathcal{E}\left\{\left(x_i - \eta_{x_i}\right)\left(x_j - \eta_{x_j}\right)\right\} = \mathcal{E}\{x_i x_j\} - \mathcal{E}\{x_i\}\,\mathcal{E}\{x_j\}\,, \tag{1.15}$$

where η_{x_i} denotes the expected value of the random variable x_i.

The covariance evaluated at two identical indices is called *variance* $\sigma^2_{x_i}$ of the random variable x_i (Papoulis, 1991)

$$\sigma^2_{x_i} = \mathcal{E}\left\{x_i^2\right\} - \mathcal{E}\left\{x_i\right\}^2. \tag{1.16}$$

Likewise, the *correlation* R_{ij} between two random variables x_i and x_j is defined as (Papoulis, 1991)

$$R_{ij} = \mathcal{E}\{x_i x_j\}\,. \tag{1.17}$$

The random variables $x_1, \ldots, x_N$ are called (mutually) *uncorrelated* if their covariances $C_{ij} = 0$ for every pair of different indices $i \neq j$ (Papoulis, 1991). Then,

from Eq. (1.15) it follows that

$$\mathcal{E}\{x_i x_j\} = \mathcal{E}\{x_i\}\,\mathcal{E}\{x_j\}\,, \qquad i \neq j \tag{1.18}$$

for (mutually) uncorrelated random variables.

1.3.1.1 Covariance Matrix

Let $\boldsymbol{C}_{\boldsymbol{x}}$ denote the *covariance matrix* of the random vector $\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^{\mathrm{T}}$ defined as

$$\boldsymbol{C}_{\boldsymbol{x}} = \begin{bmatrix} C_{11} & \cdots & C_{1N} \\ \vdots & \ddots & \vdots \\ C_{N1} & \cdots & C_{NN} \end{bmatrix}, \tag{1.19}$$

where C_{ij} is the covariance of x_i and x_j as defined by Eq. (1.15).

1.3.1.2 Correlation Matrix

Similarly, the *correlation matrix* $\boldsymbol{R}_{\boldsymbol{x}}$ of the random vector $\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^{\mathrm{T}}$ is defined as the matrix

$$\boldsymbol{R}_{\boldsymbol{x}} = \begin{bmatrix} R_{11} & \cdots & R_{1N} \\ \vdots & \ddots & \vdots \\ R_{N1} & \cdots & R_{NN} \end{bmatrix}, \tag{1.20a}$$

where R_{ij} is the correlation of x_i and x_j as defined by Eq. (1.17). Obviously, the correlation matrix $\boldsymbol{R}_{\boldsymbol{x}}$ can also be obtained as the expected value of the outer product of the random vector $\boldsymbol{x}$ with itself

$$\boldsymbol{R}_{\boldsymbol{x}} = \mathcal{E}\left\{\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\right\}. \tag{1.20b}$$

Note that both the correlation matrix and the covariance matrix are symmetric matrices, i. e.

$$\boldsymbol{R}_{\boldsymbol{x}} = \boldsymbol{R}_{\boldsymbol{x}}^{\mathrm{T}}, \tag{1.21a}$$

and likewise for the covariance matrix

$$\boldsymbol{C}_{\boldsymbol{x}} = \boldsymbol{C}_{\boldsymbol{x}}^{\mathrm{T}}. \tag{1.21b}$$

From the definitions of covariance and correlation as given in Eq. (1.15) and Eq. (1.17), respectively, we can easily deduce the following connection between the covariance matrix $\boldsymbol{C_x}$ and the correlation matrix $\boldsymbol{R_x}$

$$\boldsymbol{C_x} = \boldsymbol{R_x} - \boldsymbol{\eta_x}\boldsymbol{\eta_x}^{\mathrm{T}}, \tag{1.22}$$

where $\boldsymbol{\eta_x}$ is the mean vector corresponding to the random vector $\boldsymbol{x}$ as defined in Section 1.2.1.

It is apparent from Eq. (1.22) that for zero-mean random vectors

$$\boldsymbol{\eta_x} = \mathbf{0} \tag{1.23}$$

the correlation matrix $\boldsymbol{R_x}$ equals the covariance matrix $\boldsymbol{C_x}$. Since in this thesis all random variables are zero-mean unless stated otherwise, the following discussion applies to both the correlation matrix and the covariance matrix.

As mentioned in Hyvärinen et al. (2001), all *eigenvalues* of the correlation matrix $\boldsymbol{R_x}$ are *real* and *nonnegative*. Moreover, it is always possible to find a set of *mutually orthonormal real eigenvectors* corresponding to the correlation matrix $\boldsymbol{R_x}$.

1.3.2 Statistical Independence

When the random variables $x_1, \ldots, x_N$ are mutually *statistically independent*, it holds that (Papoulis, 1991)

$$p_{\boldsymbol{x}}(x_1, \ldots, x_N) = p_{x_1}(x_1) \cdots p_{x_N}(x_N). \tag{1.24}$$

In other words, in the case of statistical independence, the joint p. d. f. $p_{\boldsymbol{x}}$ can be factorized into the product of marginal densities p_{x_i}.

According to Papoulis (1991), it can be shown that for statistically independent random variables x_i, the random variables

$$y_1 = g_1(x_1), \ldots, y_N = g_N(x_N) \tag{1.25}$$

are statistically independent, too.

If the random variables $x_1, \ldots, x_N$ are mutually statistically independent and have the same probability density function, the random variables are referred to as i. i. d. (independent, identically distributed).

Especially in the context of Independent Component Analysis (cf. Section 5.3), it

is crucial not to confuse uncorrelatedness with statistical independence. More precisely, statistical independence is a much stronger concept than uncorrelatedness, in the sense that statistical independence implies uncorrelatedness, but *not* vice versa.[1] Therefore, two random variables can be uncorrelated, yet *not* statistically independent, whereas every pair of statistically independent random variables is also uncorrelated.

Combining these notions with Eqs. (1.25) and (1.24), we conclude that for statistically independent random variables $x_1, \ldots, x_N$ the following equation holds true (Papoulis, 1991)

$$\mathcal{E}\{g_1(x_1) \cdots g_N(x_N)\} = \mathcal{E}\{g_1(x_1)\} \cdots \mathcal{E}\{g_N(x_N)\}, \tag{1.26}$$

where, obviously, all expectations must exist (Hyvärinen et al., 2001).

1.3.3 Group Independence

A group $\mathcal{G}_{\boldsymbol{x}}$ of random variables $\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^{\mathrm{T}}$ is statistically independent of the group $\mathcal{G}_{\boldsymbol{y}}$ of random variables $\boldsymbol{y} = \begin{bmatrix} y_1 & \cdots & y_M \end{bmatrix}^{\mathrm{T}}$ if (Papoulis, 1991)

$$p(x_1, \ldots, x_N, y_1, \ldots, y_M) = p(x_1, \ldots, x_N) p(y_1, \ldots, y_M). \tag{1.27}$$

From Eq. (1.27) it can be derived that any subset of random variables out of $\mathcal{G}_{\boldsymbol{x}}$ is statistically independent of any subset of random variables out of the group $\mathcal{G}_{\boldsymbol{y}}$. Particularly, statistical independence holds for any pair of x_i and x_j (Papoulis, 1991).

On the other hand, nothing is said about statistical dependence among the random variables inside the groups $\mathcal{G}_{\boldsymbol{x}}$ or $\mathcal{G}_{\boldsymbol{y}}$, respectively: x_i may or may not be statistically independent of the other random variables inside the group $\mathcal{G}_{\boldsymbol{x}}$, and similarly for y_j inside its group $\mathcal{G}_{\boldsymbol{y}}$.

1.4 Central Limit Theorem

Consider N statistically independent random variables $x_1, \ldots, x_N$ of continuous type and a random variable x constituted by their sum

$$x = x_1 + \cdots + x_N. \tag{1.28}$$

[1] For an important exception where uncorrelatedness does imply statistical independence see Section 1.6.

The mean η_x and the variance σ_x^2 of the random variable x are given, respectively, by

$$\eta_x = \eta_{x_1} + \cdots + \eta_{x_N}, \qquad \sigma_x^2 = \sigma_{x_1}^2 + \cdots + \sigma_{x_N}^2. \tag{1.29}$$

The *central limit theorem* (Papoulis, 1991) states now that the p. d. f. $p_x(x)$ of the random variable x approaches a Gauss p. d. f. with mean η_x and variance σ_x:

$$p_x(x) \approx \frac{1}{\sqrt{2\pi\sigma_x^2}} \exp\left[-\frac{1}{2}\frac{(x-\eta_x)^2}{\sigma_x^2}\right]. \tag{1.30}$$

The p. d. f. $p_z(z)$ of a standardized random variable z (Kreyszig, 1999) related to x by

$$z = \frac{x - \eta_x}{\sigma_x} \tag{1.31}$$

tends to a standard normal distribution as the number N of random variables in Eq. (1.28) approaches infinity

$$\lim_{N\to\infty} p_z(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right). \tag{1.32}$$

In the limit in Eq. (1.32), we cannot use the random variable x because both its mean η_x as well as its variance σ_x^2 can grow without bound, as explained in Hyvärinen et al. (2001).

According to Papoulis (1991), in the case of i. i. d. random variables x_i with smooth p. d. f.'s, the central limit theorem already holds approximately for $N = 5$ random variables.

A proof of the central limit theorem as stated in Eq. (1.30) can be found in Papoulis (1991).

1.5 Higher-Order Statistics

Higher-order statistics, covered in this section, represent an extension of the notions of the mean value and the correlation as well as the covariance of random variables. They include higher-order moments and cumulants.

In this context, let us introduce the multidimensional *characteristic function* $\Phi(\omega_1, \ldots, \omega_N)$ corresponding to a set of N random variables $x_1, \ldots, x_N$ (Mathews

and Sicuranza, 2002)

$$\Phi(\omega_1, \ldots, \omega_N) = \mathcal{E}\left\{\mathrm{e}^{\mathrm{j}(\omega_1 x_1 + \cdots + \omega_N x_N)}\right\} \tag{1.33a}$$

$$= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} p_{\boldsymbol{x}}(x_1, \ldots, x_N) \mathrm{e}^{\mathrm{j}(\omega_1 x_1 + \cdots + \omega_N x_N)} \mathrm{d}x_1 \cdots \mathrm{d}x_N, \tag{1.33b}$$

where j is the imaginary unit such that $\mathrm{j}^2 = -1$. Apparently, the characteristic function $\Phi(\omega_1, \ldots, \omega_N)$ is derived from the joint p. d. f. $p_{\boldsymbol{x}}(x_1, \ldots, x_N)$ as a Fourier-type transformation.

1.5.1 Moments

The joint *moments* of order $r = k_1 + \cdots + k_N$ of a set of N real-valued random variables $x_1, \ldots, x_N$ are defined as (Mathews and Sicuranza, 2002)

$$\mathrm{mom}\left(x_1^{k_1}, \ldots, x_N^{k_N}\right) = \mathcal{E}\left\{x_1^{k_1} \cdots x_N^{k_N}\right\} \tag{1.34a}$$

$$= \left.\frac{\partial^r \Phi(\omega_1, \ldots, \omega_N)}{\partial^{k_1}(\mathrm{j}\omega_1) \cdots \partial^{k_N}(\mathrm{j}\omega_N)}\right|_{\omega_1 = \cdots = \omega_N = 0}. \tag{1.34b}$$

The moments as defined by Eq. (1.34b) are actually the coefficients of the Taylor series expansion of the joint characteristic function $\Phi(\omega_1, \ldots, \omega_N)$ at the point $\omega_1 = \cdots = \omega_N = 0$.

On the other hand, the moments can be computed as expectations of products of random variables as suggested by Eq. (1.34a).

For a single random variable x_i, Eq. (1.34a) comprises:

- as the first moment ($N = 1, r = k_1 = 1$) the mean value η_{x_i} of the random variable x_i, as introduced in Section 1.3.1,

$$\mathrm{mom}\,(x_i) = \mathcal{E}\{x_i\} = \eta_{x_i} \tag{1.35}$$

- as the second moment ($N = 1, r = k_1 = 2$) the *average power* in the random variable x_i (Hyvärinen et al., 2001), or more specifically, from Eq. (1.16)

$$\mathrm{mom}\left(x_i^2\right) = \mathcal{E}\left\{x_i^2\right\} = \sigma_{x_i}^2 + \eta_{x_i}^2 \tag{1.36}$$

For two random variables x_i and x_j, Eq. (1.34a) includes:

- as the moment of second order ($N = 2, k_1 = k_2 = 1$) the correlation R_{ij}

between the two random variables x_i and x_j (cf. Section 1.3.1)

$$\text{mom}\,(x_i x_j) = \mathcal{E}\{x_i x_j\} = R_{ij}. \tag{1.37}$$

Since the expectation in Eq. (1.37) involves a product between two random variables, correlation (and, just as well, covariance) are often called second-order statistics.

1.5.2 Cumulants

The joint *cumulants* of order $r = k_1 + \cdots + k_N$ of a set of N real-valued random variables $x_1, \ldots, x_N$ are defined as (Mathews and Sicuranza, 2002)

$$\text{cum}\left(x_1^{k_1}, \ldots, x_N^{k_N}\right) = \left.\frac{\partial^r \ln \Phi(\omega_1, \ldots, \omega_N)}{\partial^{k_1}(\mathrm{j}\omega_1) \cdots \partial^{k_N}(\mathrm{j}\omega_N)}\right|_{\omega_1 = \cdots = \omega_N = 0}, \tag{1.38}$$

where $\ln(\cdot)$ denotes the logarithm to the base e. Similar to the case for moments as discussed in the previous section, the cumulants as defined by Eq. (1.38) are obtained as the coefficients of the Taylor series expansion of the natural logarithm of the joint characteristic function $\Phi(\omega_1, \ldots, \omega_N)$ at the point $\omega_1 = \cdots = \omega_N = 0$.

In Mathews and Sicuranza (2002), there can be found a general relationship between joint cumulants and moments of order $r = N$ allowing the computation of the joint cumulants of order N from the joint moments. The latter can in turn be estimated using Eq. (1.14). That is how cumulants can be evaluated in practice in a simple way (Hyvärinen et al., 2001).

Let us again consider first the case of a single random variable x_i, so that $N = 1$ in Eq. (1.38). For simplicity, the random variable x_i is assumed to be zero-mean. The first-order cumulant can then be shown to equal the first-order moment, or zero in this case. Likewise, the second-order and third-order cumulants equal the second-order moment and the third-order moment, respectively (Mathews and Sicuranza, 2002):

$$\text{cum}\,(x_i) = 0 \tag{1.39}$$

$$\text{cum}\left(x_i^2\right) = \text{mom}\left(x_i^2\right) = \mathcal{E}\left\{x_i^2\right\} \tag{1.40}$$

$$\text{cum}\left(x_i^3\right) = \text{mom}\left(x_i^3\right) = \mathcal{E}\left\{x_i^3\right\}. \tag{1.41}$$

Note that the third-order cumulant $\text{cum}\,(x_i^3)$ is called *skewness* in the literature.

Conversely, the fourth-order cumulant is different from the fourth-order moment.

Because of its importance, we discuss it separately in the next subsection.

In Hyvärinen et al. (2001), the following formulae can be found for the multivariate case of the zero-mean random variables x_i, x_j, x_k and x_l:

$$\operatorname{cum}(x_i x_j) = \operatorname{mom}(x_i x_j) = \mathcal{E}\{x_i x_j\} \tag{1.42}$$

$$\operatorname{cum}(x_i x_j x_k) = \operatorname{mom}(x_i x_j x_k) = \mathcal{E}\{x_i x_j x_k\} \tag{1.43}$$

$$\begin{aligned}\operatorname{cum}(x_i, x_j, x_k, x_l) = {} & \mathcal{E}\{x_i x_j x_k x_l\} - \mathcal{E}\{x_i x_j\}\,\mathcal{E}\{x_k x_l\} \\ & - \mathcal{E}\{x_i x_k\}\,\mathcal{E}\{x_j x_l\} - \mathcal{E}\{x_i x_l\}\,\mathcal{E}\{x_j x_k\}\,.\end{aligned} \tag{1.44}$$

Cumulants of orders higher than the forth order are seldom used in practice (Hyvärinen et al., 2001), which is why we stop our discussion here at the forth order.

1.5.2.1 Kurtosis

The forth-order cumulant of a single zero-mean random variable x_i is termed *kurtosis* and given by (Hyvärinen et al., 2001)

$$\operatorname{kurt}(x_i) := \operatorname{cum}\left(x_i^4\right) = \mathcal{E}\left\{x_i^4\right\} - 3\left(\mathcal{E}\left\{x_i^2\right\}\right)^2. \tag{1.45}$$

As indicated in Hyvärinen et al. (2001), the kurtosis can be used as the simplest quantitative measure of non-Gaussianity of zero-mean symmetric distributions. More precisely, Mathews and Sicuranza (2002) show that for a random variable with a Gauss distribution, all cumulants of order larger that two, and with it the kurtosis, are zero.

1.5.3 Properties of Moments and Cumulants

Moments and cumulants possess, among others, the following properties (Mathews and Sicuranza, 2002):

1. The homogeneity property holds for both moments and cumulants. To be exact, given a set of random variables $x_1, \ldots, x_N$ and a set of constants $a_1, \ldots, a_N$, it holds that

$$\operatorname{mom}(a_1 x_1, \ldots, a_N x_N) = a_1 \cdots a_N \operatorname{mom}(x_1, \ldots, x_N)\,, \tag{1.46}$$

$$\operatorname{cum}(a_1 x_1, \ldots, a_N x_N) = a_1 \cdots a_N \operatorname{cum}(x_1, \ldots, x_N)\,. \tag{1.47}$$

2. If a subset of the N random variables $x_1, \ldots, x_N$ is statistically independent of the remaining ones, the cumulant of order N is identically equal to zero, i. e.

$$\operatorname{cum}(x_1, \ldots, x_N) = 0. \tag{1.48}$$

   Note that this is not generally true for the moment of order N.

3. If a group of random variables $x_1, \ldots, x_N$ is statistically independent of another group of random variables $y_1, \ldots, y_N$ (cf. Section 1.3.3), it holds that

$$\operatorname{cum}(x_1 + y_1, \ldots, x_N + y_N) = \operatorname{cum}(x_1, \ldots, x_N) + \operatorname{cum}(y_1 + \ldots, y_N). \tag{1.49}$$

1.6 Gauss Distribution

The multivariate (or, joint) Gauss distribution for an N-dimensional random vector $\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^{\mathrm{T}}$ is defined by (Hyvärinen et al., 2001)

$$p_{\boldsymbol{x}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{N}{2}} \sqrt{\det \boldsymbol{C}_{\boldsymbol{x}}}} \exp\left[-\frac{1}{2} (\boldsymbol{x} - \boldsymbol{\eta}_{\boldsymbol{x}})^{\mathrm{T}} \boldsymbol{C}_{\boldsymbol{x}}^{-1} (\boldsymbol{x} - \boldsymbol{\eta}_{\boldsymbol{x}})\right], \tag{1.50}$$

where $\boldsymbol{\eta}_{\boldsymbol{x}}$ is the mean vector and $\boldsymbol{C}_{\boldsymbol{x}}$ is the covariance matrix of the distribution. From Eq. (1.50), the well-known p. d. f. of a single random variable can be obtained easily.

1.6.1 Properties of the Gauss Distribution

As exposed in Hyvärinen et al. (2001), the Gauss distribution has the following properties:

1. A random variable constituted as a linear combination of Gaussian-distributed random variables is Gaussian again.

2. For Gaussian-distributed random variables, uncorrelatedness is indeed equivalent to statistical independence, which is proven in Papoulis (1991).

1.6.2 Gaussianity as Measured by Kurtosis

In Independent Component Analysis (Section 5.3), there is often the need for a quantitative measure of the departure of a distribution from the Gauss distribution.

We have already touched on this issue in Section 1.5.2, where the kurtosis turned out to be potentially suitable for that purpose. To repeat, kurtosis is always zero for a Gauss distribution, whereas most other distributions generally have a kurtosis different from zero.

Of course, one can think of (non-Gaussian) distributions that do have zero kurtosis, but where some or all cumulants of order higher than the forth are nonzero. Even so, according to Hyvärinen et al. (2001) such densities are considered rather rare in practice.

In this context, a distribution whose kurtosis is positive is called *super-Gaussian*. A typical example of such a super-Gaussian distribution is the Laplace distribution $p_{x,\,\mathrm{Laplace}}(x)$ shown in Fig. 1.2(a) (Hyvärinen et al., 2001). It is given for a zero-mean random variable x as

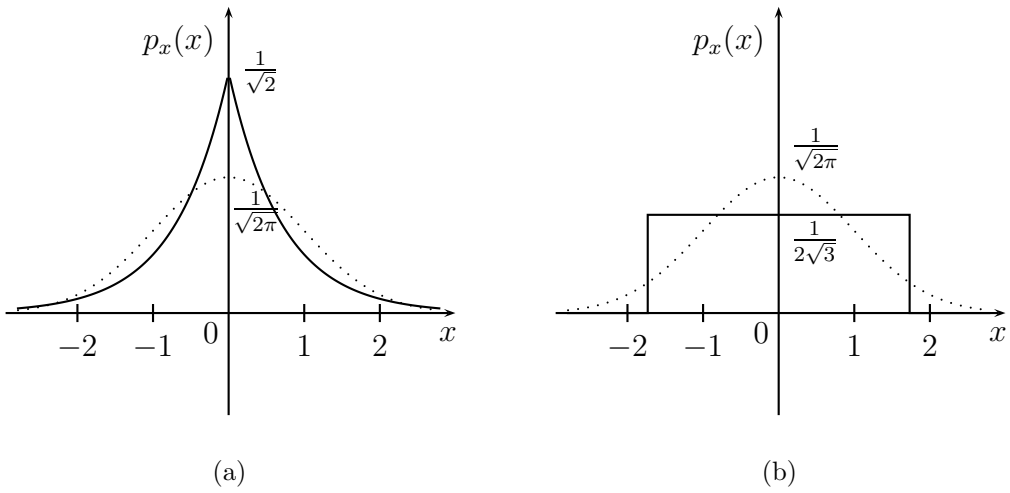
$$p_{x,\,\mathrm{Laplace}}(x) = \frac{\lambda_{\mathrm{Laplace}}}{2} \exp\left(-\lambda_{\mathrm{Laplace}}\,|x|\right), \tag{1.51}$$

with $\lambda_{\mathrm{Laplace}}$ the parameter of the distribution.

On the other hand, a distribution with a negative kurtosis is called *sub-Gaussian* (Hyvärinen et al., 2001). As an illustration for the zero-mean case, consider the uniform distribution $p_{x,\,\mathrm{Uniform}}(x)$ depicted in Fig. 1.2(b)

$$p_{x,\,\mathrm{Uniform}}(x) = \begin{cases} \frac{1}{\Delta_{\mathrm{Uniform}}}, & |x| \leq \frac{\Delta_{\mathrm{Uniform}}}{2} \\ 0 & \text{otherwise,} \end{cases} \tag{1.52}$$

where the parameter $\Delta_{\mathrm{Uniform}}$ determines the width and height of the distribution. In both figures, the standardized normal distribution with zero kurtosis is also plotted for reference.



(a) (b)

Figure 1.2: Examples of distributions with zero mean and unit variance, but different kurtoses. (a) Laplacian distribution, super-Gaussian. (b) Uniform distribution, sub-Gaussian. For comparison, the standardized normal distribution is also plotted (dotted).

2 Parameter Estimation

The variables occurring in the mathematical descriptions of distributions are called *parameters* of the distribution (Kreyszig, 1999). In practice, the exact value of the parameters of a distribution are usually unknown, so that we have to content ourselves with approximations thereof. As such an approximation we consider here the so-called (*point*) *estimate* of the parameter, which is computed from a finite set of real-world data samples (Kreyszig, 1999).

In this chapter we treat some properties of estimates, as well as the popular maximum likelihood method for parameter estimation.

2.1 Properties of Estimates

Assume that there are K data samples $\boldsymbol{x}[k], k = 1, \ldots, K$ at our disposal, and let $\hat{\vartheta}$ denote the estimate of the single parameter ϑ that is of interest to us. Then, according to Bartsch (1999), the estimate ϑ can be characterized by the following criteria:

- Unbiasedness: The estimate $\hat{\vartheta}$ is called *unbiased* if its bias

  $$b = \mathcal{E}\Big\{\hat{\vartheta}\Big\} - \vartheta \tag{2.1}$$

  is zero. In other words, the expected value of an unbiased estimate is the true value of the parameter

  $$\mathcal{E}\Big\{\hat{\vartheta}\Big\} = \vartheta. \tag{2.2}$$

- Consistency: The estimate $\hat{\vartheta}$ is called consistent if it converges to the true parameter ϑ as the number of data samples K increases.

- Efficiency: The variance of $\hat{\vartheta}$, which is itself a random variable, should be as low as possible.

- Robustness: The estimate $\hat{\vartheta}$ should not be corrupted by extreme (erroneous) values.

2.2 Maximum Likelihood Method

Suppose the probability density function (p. d. f.) $p_x(x)$ of a random variable x depends on several parameters of unknown value that we collect in an r-dimensional *parameter vector*

$$\boldsymbol{\vartheta} = \begin{bmatrix} \vartheta_1 & \cdots & \vartheta_r \end{bmatrix}^{\mathrm{T}} \tag{2.3}$$

and we are given K observations $x[k]$ of the random variable x

$$x[k], \qquad k = 1, \ldots, K. \tag{2.4}$$

The key concept of the *maximum likelihood method* is to decide on the value of the parameter vector that is most likely to have generated the observations $x[k]$ (Haykin, 2002). Here, for randomly drawn observations, this parameter vector can be found at the global maximum of the so-called *likelihood function*

$$\ell\left(\boldsymbol{\vartheta}\right) = p_x(x[1]|\boldsymbol{\vartheta}) \cdots p_x(x[K]|\boldsymbol{\vartheta}), \tag{2.5}$$

which according to basic concepts of probability theory consists of the product of the likelihood of the single observations. In Eq. (2.5), $p_x(x[k]|\boldsymbol{\vartheta})$ explicitly shows the dependence of the p. d. f. on the parameter vector $\boldsymbol{\vartheta}$. Note that the likelihood function is a function of the parameter vector $\boldsymbol{\vartheta}$ only and that the random variable x is assumed known. At a maximum inside the domain of the likelihood function $\ell\left(\boldsymbol{\vartheta}\right)$, the gradient with respect to the parameter vector has to be zero (cf. Section 4.1)

$$\frac{\partial \ell\left(\boldsymbol{\vartheta}\right)}{\partial \boldsymbol{\vartheta}} \stackrel{!}{=} \mathbf{0}. \tag{2.6}$$

A solution of Eq. (2.6) is called *maximum likelihood estimate* for $\boldsymbol{\vartheta}$ and is denoted by $\hat{\boldsymbol{\vartheta}}$ (Kreyszig, 1999).

In some cases, it is more convenient to consider the *log-likelihood function* instead

$$\mathcal{L}\left(\boldsymbol{\vartheta}\right) = \ln \ell\left(\boldsymbol{\vartheta}\right), \tag{2.7}$$

which can be used in the optimization problem just as well due to the monotonicity of the natural logarithm. In other words, the maximum likelihood estimate $\hat{\boldsymbol{\vartheta}}$ is

likewise obtained as the solution of the equation

$$\frac{\partial \ln \ell\left(\boldsymbol{\vartheta}\right)}{\partial \boldsymbol{\vartheta}} \stackrel{!}{=} \mathbf{0}. \tag{2.8}$$

Note that the most efficient (unbiased) estimate, i. e. the one whose variance reaches the so-called *Cramér-Rao lower bound*, can be obtained using the maximum likelihood method (Haykin, 2002).

3 Information Theory

One approach to solving the problem of estimating the parameters of the Independent Component Analysis model (cf. Section 8) is based on the information-theoretic concept of mutual information presented in this chapter alongside the notions of entropy and negentropy of random variables. Furthermore, we show how to approximate entropy and with it negentropy and mutual information.

3.1 Entropy

3.1.1 Entropy of a Discrete-Valued Random Variable

In Papoulis (1991), the *entropy* $H(x)$ of a single discrete-valued random variable x with probabilities $p_i, i = 1, \ldots, N$ is defined as

$$H(x) = -\sum_{i=1}^{N} p_i \log p_i, \tag{3.1}$$

where the logarithm is usually to the base 2 (Hyvärinen et al., 2001).

By examining the shape of the function inside the summation it is easy to show that the entropy $H(x)$ of a discrete-valued random variable x is always nonnegative

$$0 \leq H(x) < \infty. \tag{3.2}$$

The entropy $H(x)$ can be used as a measure of the uncertainty about a discrete-valued random variable x (Papoulis, 1991).

3.1.2 Differential Entropy

Along the lines of Eq. (3.1), according to Papoulis (1991) the *joint differential*[1] *entropy* $h(\boldsymbol{x})$ of N continuous-valued random variables $\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^{\mathrm{T}}$ with

[1] For continuous-valued random variables, the term *differential entropy* is frequently used (Hyvärinen et al., 2001).

corresponding joint p. d. f. $p_{\boldsymbol{x}}(\boldsymbol{x})$ is defined by the multidimensional integral

$$h(\boldsymbol{x}) = -\int_{-\infty}^{\infty} p_{\boldsymbol{x}}(\boldsymbol{x}) \log p_{\boldsymbol{x}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{3.3a}$$

which can be considered a mathematical expectation as introduced in Section 1.2

$$h(\boldsymbol{x}) = \mathcal{E}\{-\log p_{\boldsymbol{x}}(\boldsymbol{x})\}. \tag{3.3b}$$

Note that the range of the differential entropy $h(\boldsymbol{x})$ of continuous-type random variables $\boldsymbol{x}$ spans the whole set of real numbers (Papoulis, 1991)

$$-\infty < h(\boldsymbol{x}) < +\infty. \tag{3.4}$$

If only differences between differential entropies are considered, e. g. in the definition of negentropy in Section 3.3, entropies can be used as a measure of uncertainty about the random variables involved exactly like the entropy in the case of discrete random variables (Papoulis, 1991).

For more information about the link between entropy and its differential counterpart, consult Papoulis (1991).

3.1.3 Entropy of a Transformation

In Hyvärinen et al. (2001), the joint differential entropy $h(\boldsymbol{y})$ of the invertible transformation

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x}) \tag{3.5}$$

is shown to equal

$$h(\boldsymbol{y}) = h(\boldsymbol{x}) + \mathcal{E}\{\log |\det \boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x})|\}, \tag{3.6}$$

where $\boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x})$ is the Jacobian matrix of the vector function $\boldsymbol{f}$ evaluated at the point $\boldsymbol{x}$ (cf. Eq. (1.8)).

As a special case consider the linear transformation

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{3.7}$$

with a matrix $\boldsymbol{A}$ of suitable size, in which case Eq. (3.6) yields for the joint differ-

ential entropy $h(\boldsymbol{y})$ of the output of the transformation the expression

$$h(\boldsymbol{y}) = h(\boldsymbol{x}) + \log|\det \boldsymbol{A}|\,. \tag{3.8}$$

Note that the joint differential entropy $h(\boldsymbol{x})$ remains invariant under the application of orthogonal transformations because for orthogonal matrices $\boldsymbol{A}$ (Bartsch, 1999)

$$|\det \boldsymbol{A}| = 1, \tag{3.9}$$

$$\log|\det \boldsymbol{A}| = 0, \tag{3.10}$$

and therefore from Eq. (3.8)

$$h(\boldsymbol{y}) = h(\boldsymbol{x}). \tag{3.11}$$

3.1.4 Maximum Entropy Distributions

It can be shown that the entropy $H(x)$ of a discrete-valued random variable x is maximized if all N events possible for x are equally likely (Papoulis, 1991), i. e.

$$p_i = \frac{1}{N}. \tag{3.12}$$

In contrast, in the case of continuous-valued random variables $\boldsymbol{x}$, among all distributions with a given correlation matrix $\boldsymbol{R}_{\boldsymbol{x}}$, the multidimensional Gaussian distribution with zero mean introduced in Section 1.6 is the one maximizing the joint differential entropy $h(\boldsymbol{x})$ (Papoulis, 1991). In other words, with what we said in Section 3.1.2 about the entropy measuring the uncertainty about the random variables, by settling on the distribution possessing the maximum entropy we make the minimum number of assumptions on the data (Hyvärinen et al., 2001).

3.2 Mutual Information

Primarily, the *mutual information* $I(x_1, x_2)$ of the two random variables x_1 and x_2 is defined as (Papoulis, 1991)

$$I(x_1, x_2) = I(x_2, x_1) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{x_1,x_2}(x_1, x_2)\log\frac{p_{x_1,x_2}(x_1, x_2)}{p_{x_1}(x_1)p_{x_2}(x_2)}\mathrm{d}x_1\mathrm{d}x_2. \tag{3.13}$$

Papoulis (1991) shows that the mutual information $I(x_1, x_2)$ is always nonnegative. Furthermore, the mutual information $I(x_1, x_2)$ is zero if and only if x_1 and x_2 are statistically independent because then by Eq. (1.24) the joint p. d. f. $p_{x_1,x_2}(x_1, x_2)$ can be factorized

$$p_{x_1,x_2} = p_{x_1}(x_1)p_{x_2}(x_2), \tag{3.14a}$$

and Eq. (3.13) yields

$$I(x_1, x_2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p_{x_1,x_2}(x_1, x_2) \log \frac{p_{x_1}(x_1)p_{x_2}(x_2)}{p_{x_1}(x_1)p_{x_2}(x_2)} \mathrm{d}x_1\mathrm{d}x_2 = 0. \tag{3.14b}$$

Similarly to Eq. (3.13), it is convenient in Independent Component Analysis (Section 5.3) to define the mutual information $I(\boldsymbol{x})$ of the random variables $\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^{\mathrm{T}}$ as (Hyvärinen et al., 2001)

$$I(\boldsymbol{x}) = \int_{-\infty}^{\infty} p_{\boldsymbol{x}}(\boldsymbol{x}) \log \frac{p_{\boldsymbol{x}}(\boldsymbol{x})}{p_{x_1}(x_1) \cdots p_{x_N}(x_N)} \mathrm{d}\boldsymbol{x}, \tag{3.15a}$$

which, using the definition of differential entropy in Eq. (3.3a), can also be written as

$$I(\boldsymbol{x}) = \sum_{i=1}^{N} h(x_i) - h(\boldsymbol{x}). \tag{3.15b}$$

Here again, as shown in Hyvärinen et al. (2001), the mutual information $I(\boldsymbol{x})$ is always nonnegative and zero for statistically independent random variables $\boldsymbol{x}$ only. In this sense, mutual information can be used as a kind of distance between two multidimensional p. d. f. 's, namely

1. the joint p. d. f. $p_{\boldsymbol{x}}(\boldsymbol{x})$ and

2. the product of the marginal densities $p_{x_i}(x_i)$.

In fact, the integral in Eq. (3.15a) corresponds to the so-called *Kullback-Leibler divergence* $D_{p_{\boldsymbol{x}} \| \prod_i p_{x_i}}$ between these two densities (Haykin, 2002).

3.3 Negentropy

In Hyvärinen et al. (2001), the so-called negentropy $\mathcal{N}(\boldsymbol{x})$ of the random variables $\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^{\mathrm{T}}$ is defined as

$$\mathcal{N}(\boldsymbol{x}) = h(\boldsymbol{x}_{\mathrm{Gauss}}) - h(\boldsymbol{x}), \tag{3.16}$$

where $\boldsymbol{x}_{\mathrm{Gauss}}$ is a random vector with multidimensional Gaussian distribution (cf. Section 1.6) of the same covariance matrix as $\boldsymbol{x}$. Since the multidimensional Gaussian distribution possesses the maximum entropy as discussed in Section 3.1.4 for a given correlation matrix[2], i. e.

$$h(\boldsymbol{x}_{\mathrm{Gauss}}) \geq h(\boldsymbol{x}), \tag{3.17}$$

the negentropy $\mathcal{N}(\boldsymbol{x})$ is always nonnegative and zero if and only if the random vector $\boldsymbol{x}$ is jointly Gaussian distributed. Accordingly, the negentropy $\mathcal{N}(\boldsymbol{x})$ can be used as a measure of non-Gaussianity of the random vector $\boldsymbol{x}$.

3.3.1 Negentropy of a Linear Transformation

Using the results derived in Section 3.1.3 for the differential entropy of a linear transformation, in particular Eq. (3.8), as well as the linearity of the negentropy operator, one can show easily that negentropy is not changed by an invertible linear transformation of the form

$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}, \tag{3.18}$$

such that in this case

$$\mathcal{N}(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{x}). \tag{3.19}$$

3.3.2 Approximation of Negentropy

From a practical point of view, the direct usability of the negentropy $\mathcal{N}(x)$ as a measure of non-Gaussianity is limited for the following reasons, evident from the definition of differential entropy, on which negentropy is based (Hyvärinen et al.,

[2]Note that $\boldsymbol{x}_{\mathrm{Gauss}}$ has the same *covariance* matrix as $\boldsymbol{x}$. On the other hand, the maximum entropy property in Section 3.1.4 is defined using the *correlation* matrix. Nevertheless, we can exchange the two in this case since entropy is blind to additive constants, as can be seen from Eq. (3.6).

2001):

- The formula includes the p. d. f., which is (1) seldom known in practice and (2) not at all straightforward to estimate.[3]
- The integral involved in the formula might be computationally too expensive to evaluate for real-time application even if the p. d. f. is known.

Our methods for solving the problem of Independent Component Analysis (Section 5.3) that use negentropy are all based on approximations of this quantity, in particular on approximations based on cumulants or on approximations based on nonpolynomial functions. Here, it suffices to consider approximations of negentropy of a single random variable only.

3.3.2.1 Approximation of Negentropy by Cumulants

In approximating negentropy, let us assume that the distribution of the random variable under consideration is not far away from the Gaussian distribution. Setting the Gaussian distribution as the point of reference seems reasonable since we intend to use negentropy as a measure of non-Gaussianity. Then, in a classical approach to approximating negentropy, we can use a truncated *Gram-Charlier expansion* of the p. d. f. of the standardized random variable x in the vicinity of the standardized Gaussian distribution and a *second-order Taylor series expansion of the logarithm*[4] involved in the definition of differential entropy. A step-by-step derivation for the interested reader can be found in Hyvärinen et al. (2001). We obtain for the negentropy $\mathcal{N}(x)$ of a standardized random variable x

$$\mathcal{N}(x) \approx \frac{1}{12}\left(\mathcal{E}\left\{x^3\right\}\right)^2 + \frac{1}{48}\operatorname{kurt}(x)^2, \tag{3.20a}$$

where $\operatorname{kurt}(x)$ denotes the kurtosis of the random variable x defined in Section 1.5.2.1. Note that the first term involving the skewness $\mathcal{E}\{x^3\}$ vanishes for symmetric distributions, so we are left with the expression

$$\mathcal{N}(x) \approx \frac{1}{48}\operatorname{kurt}(x)^2. \tag{3.20b}$$

[3] Elaborating on the latter issue, we state that for example a simple histogram estimate of the p. d. f. would not be appropriate in this context since it would lead inherently to a description of discrete-valued random variables only, and it is not the Gaussian distribution that has the maximum entropy for discrete-valued random variables (cf. Section 3.1.4). We conclude that the negentropy derived from such a histogram estimate would not be minimized by a Gaussian-distributed random variable and therefore not be a suitable measure of non-Gaussianity.

[4] Here, the logarithm is to the base e, though.

As pointed out by Hyvärinen et al. (2001), the approximation of negentropy in Eq. (3.20) is computationally very simple. As a drawback, those very authors mention the sensitivity of finite-sample estimators to outliers with large values (lack of robustness, cf. Section 2.1). Furthermore, cumulants basically measure the tails of distributions while having a tendency to being unaffected by structure near the center of the distribution since estimators of cumulants are stronger influenced by large values of the random variable. This is especially true for cumulants of higher-order.

The above-mentioned limitations can be overcome by the approximation of negentropy by nonpolynomial functions presented in the next section.

3.3.2.2 Approximation of Negentropy by Nonpolynomial Functions

As an alternative to the approximation of negentropy by cumulants, here we mention a method based on an *approximative maximum entropy method* as proposed in Hyvärinen et al. (2001). More specifically, suppose that we have estimated some expectations

$$\mathcal{E}\{F_i(x)\} = \int_{-\infty}^{\infty} p_x(x) F_i(x) \mathrm{d}x = c_i, \qquad i = 1, \ldots, N, \tag{3.21}$$

where $\{F_i(x)\}$ is a set of N nonlinear functions in x, and we want to employ the maximum entropy method to help us in making the right choice from all distributions compatible with the estimated expectations. To see the reason for this, bear in mind that in Independent Component Analysis we always try to minimize negentropy and that in the end the minimization of a cost function based on the maximum entropy distribution hopefully minimizes the true entropy as well.

Unfortunately, the nonlinear functions $F_i(x)$ in Eq. (3.21) make it impossible to solve analytically the problem of finding the maximum entropy distribution. In order to find a reasonable simplification, remember from Section 3.1.4 that the maximum entropy has the form of a Gaussian distribution. Therefore, we make a *first-order approximation of the exponential function* describing our maximum entropy distribution. In other words, we again consider a random variable x in the vicinity of the Gaussian distribution, which is similar to what we did in the previous section.

In the last step, once again using a second-order Taylor series expansion of the logarithm in the definition of differential entropy, from the resulting approximative maximum entropy distribution we obtain the following approximation of the

negentropy $\mathcal{N}(x)$ (Hyvärinen et al., 2001):

$$\mathcal{N}(x) \approx \frac{1}{2} \sum_{i=1}^{N} \mathcal{E}\{F_i(x)\}^2 . \tag{3.22}$$

It can be shown that the approximation in Eq. (3.22) is a suitable measure of non-Gaussianity because it is minimized for a Gaussian-distributed random variable.

In Hyvärinen et al. (2001), you can find guidelines on how to choose the best functions $F_i(x)$ in general. In particular, the functions $F_i(x)$ should not grow faster than quadratically with increasing absolute value of the independent variable x since otherwise the resulting approximation of negentropy might lack the desired robustness property.

The simplest approximation of negentropy based on nonpolynomial functions uses just one nonlinear function $G(x)$ so that $N = 1$ in Eq. (3.22) and in Eq. (3.21):

$$\mathcal{N}(x) \approx k_1 \big(\mathcal{E}\{G(x)\} - \mathcal{E}\{G(\nu)\}\big)^2 \tag{3.23}$$

with k_1 a proper constant and ν denoting a standardized random variable with Gaussian distribution (Hyvärinen et al., 2001). For the nonlinear function $G(x)$ in Eq. (3.23), Hyvärinen et al. (2001) propose

$$G_1(x) = \frac{1}{a_1} \ln \cosh(a_1 x), \tag{3.24a}$$

where the constant $1 \leq a_1 \leq 2$ is often chosen as unity, or the function

$$G_2(x) = -\mathrm{e}^{-\frac{x^2}{2}}. \tag{3.24b}$$

Both the function $G_1(x)$ and the function $G_2(x)$ are plotted in Fig. 3.1. For purposes of comparison, the forth power of x corresponding to the approximation of negentropy by cumulants as discussed in the previous section is also shown, which obviously grows much faster with increasing absolute values of x.

To conclude, as mentioned in Hyvärinen et al. (2001), the approximation of negentropy by nonpolynomial functions is both more robust against erroneous outliers and more accurate than the approximation of negentropy by cumulants, yet of comparable computational complexity.

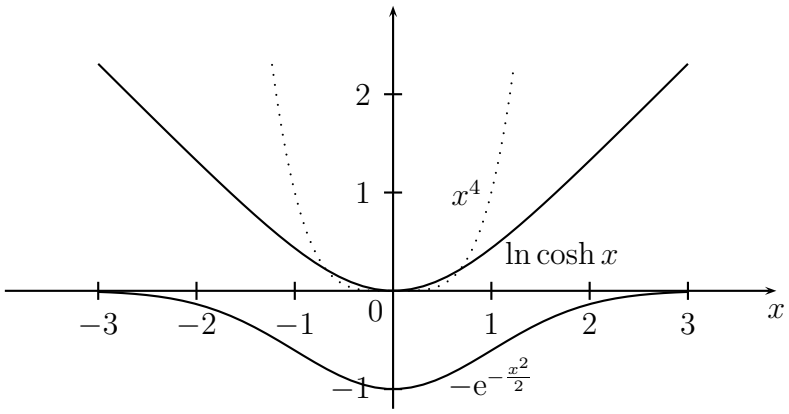


Figure 3.1: Two functions suitable for approximating negentropy by one nonlinear function. For reference, the forth power of x is also shown.

4 Optimization Theory

4.1 Basic Concepts

Suppose we want to tackle a so-called *optimization problem*, i. e. we are given a function $\mathcal{I}$ that we wish to *optimize* (*maximize* or *minimize*). In *optimization theory*, this function $\mathcal{I}$ is typically called the *objective function* (Kreyszig, 1999), whereas in the field of adaptive signal processing the terms *cost function* or *contrast function* are more common (Haykin, 2002).

4.1.1 Constrained and Unconstrained Optimization

As for example in the context of blind deconvolution in the frequency domain, the cost function $\mathcal{I}$ may generally depend on several complex-valued variables

$$\mathcal{I} = \mathcal{I}(w_1, \ldots, w_N) = \mathcal{I}(\boldsymbol{w}), \tag{4.1}$$

where we introduced the vector notation $\boldsymbol{w} = \begin{bmatrix} w_1 & \cdots & w_N \end{bmatrix}^{\mathrm{T}}$ for convenience of presentation.

In contrast to this *unconstrained optimization*, additional equations or inequalities (*constraints*) involving the variables $\boldsymbol{w}$ are sometimes to be met at the same time, e. g. in the case of Independent Component Analysis (Section 5.3), where in addition to minimizing some contrast function we often require our solution vector to be of unit norm. This kind of optimization is called *constrained optimization*.

4.1.2 Minima and Maxima

From calculus recall the definition of *local* minima and maxima (Bartsch, 1999, Kreyszig, 1999):

Local minimum A differentiable function $\mathcal{I}(\boldsymbol{w})$ is said to have a *local* (or, *relative*) *minimum* at the point $\boldsymbol{w}_{\text{ext}}$ if in a region R around that point $\boldsymbol{w}_{\text{ext}}$ it holds

that

$$\mathcal{I}(\boldsymbol{w}) > \mathcal{I}(\boldsymbol{w}_{\text{ext}}) \text{ for all } \boldsymbol{w} \neq \boldsymbol{w}_{\text{ext}}. \tag{4.2}$$

Local maximum Likewise, a differentiable function $\mathcal{I}(\boldsymbol{w})$ is said to have a *local* (or, *relative*) *maximum* at the point $\boldsymbol{w}_{\text{ext}}$ if in a region R around that point $\boldsymbol{w}_{\text{ext}}$ it holds that

$$\mathcal{I}(\boldsymbol{w}) < \mathcal{I}(\boldsymbol{w}_{\text{ext}}) \text{ for all } \boldsymbol{w} \neq \boldsymbol{w}_{\text{ext}}. \tag{4.3}$$

Minima and maxima together are called *extrema*.

If Eqs. (4.2) and (4.3) hold for all $\boldsymbol{w}$ inside the domain of the cost function $\mathcal{I}(\boldsymbol{w})$, the point $\boldsymbol{w}_{\text{ext}}$ is called *global* (or, *absolute*) minimum and maximum, respectively.

A necessary condition for the point $\boldsymbol{w}_{\text{ext}}$ to be an extremum of the cost function $\mathcal{I}(\boldsymbol{w})$ is that all partial derivatives $\frac{\partial \mathcal{I}(\boldsymbol{w})}{\partial w_1}, \ldots, \frac{\partial \mathcal{I}(\boldsymbol{w})}{\partial w_N}$ must exist and be zero at $\boldsymbol{w}_{\text{ext}}$. Since these partial derivatives are nothing but the components of the *gradient* $\operatorname{grad} \mathcal{I}(\boldsymbol{w})$ of the scalar cost function $\mathcal{I}(\boldsymbol{w}) = \mathcal{I}(w_1, \ldots, w_N)$

$$\operatorname{grad} \mathcal{I}(\boldsymbol{w}) = \frac{\partial \mathcal{I}(\boldsymbol{w})}{\partial \boldsymbol{w}} = \begin{bmatrix} \frac{\partial \mathcal{I}(\boldsymbol{w})}{\partial w_1} \\ \vdots \\ \frac{\partial \mathcal{I}(\boldsymbol{w})}{\partial w_N} \end{bmatrix}, \tag{4.4}$$

we can also write

$$\operatorname{grad} \mathcal{I}(\boldsymbol{w}_{\text{ext}}) \stackrel{!}{=} \mathbf{0}. \tag{4.5}$$

Since the condition in Eq. (4.5) is not also a sufficient condition for $\boldsymbol{w}_{\text{ext}}$ to be an extremum, after solving this equation, we still have to determine whether $\boldsymbol{w}_{\text{ext}}$ is in fact a maximum or a minimum. Even worse, our solution $\boldsymbol{w}_{\text{ext}}$ could also be a saddle-point, which is not an extremum in the first place as pointed out by Kreyszig (1999).

4.1.3 Solving Optimization Problems by Numerical Methods

If the cost function $\mathcal{I}(\boldsymbol{w})$ is a general nonlinear function, it might not be possible to solve Eq. (4.5) analytically at all. In this case, we have to fall back to numerical methods for solving Eq. (4.5), a few of which are presented in this chapter. More specifically, we will cover the *fixed-point iteration*, *Newton's method* and the *method of steepest descent*.

4.2 Solution of Equations by Iteration

Consider a system of N nonlinear equations in N independent variables $w_1, \ldots, w_N$

$$\begin{cases} g_1(w_1, \ldots, w_N) = 0 \\ \quad \vdots \qquad\qquad N \geq 2 \\ g_N(w_1, \ldots, w_N) = 0 \end{cases} \tag{4.6a}$$

or in matrix form

$$\boldsymbol{g}(\boldsymbol{w}) = \mathbf{0}, \tag{4.6b}$$

where $\boldsymbol{g}(\boldsymbol{w})$ is a known (given) vector function, i. e. a vector of scalar functions $g_i(\boldsymbol{w}), i = 1, \ldots, N$, and $\boldsymbol{w}$ is the vector of independent variables. The vector function $\boldsymbol{g}(\boldsymbol{w})$ could for example be the gradient of a scalar cost function $\mathcal{I}(\boldsymbol{w})$ as required for an extreme value of $\mathcal{I}(\boldsymbol{w})$. A vector $\boldsymbol{w}_{\text{opt}}$ such that $\boldsymbol{g}(\boldsymbol{w}_{\text{opt}}) = \mathbf{0}$ is met is called a *solution* of Eq. (4.6) (Kreyszig, 1999). Analytical formulae for the task of solving Eq. (4.6) exist only in very simple cases.[1] Therefore, we almost entirely depend on numerical methods for solving the equation, a few of which are discussed in the following sections.

4.2.1 Update Rule

The numerical methods considered here are all *iterative methods*. The idea of iterative methods is to approach the true solution $\boldsymbol{w}_{\text{opt}}$ of a given equation step by step by starting from an initial guess $\boldsymbol{w}[0]$ and then iteratively computing a sequence of vectors $\boldsymbol{w}[1], \boldsymbol{w}[2], \ldots$ which approximate the desired solution better and better each iteration step (Kreyszig, 1999). More precisely, we compute $\boldsymbol{w}[1]$ from $\boldsymbol{w}[0]$, then $\boldsymbol{w}[2]$ from $\boldsymbol{w}[1]$ and so on, or – generally – we recursively determine the new value $\boldsymbol{w}[n+1]$ from the old value $\boldsymbol{w}[n]$ according to a specific *update rule* $\boldsymbol{\varphi}(\boldsymbol{w}[n])$, where n denotes the iteration step.[2] Let us formalize this update procedure by

$$\boldsymbol{w}[n+1] = \boldsymbol{\varphi}(\boldsymbol{w}[n]), \qquad n = 0, 1, \ldots . \tag{4.7}$$

[1] To be more specific, according to Bartsch (1999) no general formulae are possible for algebraic equation of 5th and higher degree.

[2] In general, the update rule can of course include longer memory, e. g. in some iterative methods $\boldsymbol{w}[n+1]$ explicitly depends not only on $\boldsymbol{w}[n]$ but also on $\boldsymbol{w}[n-1]$.

4.2.2 Convergence

Needless to say, the iterative method should eventually lead to the true solution $\boldsymbol{w}_{\text{opt}}$, i. e.

$$\lim_{n\to\infty} \boldsymbol{w}[n] = \boldsymbol{w}_{\text{opt}}. \tag{4.8}$$

Whenever Eq. (4.8) holds for a particular choice of $\boldsymbol{w}[0]$, the iteration process Eq. (4.7) is said to *converge* onto the true solution $\boldsymbol{w}_{\text{opt}}$ (Kreyszig, 1999). Otherwise, the series $\boldsymbol{w}[n]$ *diverges* (Kreyszig, 1999, Bartsch, 1999).

4.2.3 Order of an Iteration Method, Convergence Speed

The speed of convergence of an iteration method can be measured by the *order* of the iteration method (Kreyszig, 1999). For the analysis of the link between the order and the speed of convergence, define the departure of the solution in iteration step n from the true solution $\boldsymbol{w}_{\text{opt}}$ by the error $\boldsymbol{\varepsilon}[n]$. Thus,

$$\boldsymbol{\varepsilon}[n] = \boldsymbol{w}[n] - \boldsymbol{w}_{\text{opt}}. \tag{4.9}$$

Then according to Deuflhard and Hohmann (2002), a sequence $\boldsymbol{w}[0], \boldsymbol{w}[1], \ldots$ converges of order $p \geq 1$ if there exists a constant M such that

$$\|\boldsymbol{\varepsilon}[n+1]\| \leq M \|\boldsymbol{\varepsilon}[n]\|^p, \quad M \in \mathbb{R}_{\geq 0}. \tag{4.10}$$

In practice, we frequently encounter iteration processes of

1. *linear convergence*, in which case $p = 1$ and $M < 1$ in Eq. (4.10) and
2. *quadratic convergence*, in which case $p = 2$ in Eq. (4.10).

Obviously, we prefer algorithms with a higher order because then the desired numerical accuracy of the solution is reached in just a small number of iterations. For example, the typical behavior of iteration methods of quadratic convergence is that the number of significant digits of the computed solution is roughly doubled from one iteration step to the next (Deuflhard and Hohmann, 2002, Kreyszig, 1999).

4.2.4 Termination Criterion

Of great practical importance is the question of when to stop the iteration procedure. We can base one plausible *termination criterion* on reaching the solution

with a desired accuracy, i. e. we quit our computations in step $n+1$ and take the value of that iteration step $\boldsymbol{w}[n+1]$ as the solution to Eq. (4.6) when either the *relative error* or the *absolute error* becomes sufficiently small (Bartsch, 1999). More specifically, for a chosen relative error δ_{rel}, we stop when it holds that

$$\|\boldsymbol{w}[n+1] - \boldsymbol{w}[n]\| \leq \|\boldsymbol{w}[n+1]\|\,\delta_{\text{rel}}, \quad \delta_{\text{rel}} \in \mathbb{R}_{>0}, \tag{4.11}$$

whereas for an absolute error δ_{abs}, we stop when

$$\|\boldsymbol{w}[n+1] - \boldsymbol{w}[n]\| \leq \delta_{\text{abs}}, \quad \delta_{\text{abs}} \in \mathbb{R}_{>0}. \tag{4.12}$$

In both equations, $\|\cdot\|$ denotes a suitable vector norm, e. g. the Euclidean norm defined as

$$\|\boldsymbol{w}\| = \sqrt{{w_1}^2 + \cdots + {w_N}^2}. \tag{4.13}$$

As pointed out in Kreyszig (1999), such termination criteria do not imply convergence. Moreover, Eqs. (4.11) and (4.12) are blind to changes in sign from $\boldsymbol{w}[n]$ to $\boldsymbol{w}[n+1]$.

Independently of reaching the solution with a desired accuracy, we should abort the iteration procedure when the *allowed number of iterations* is *exceeded*, in which case the algorithm either diverges for a given initial value $\boldsymbol{w}[0]$ or converges in an unacceptably slow way. Consequently, then the algorithm has failed and we should refrain from using the value of the last iteration as a solution to Eq. (4.6).

4.2.5 Multiple Solutions

Whenever Eq. (4.6) possesses multiple solutions, it is of interest which solution an iterative algorithm converges to for a specific initial value $\boldsymbol{w}[0]$. On the other hand, an iterative algorithm should be able to yield all possible solutions.

4.2.6 Summary

In the diagram of Fig. 4.1 we summarize the steps generally involved in iterative algorithms. From this figure we conclude that implementing iterative algorithms on computers is not too complex an issue since basically the same code is executed in each iteration, albeit on different data (Kreyszig, 1999).

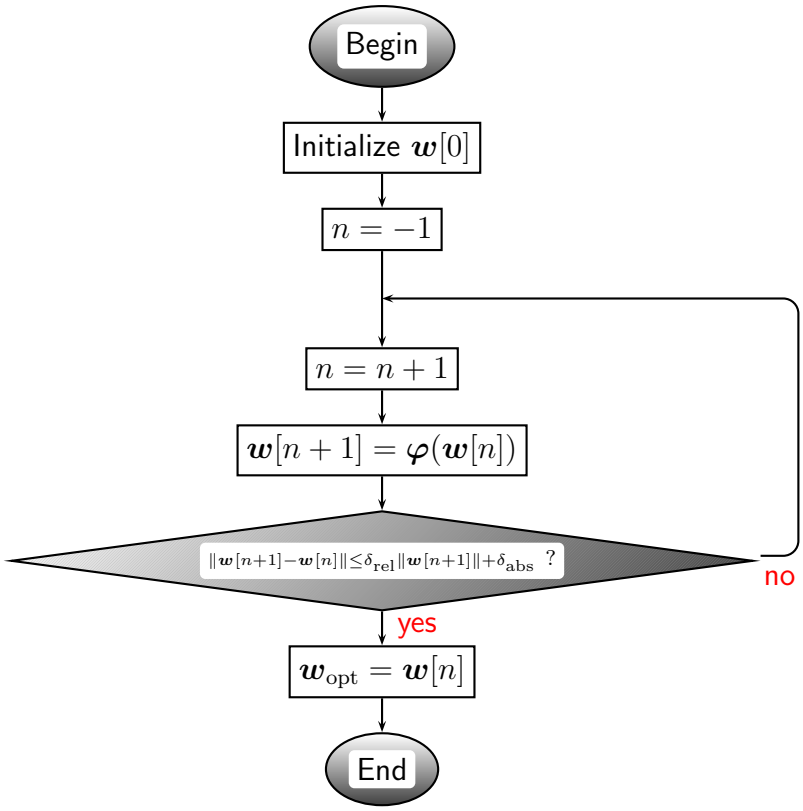


Figure 4.1: Flow diagram for iterative algorithms. Note that only one of the constants δ_{abs} or δ_{rel} is nonzero.

4.3 Fixed-Point Iteration

In the fixed-point iteration (Kreyszig, 1999, Bartsch, 1999) for solving systems of equations of the form in Eq. (4.6), we transform Eq. (4.6) algebraically to the so-called *fixed-point form*

$$\boldsymbol{w} = \boldsymbol{\varphi}(\boldsymbol{w}). \tag{4.14}$$

Usually, more than one $\boldsymbol{\varphi}(\boldsymbol{w})$ can be found for a given system (Deuflhard and Hohmann, 2002).

According to Kreyszig (1999), a vector $\boldsymbol{w}^*$ is called *fixed point* if it holds that

$$\boldsymbol{w}^* = \boldsymbol{\varphi}(\boldsymbol{w}^*). \tag{4.15}$$

In other words, a fixed point $\boldsymbol{w}^*$ is a point that remains unchanged under an application of the mapping described by $\boldsymbol{\varphi}(\boldsymbol{w})$.

Then, as suggested by Eq. (4.15), the update rule in the fixed-point iteration is

$$\boldsymbol{w}[n+1] = \boldsymbol{\varphi}(\boldsymbol{w}[n]). \tag{4.16}$$

It can be shown (Bartsch, 1999) that exactly one fixed point can be found if there is a constant L such that

$$\|\boldsymbol{J}_{\boldsymbol{\varphi}}(\boldsymbol{w})\| \leq L < 1, \tag{4.17}$$

where $\boldsymbol{J}_{\boldsymbol{\varphi}}(\boldsymbol{w})$ is the Jacobian matrix of the vector function $\boldsymbol{\varphi}(\boldsymbol{w})$ and $\|\cdot\|$ denotes a matrix norm.

The speed of convergence of the fixed-point iteration depends on the value of the constant L in Eq. (4.17). To be more exact, the less L is, the faster the algorithm converges (Bartsch, 1999).

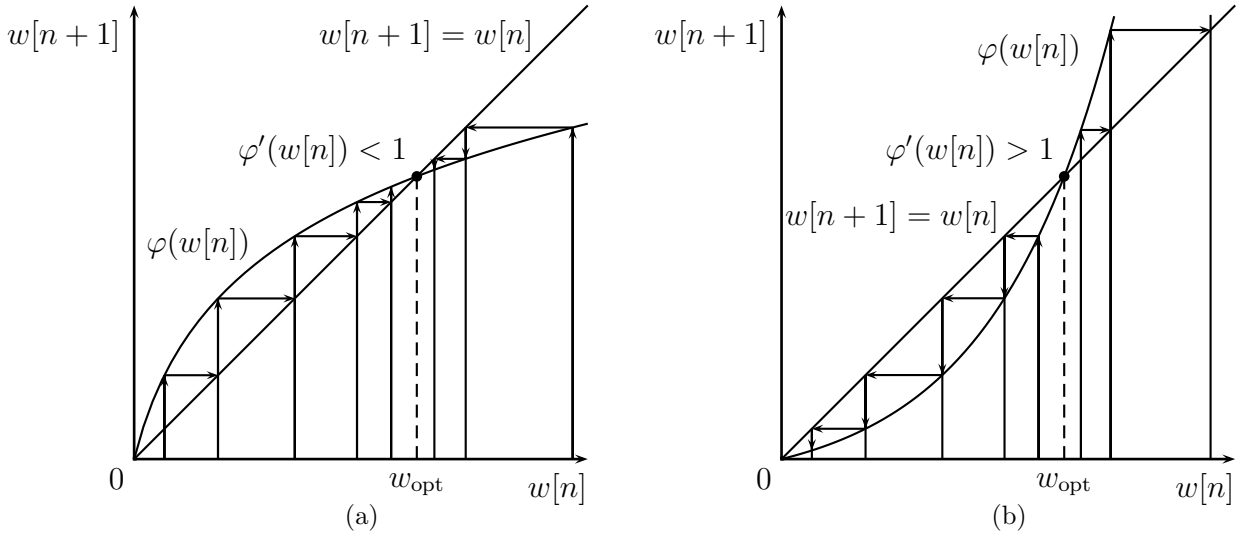


Figure 4.2: Fixed-point iteration, one-dimensional case. (a) Convergence. (b) Divergence. (Adapted from Kreyszig, 1999 and Bartsch, 1999, based on source code from `http://pstricks.de`.)

Example 4.1 (Convergence and Divergence in Fixed-Point Iteration) As an illustration of the fixed-point iteration, consider the examples in Fig. 4.2. From Fig. 4.2(a) it is obvious that for both starting values the fixed-point iteration converges toward the true solution w_{opt} because the slope of $J_{\varphi}(w) = \varphi'(w[n])$ is apparently less than unity as required by the condition in Eq. (4.17).

In contrast, both starting values in Fig. 4.2(b) correspond to unstable series since the slope of $\varphi'(w[n])$ is greater than unity. Consequently, in the latter case the algorithm does not converge. ■

4.4 Newton's Method

As another method for solving a system of N nonlinear equations in N independent variables $w_1, \ldots, w_N$ of the form

$$\boldsymbol{g}(\boldsymbol{w}) = \mathbf{0}, \tag{4.18}$$

we consider Newton's method. The idea of Newton's method is to approximate the nonlinear function $\boldsymbol{g}(\boldsymbol{w})$ in Eq. (4.18) by a *linear function* at a point $\boldsymbol{w}[0]$ (Deuflhard and Hohmann, 2002). Expanding $\boldsymbol{g}(\boldsymbol{w})$ into a power series around the starting point $\boldsymbol{w}[0]$ and neglecting higher order terms, we get from Eq. (4.18) in the vicinity of the point $\boldsymbol{w}[0]$

$$\mathbf{0} = \boldsymbol{g}(\boldsymbol{w}) \approx \underbrace{\boldsymbol{g}(\boldsymbol{w}[0]) + \boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{w}[0])\,(\boldsymbol{w} - \boldsymbol{w}[0])}_{:= \bar{\boldsymbol{g}}(\boldsymbol{w})}, \tag{4.19}$$

where $\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{w}[0])$ is the Jacobian matrix of $\boldsymbol{g}(\boldsymbol{w})$ evaluated at the point $\boldsymbol{w}[0]$ and $\bar{\boldsymbol{g}}(\boldsymbol{w})$ is the linear approximation of $\boldsymbol{g}(\boldsymbol{w})$ at the point $\boldsymbol{w}[0]$. The solution $\boldsymbol{w}[1]$ of the equation

$$\bar{\boldsymbol{g}}(\boldsymbol{w}) = \mathbf{0} \tag{4.20}$$

is then

$$\boldsymbol{w}[1] = \boldsymbol{w}[0] - \boldsymbol{J}_{\boldsymbol{g}}{}^{-1}(\boldsymbol{w}[0]) \cdot \boldsymbol{g}(\boldsymbol{w}[0]) \qquad \text{for } \det \boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{w}[0]) \neq 0. \tag{4.21}$$

In a similar fashion, Newton's method now consists of iterating

$$\boldsymbol{w}[n+1] = \boldsymbol{w}[n] + \Delta\boldsymbol{w}[n], \tag{4.22}$$

where instead of actually inverting the Jacobian matrix $\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{w}[n])$ as suggested by Eq. (4.21), we obtain the so-called *Newton correction* $\Delta\boldsymbol{w}[n]$ by first solving the system of linear equations

$$\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{w}[n]) \cdot \Delta\boldsymbol{w}[n] = -\boldsymbol{g}(\boldsymbol{w}[n]), \tag{4.23}$$

by any standard method (Deuflhard and Hohmann, 2002, Bartsch, 1999).

It can be shown that locally Newton's method converges of second order (Deuflhard and Hohmann, 2002).

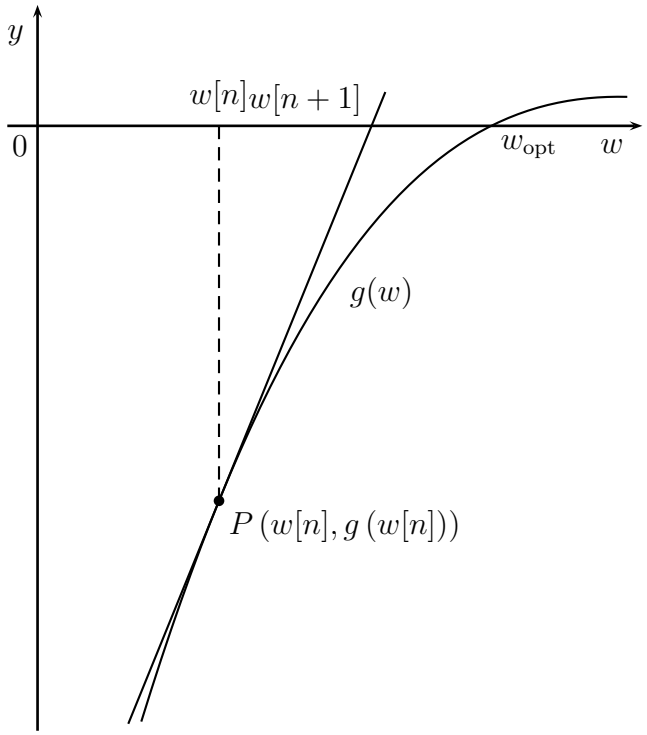


Figure 4.3: Illustration of Newton's method, adapted from Bartsch (1999).

Example 4.2 (Newton's Method as a Linear Approximation)
Fig. 4.3 illustrates Newton's method for a single nonlinear function $g(w)$ and one independent variable w. In this case, $N = 1$ in Eq. (4.18), and Eq. (4.22) reduces to

$$w[n+1] = w[n] - \left.\frac{1}{\frac{\mathrm{d}g(w)}{\mathrm{d}w}}\right|_{w=w[n]} \cdot g(w[n]), \tag{4.24}$$

where the derivative of $g(w)$ is evaluated at the current $w[n]$. As we know, the Taylor series expansion of the function $g(w)$ in the vicinity of the point $w[n]$ is

$$g(w) \approx g(w[n]) + \left.\frac{\mathrm{d}g(w)}{\mathrm{d}w}\right|_{w=w[n]} \cdot (w - w[n])\,. \tag{4.25}$$

Equating Eq. (4.25) to zero and denoting the solution by $w[n+1]$, we get the update rule in Eq. (4.24) after straightforward algebraic manipulations. ■

As pointed out by Kreyszig (1999) for the one-dimensional case in the previous example, we must pay attention to situations where the linear system of Eq. (4.23) is *ill-conditioned*.

4.5 Method of Steepest Descent

The last iterative algorithm we cover is the *method of steepest descent*. It is a popular and old algorithm (Kreyszig, 1999) for finding points where the gradient of some cost function $\mathcal{I}(\boldsymbol{w})$ is zero

$$\operatorname{grad}\mathcal{I}(\boldsymbol{w}) = \frac{\partial\mathcal{I}(\boldsymbol{w})}{\partial\boldsymbol{w}} \stackrel{!}{=} \mathbf{0}. \tag{4.26}$$

More precisely, to find potential minima of the cost function $\mathcal{I}(\boldsymbol{w})$, in each iteration step we subtract[3] from the current vector $\boldsymbol{w}[n]$ a vector pointing in the direction of the gradient of the cost function $\mathcal{I}(\boldsymbol{w})$

$$\boldsymbol{w}[n+1] = \boldsymbol{w}[n] - \alpha[n]\frac{\partial\mathcal{I}(\boldsymbol{w})}{\partial\boldsymbol{w}}\bigg|_{\boldsymbol{w}=\boldsymbol{w}[n]}, \qquad n = 0, 1, \ldots, \tag{4.27}$$

where the gradient $\frac{\partial\mathcal{I}(\boldsymbol{w})}{\partial\boldsymbol{w}}$ is evaluated at the point $\boldsymbol{w}[n]$ (Hyvärinen et al., 2001, Haykin, 2002) and the positive scalar $\alpha[n]$ is called *step-size parameter* or *learning rate*.

Geometrically, since the gradient always points in the direction of maximum increase of a function, the procedure described in Eq. (4.27) corresponds to a stepwise "downhill" move along the steepest path on the hyperplane described by the cost function $\mathcal{I}(\boldsymbol{w})$. As noted in Hyvärinen et al. (2001), this yields a local extremum in the neighborhood of the starting vector $\boldsymbol{w}[0]$.

4.5.1 Convergence Speed and Step-Size Parameter

The convergence of the method of steepest descent can be shown to be *linear* (cf. Section 4.2.3). Actually, in the vicinity of the optimum $\boldsymbol{w}_{\text{opt}}$ the speed of convergence depends (1) on the Hessian matrix $\boldsymbol{H}_{\mathcal{I}}(\boldsymbol{w})$ of the cost function $\mathcal{I}(\boldsymbol{w})$

$$\boldsymbol{H}_{\mathcal{I}}(\boldsymbol{w}) = \frac{\partial^2\mathcal{I}(\boldsymbol{w})}{\partial\boldsymbol{w}^2} = \begin{bmatrix} \frac{\partial^2\mathcal{I}}{\partial w_1^2} & \cdots & \frac{\partial^2\mathcal{I}}{\partial w_1 \partial w_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2\mathcal{I}}{\partial w_1 \partial w_N} & \cdots & \frac{\partial^2\mathcal{I}}{\partial w_N^2} \end{bmatrix}, \tag{4.28}$$

which measures the curvature of the cost function, and (2) on the step-size parameter $\alpha[n]$ (Hyvärinen et al., 2001). Consequently, stability and convergence behavior are governed only by the value of the step-size parameter α if the cost

[3] Maxima are found similarly by *adding* a vector pointing in the direction of the gradient in Eq. (4.27).

function is fixed. Specifically, in choosing the step-size parameter too small, we might not get convergence in an acceptable number of iteration steps. On the other hand, if the step-size parameter is too large, the algorithm will become unstable.

Moreover, in stationary environments the step-size parameter should be dampened with time (Hyvärinen et al., 2001)

$$\alpha[n] = \frac{\beta}{\beta + n}, \tag{4.29}$$

where for example $\beta = 100$.

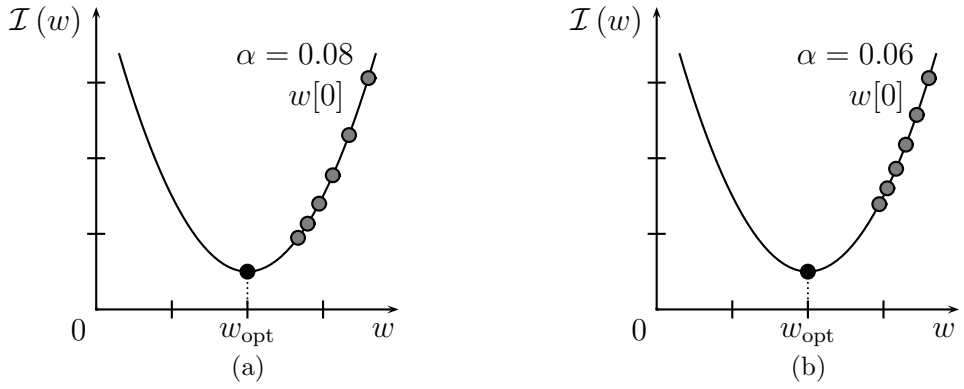


Figure 4.4: Illustration of the method of steepest descent, one-dimensional case. (a) Larger step-size parameter. (b) Smaller step-size parameter.

Example 4.3 (Convergence speed of the Method of Steepest Descent) In Fig. 4.4, we compare the convergence speed of the method of steepest descent optimizing a given cost function $\mathcal{I}(w)$ for two different values of the step-size parameter $\alpha[n] = \alpha$, which is constant in this case. Here, both iterations start from the same point $w[0]$. After the number of iteration steps considered in the example, the update rule with the larger step-size parameter $\alpha = 0.08$ produces a solution that is obviously much closer to the optimum w_{opt} than the one with the smaller step-size parameter $\alpha = 0.06$. ■

4.5.2 Application to Specific Cost Functions

In many of the algorithms for solving the Independent Component Analysis model parameters, the cost function $\mathcal{I}(\boldsymbol{w})$ has the form of a mathematical expectation (cf. Section 1.2)

$$\mathcal{I}(\boldsymbol{w}) = \mathcal{E}\{f(\boldsymbol{w}, \boldsymbol{x})\}, \tag{4.30}$$

where the *random* vector $\boldsymbol{x}$ describes the available input data of the algorithm and the expectation is computed with respect to the unknown joint p. d. f. $p_{\boldsymbol{x}}(\boldsymbol{x})$ of the random vector of the input data. Thus, the update rule of the method of steepest descent can be written as

$$\boldsymbol{w}[n+1] = \boldsymbol{w}[n] - \alpha[n] \frac{\partial \mathcal{E}\{f(\boldsymbol{w}, \boldsymbol{x})\}}{\partial \boldsymbol{w}}\bigg|_{\boldsymbol{w}=\boldsymbol{w}[n]} . \tag{4.31}$$

4.5.2.1 Batch Learning

Note that in practice, the expectation in Eq. (4.31) can be estimated as discussed in Section 1.14 from the samples that constitute the available data set. In the literature, this is called *batch learning* (Hyvärinen et al., 2001).

4.5.2.2 Online Learning, Stochastic Gradient Algorithms

Sometimes, the batch learning paradigm introduced in the previous section may not be the optimal choice. In particular, consider the following (Hyvärinen et al., 2001):

- If employed in a nonstationary environment, the algorithm should allow for (fast) *tracking* of the time-varying statistical properties of the input data
- Estimating the mathematical expectation in Eq. (4.31) at each iteration step might be (1) inappropriate for computational reasons or (2) impossible if the input data cannot be presented to the algorithm as a whole.

These are the application ranges of the so-called *online algorithm* (Hyvärinen et al., 2001), where in the update rule described by Eq. (4.31) we use only *one sample at a time*, usually the most recent one, instead of estimating the mathematical expectation

$$\boldsymbol{w}[n+1] = \boldsymbol{w}[n] - \alpha[n] \frac{\partial f(\boldsymbol{w}, \boldsymbol{x})}{\partial \boldsymbol{w}}\bigg|_{\boldsymbol{w}=\boldsymbol{w}[n]} . \tag{4.32}$$

This results in quite random directions of the gradient (Haykin, 2002, Hyvärinen et al., 2001), for which reason the update rule in Eq. (4.32) is often called *stochastic gradient algorithm*. Yet, under certain conditions (Hyvärinen et al., 2001), the stochastic gradient algorithm converges onto the same solution as the corresponding batch algorithm.

The computational complexity of stochastic gradient algorithms is considerably reduced with respect to the corresponding batch algorithms. On the other hand, many more iteration steps are needed for convergence (Hyvärinen et al., 2001).

4.5.2.3 Shuffling of the Input Data

If applying the stochastic gradient algorithm in cases where the training set *is* available *a priori*, we might need to cycle over the set more than once in order for the algorithm to converge. Then, it is preferable to shuffle the samples, as pointed out by Hyvärinen et al. (2001).

4.6 Constrained Optimization

4.6.1 Method of Lagrange Multipliers

The most classical approach to solving constrained optimization problems is the *method of Lagrange multipliers* (Haykin, 2002), which is basically a technique for converting constrained optimization problems into unconstrained optimization problems with additional variables. See the reference for more details.

Example 4.4 (Method of Lagrange Multipliers)
As an example of an application of the method of Lagrange multipliers, let us suppose we are interested in the maxima of the cost function $\mathcal{I}(\boldsymbol{w})$ under the constraint that the norm of the vector $\boldsymbol{w}$ be unity

$$\arg\max_{\boldsymbol{w}} \mathcal{I}(\boldsymbol{w}) \tag{4.33a}$$

$$\|\boldsymbol{w}\|^2 \stackrel{!}{=} 1. \tag{4.33b}$$

First, we bring the constraint in Eq. (4.33b) to the form

$$C(\boldsymbol{w}) = 0, \tag{4.34}$$

so that it reads now

$$\|\boldsymbol{w}\|^2 - 1 = 0. \tag{4.35}$$

Next, we form a new function $q(\boldsymbol{w}, \lambda_{\mathrm{MP}})$ as a combination of the original cost function $\mathcal{I}(\boldsymbol{w})$ and the constraint in Eq. (4.35) (Haykin, 2002)

$$q(\boldsymbol{w}, \lambda_{\mathrm{MP}}) = \mathcal{I}(\boldsymbol{w}) + \lambda_{\mathrm{MP}} \left(\|\boldsymbol{w}\|^2 - 1 \right), \tag{4.36}$$

where the new scalar variable λ_{MP} is called the *Lagrange multiplier*. Maximization of the new objective function $q(\boldsymbol{w}, \lambda_{\mathrm{MP}})$ with respect to the vector $\boldsymbol{w}$ yields the condition

$$\frac{\partial \mathcal{I}(\boldsymbol{w})}{\partial \boldsymbol{w}} + 2\lambda_{\mathrm{MP}}\boldsymbol{w} \stackrel{!}{=} \mathbf{0}. \tag{4.37}$$

Eq. (4.37) is called the *adjoint equation*. The system of equations in Eq. (4.37) along with the constraint in Eq. (4.35) constitute a new optimization problem in the variables $\boldsymbol{w}$ and λ_{MP} that can be solved by any standard method.

To conclude, note from Eq. (4.37) that for the constraint optimization problem in Eq. (4.33), the gradient always points in the direction of the vector $\boldsymbol{w}$. ■

4.6.2 Projection on the Constraint Set

A much simpler method for solving constrained optimization problems, which more often than not suffices for the purposes of Independent Component Analysis, is the *method of projection on the constraint set* (Hyvärinen et al., 2001). In the projection on the constraint set, we obtain an intermediate result $\boldsymbol{w}'[n]$ by first applying an update rule $\boldsymbol{\varphi}(\boldsymbol{w}[n])$ as if we were to find an extremum in an unconstrained optimization problem

$$\boldsymbol{w}'[n] = \boldsymbol{\varphi}(\boldsymbol{w}[n]). \tag{4.38a}$$

But after this, we project $\boldsymbol{w}'[n]$ orthogonally onto the constraint set. This is applicable for instance when the constraint consists of the requirement that the solution vector $\boldsymbol{w}_{\mathrm{opt}}$ be of unit norm. In this case, the constraint set is the N-dimensional unit sphere, where N is the length of the vector $\boldsymbol{w}[n]$, and the projection corresponds to dividing the vector $\boldsymbol{w}'[n]$ by its Euclidean norm in each iteration step

$$\boldsymbol{w}[n+1] = \frac{\boldsymbol{w}'[n]}{\|\boldsymbol{w}'[n]\|}. \tag{4.38b}$$

Example 4.5 (Projection on the Constraint Set)
In Fig. 4.5, all vectors involved in Eq. (4.38) are plotted for a twodimensional example. To be more exact, the iteration starts with the vector $\boldsymbol{w}[n]$. First, the vector $\boldsymbol{w}'[n]$ is computed according to an update rule that adds a vector $\boldsymbol{\Delta w}$ to $\boldsymbol{w}[n]$. Then, the vector is divided by its norm to yield $\boldsymbol{w}[n+1]$ as the result in this iteration step. Thus, in this example the constraint set is the unit circle. ■

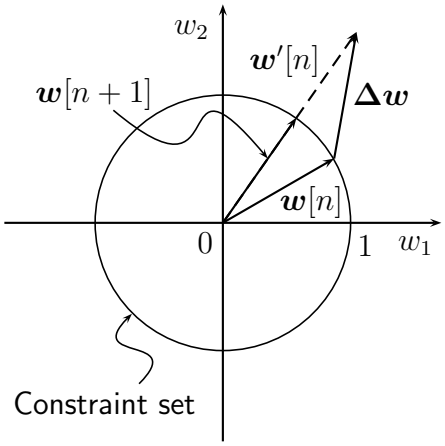


Figure 4.5: Projection on the constraint set.

Appendix: Vector and Matrix Derivatives

In this appendix, we give some examples of gradient vectors, Hessian matrices, and matrix gradients of some functions needed in Part II for the derivations of various algorithms.

Note that the gradient of composite functions is computed along the lines of the derivative of a composite function of one variable and likewise for the gradient of products and quotients of functions (Hyvärinen et al., 2001).

Gradient Vector and Hessian Matrix of Scalar Functions

Example 4.6 (Inner Product)
As simple examples of gradients of a scalar function consider the gradient of the inner product between the vector of independent variables $\boldsymbol{w}$ and a vector of constants $\boldsymbol{a}$ of suitable size

$$\frac{\partial \boldsymbol{a}^{\mathrm{T}}\boldsymbol{w}}{\partial \boldsymbol{w}} = \frac{\partial \boldsymbol{w}^{\mathrm{T}}\boldsymbol{a}}{\partial \boldsymbol{w}} = \boldsymbol{a}. \tag{4.39}$$

Since the gradient in Eq. (4.39) is constant, the Hessian matrix is the zero matrix

$$\frac{\partial^2 \boldsymbol{a}^{\mathrm{T}}\boldsymbol{w}}{\partial \boldsymbol{w}^2} = \frac{\partial^2 \boldsymbol{w}^{\mathrm{T}}\boldsymbol{a}}{\partial \boldsymbol{w}^2} = \boldsymbol{0}. \tag{4.40}$$

■

Example 4.7 (Quadratic Form)
Evaluating the gradient of a quadratic form in the variables w_i with respect to this

variables gives

$$\frac{\partial \boldsymbol{w}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{w}}{\partial \boldsymbol{w}} = \boldsymbol{A}\boldsymbol{w} + \boldsymbol{A}^{\mathrm{T}}\boldsymbol{w}. \tag{4.41}$$

From the gradient in Eq. (4.41), we get for the Hessian of the quadratic form

$$\frac{\partial^2 \boldsymbol{w}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{w}}{\partial \boldsymbol{w}^2} = \boldsymbol{A} + \boldsymbol{A}^{\mathrm{T}}. \tag{4.42}$$

Here, the matrix $\boldsymbol{A}$ is assumed square, yet not necessarily symmetric. ■

Matrix Gradients of Scalar Functions

Let $\frac{\partial \mathcal{I}(\boldsymbol{W})}{\partial \boldsymbol{W}}$ denote the *matrix gradient* of the function $\mathcal{I}(\boldsymbol{W})$ with respect to the $M \times N$ matrix

$$\boldsymbol{W} = \begin{bmatrix} w_{11} & \cdots & w_{1N} \\ \vdots & \ddots & \vdots \\ w_{M1} & \cdots & w_{MN} \end{bmatrix}. \tag{4.43}$$

Here, the independent variables w_{ij} are arranged in matrix form for convenience. The matrix gradient is defined as the matrix of partial derivatives (Hyvärinen et al., 2001)

$$\frac{\partial \mathcal{I}(\boldsymbol{W})}{\partial \boldsymbol{W}} = \begin{bmatrix} \frac{\partial \mathcal{I}}{\partial w_{11}} & \cdots & \frac{\partial \mathcal{I}}{\partial w_{1N}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathcal{I}}{\partial w_{M1}} & \cdots & \frac{\partial \mathcal{I}}{\partial w_{MN}} \end{bmatrix}. \tag{4.44}$$

Example 4.8 (Quadratic Form)
Consider as a cost function $\mathcal{I}(\boldsymbol{A})$ the quadratic form in the variables w_i. Then, the matrix gradient of the quadratic form with respect to the square matrix $\boldsymbol{A}$ is

$$\frac{\partial \boldsymbol{w}^{\mathrm{T}} \boldsymbol{A} \boldsymbol{w}}{\partial \boldsymbol{A}} = \boldsymbol{w}\boldsymbol{w}^{\mathrm{T}}. \tag{4.45}$$

This result is not to be confused with the derivative with respect to the variables w_i in Eq. (4.41). ■

Example 4.9 (Determinant of a Matrix)
Using some properties of the determinant of a matrix as shown in Hyvärinen et al. (2001), we get for the matrix gradient of the determinant of a matrix $\boldsymbol{W}$ with

respect to that matrix

$$\frac{\partial}{\partial \boldsymbol{W}} \det \boldsymbol{W} = \left(\boldsymbol{W}^{\mathrm{T}}\right)^{-1} \det \boldsymbol{W}. \tag{4.46}$$

Here is is assumed that the inverse of the transpose of $\boldsymbol{W}$ exists. ■

Example 4.10 (Maximum Likelihood Estimation)
The following identity will be needed in Section 7 for the derivation of the natural logarithm of the likelihood function.

$$\frac{\partial}{\partial \boldsymbol{W}} \ln \det \boldsymbol{W} = \left(\boldsymbol{W}^{\mathrm{T}}\right)^{-1}. \tag{4.47}$$

For a proof of Eq. (4.47) see Hyvärinen et al. (2001). ■

Part II

Blind Source Separation

5 Introduction to Blind Source Separation and Independent Component Analysis

In this chapter, we first set the scene of the blind source separation problem. Then, Independent Component Analysis is introduced as a widely used technique for solving the blind source separation problem. Finally, we briefly discuss some applications of Independent Component Analysis.

5.1 Blind Source Separation

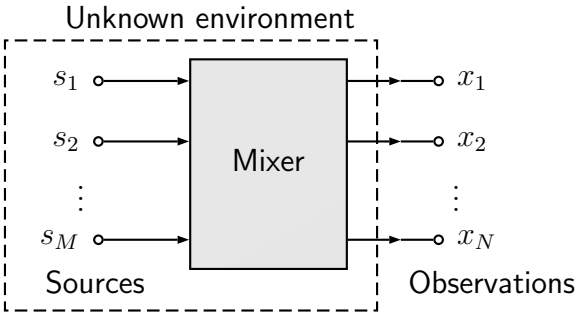


Figure 5.1: Generation of observations x_i in an unknown environment, adapted from Haykin (2002).

To illustrate the problem of blind source separation, consider the *generative model* (Hyvärinen et al., 2001) depicted in Fig. 5.1. Here, we observe N random variables

$$\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^{\mathrm{T}} \tag{5.1}$$

at the output of a mixer whose input are M random variables

$$\boldsymbol{s} = \begin{bmatrix} s_1 & \cdots & s_M \end{bmatrix}^{\mathrm{T}}. \tag{5.2}$$

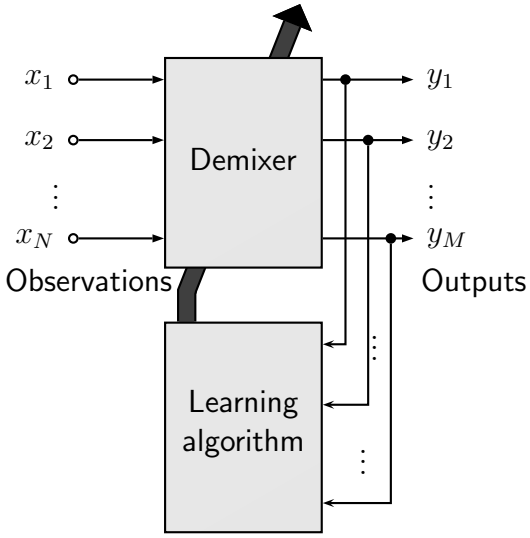


Figure 5.2: Adaptive estimation of the original sources signals from observations x_i, adapted from Haykin (2002).

In *blind source separation* (BSS), the task is to unravel the effects of the mixing process from knowledge of the output variables $\boldsymbol{x}$ alone (Haykin, 2002). More specifically, given only the mixtures $\boldsymbol{x}$, we want to find estimates

$$\boldsymbol{y} = \begin{bmatrix} y_1 & \cdots & y_M \end{bmatrix}^{\mathrm{T}} \tag{5.3}$$

that approximate the original sources $\boldsymbol{s}$ well, both the details of the mixing process and the exact statistical properties of the original input variables $\boldsymbol{s}$ being unknown, which explains the usage of the adjective "blind". This is shown in Fig. 5.2.

5.2 General Mixing Process

Not surprisingly, the mixing process determines how difficult the estimation of the original variables $\boldsymbol{s}$ from the observable random variables $\boldsymbol{x}$ will really be. In particular, the following general properties of the mixing process have to be considered (Haykin, 2002):

Linear vs. nonlinear mixing Each observable random variable x_i at the output of the mixer can be either a linear combination of the source variables

$$x_i = \sum_{j=1}^{M} a_{ij} s_j \tag{5.4}$$

or a nonlinear function of the source variables. In the latter case, x_i might

be a nonlinear function of a linear combination of the source variables

$$x_i = g\left(\sum_{j=1}^{M} a_{ij} s_j\right), \tag{5.5}$$

or even an arbitrary nonlinear function of some or all source variables

$$x_i = g\left(s_1, \ldots, s_M\right), \tag{5.6}$$

with $g(\cdot)$ a nonlinear scalar function (Hyvärinen et al., 2001).

Time-varying vs. fixed mixing The mixing is referred to as time-varying if its properties change with time. In other words, the observable random variables are produced in a nonstationary mixing environment.

Nonconvolutive vs. convolutive mixer A convolutive mixer is a mixer possessing memory, i. e. the output variables viewed as observations of a stochastic process (Papoulis, 1991, Hyvärinen et al., 2001) are a function of the input variables at different times. On the other hand, the much simpler memoryless, nonconvolutive case produces instantaneous mixtures depending on the current values only.

Noiseless environment vs. presence of noise Here, the noise can consist of noise as input to the mixer, or it can be introduced inside the mixing process.

Number of mixtures vs. number of sources Depending on the method used for solving the blind source separation problem, the relation between the number of observable random variables and the number of sources determines the number of estimates that can be inferred from the observable random variables.

5.3 Independent Component Analysis

Independent Component Analysis (ICA) is a statistical technique, perhaps the most widely used, for solving the blind source separation problem (Hyvärinen et al., 2001). In this section, we present the basic Independent Component Analysis model and show under which conditions its parameters can be estimated.

5.3.1 The Mixing Process

The generative model used in Independent Component Analysis assumes a mixing process of the very simplest form (cf. Section 5.2). In particular, the mixing is (Hyvärinen et al., 2001)

- linear,
- fixed, i. e. time-invariant,
- memoryless, and
- free of known noise sources.

Consequently, the mixing process can be written in matrix notation as

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}, \tag{5.7}$$

where the matrix $\boldsymbol{A}$ is called the *mixing matrix*. Note that in Independent Component Analysis, the entries of the mixing matrix are real-valued coefficients a_{ij}.

5.3.2 The Unmixing Process

On the assumption that the mixing matrix $\boldsymbol{A}$ is nonsingular, the original independent components $\boldsymbol{s}$ could be recovered from the observable random variables $\boldsymbol{x}$ by matrix inversion—if the mixing matrix $\boldsymbol{A}$ were known. With Eq. (5.7), we get

$$\boldsymbol{s} = \boldsymbol{B}\boldsymbol{x} = \underbrace{\boldsymbol{B}\boldsymbol{A}}_{\stackrel{!}{=}\boldsymbol{I}}\boldsymbol{s} = \boldsymbol{s}, \tag{5.8}$$

when the *unmixing matrix* $\boldsymbol{B}$ equals the inverse of the mixing matrix, i. e.

$$\boldsymbol{B} = \boldsymbol{A}^{-1}. \tag{5.9}$$

As a consequence, even if the true mixing matrix $\boldsymbol{A}$ is unknown in practice, estimates of the independent components can be obtained as linear combinations of the observations $\boldsymbol{x}$. Hence, the task of Independent Component Analysis can be interpreted as the design of methods or algorithms that are able to determine an unmixing matrix $\boldsymbol{B}$ such that the estimates

$$\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x} \tag{5.10a}$$

or, equivalently

$$y_i = \boldsymbol{b}_i^{\mathrm{T}} \boldsymbol{x}, \qquad i = 1, \ldots, N, \tag{5.10b}$$

approximate the unknown sources as close as possible (Hyvärinen et al., 2001).

5.3.3 Conditions in Independent Component Analysis

For the Independent Component Analysis model to be identifiable, three assumptions must be met (Hyvärinen et al., 2001):

- Most importantly, the unknown sources s_j have to be mutually *statistically independent* (cf. Section 1.3.2), which explains the name "Independent Component Analysis". In this context, the sources s_j are also called independent components.
- At most one independent component is allowed to have a Gaussian p. d. f.
- The number of sources M should equal the number of observations N.

5.3.4 Ambiguities in the Independent Component Analysis Model Estimation

The following ambiguities cannot be eliminated by any method for solving the Independent Component Analysis model (Hyvärinen et al., 2001):

- The true variance of the independent components cannot be determined. To explain, we can rewrite the mixing in Eq. (5.7) in the form

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{A}\boldsymbol{s} \\ &= \sum_{j=1}^{N} \boldsymbol{a}_j s_j, \end{aligned} \tag{5.11}$$

  where $\boldsymbol{a}_j$ denotes the jth column of the mixing matrix $\boldsymbol{A}$. Since both the coefficients $\boldsymbol{a}_j$ of the mixing matrix and the independent components s_j are unknown, we can transform Eq. (5.11)

$$= \sum_{j=1}^{N} \underbrace{\frac{1}{\chi_j} \boldsymbol{a}_j}_{\bar{\boldsymbol{a}}_j} \underbrace{\chi_j s_j}_{\bar{s}_j}, \qquad \text{where } \chi_j \neq 0 \tag{5.12}$$

so as to obtain new mixing coefficients $\bar{\boldsymbol{a}}_j$ with corresponding new independent components $\bar{s}_j$

$$= \sum_{j=1}^{N} \bar{\boldsymbol{a}}_j \bar{s}_j. \tag{5.13}$$

- The order of the estimated independent components is unspecified. Formally, introducing a permutation matrix $\boldsymbol{P}$ and its inverse into the mixing process in Eq. (5.7)

$$\begin{aligned} \boldsymbol{x} &= \boldsymbol{A}\boldsymbol{s} \\ &= \underbrace{\boldsymbol{A}\boldsymbol{P}^{-1}}_{\bar{\boldsymbol{A}}} \underbrace{\boldsymbol{P}\boldsymbol{s}}_{\bar{\boldsymbol{s}}}, \end{aligned} \tag{5.14}$$

we get the equivalent mixing model

$$= \bar{\boldsymbol{A}}\bar{\boldsymbol{s}}. \tag{5.15}$$

To conclude, when rating the performance of any method for solving the Independent Components Analysis model, we cannot expect the estimated unmixing matrix $\boldsymbol{B}$ to match the mixing matrix $\boldsymbol{A}$ exactly.[1] In other words, the matrix product of the unmixing matrix and the mixing matrix will not generally equal the identity matrix as it should by Eq. (5.8). Rather, the result will ideally produce a matrix that has only one nonzero element per row and column.

Since the true variance of the independent components s_j cannot be determined anyway, we can just fix them to unity, i. e.

$$\sigma_{s_j}^2 \stackrel{!}{=} 1. \tag{5.16a}$$

As a consequence, the variance of their estimates y_i is unity as well

$$\sigma_{y_i}^2 \stackrel{!}{=} 1. \tag{5.16b}$$

Note that this still leaves the uncertainty about the sign, which is irrelevant in most applications, though.

[1] Note that in computer experiments, we obviously *know* the mixing matrix.

The estimate y_i is computed as a linear combination of the original independent components s_j

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{B}\boldsymbol{x} \\ &= \underbrace{\boldsymbol{B}\boldsymbol{A}}_{\boldsymbol{Q}}\boldsymbol{s} \end{aligned} \tag{5.17}$$

$$y_i = \boldsymbol{q}_i^{\mathrm{T}}\boldsymbol{s} \tag{5.18}$$

with the coefficients compiled in the vector $\boldsymbol{q}_i$. Since the random variables s_j are statistically independent, with Eq. (5.16a) we get for the variance of the estimate y_i

$$\sigma_{y_i}^2 = q_{i1}^2\sigma_{s_1}^2 + \cdots + q_{in}^2\sigma_{s_n}^2 \tag{5.19a}$$

$$= q_{i1}^2 + \cdots + q_{in}^2 \tag{5.19b}$$

$$= \|\boldsymbol{q}_i\|^2. \tag{5.19c}$$

Finally, it follows from Eq. (5.16b) and Eq. (5.19c) that (Hyvärinen et al., 2001)

$$\|\boldsymbol{q}_i\|^2 \stackrel{!}{=} 1. \tag{5.20}$$

5.3.5 Sphering Transformation

The linear transformation of the mixtures $\boldsymbol{x}$ given by

$$\boldsymbol{z} = \boldsymbol{V}\boldsymbol{x} \tag{5.21}$$

is called a *sphering* (or, *whitening*) *transformation* if the components of the random vector $\boldsymbol{z}$ are mutually uncorrelated (cf. Section 1.3.1) and have unit variance (Hyvärinen et al., 2001). Here, the matrix $\boldsymbol{V}$ is called the *sphering* (or, *whitening*) *matrix*. In other words, the correlation matrix $\boldsymbol{C}_{\boldsymbol{z}}$ of the random vector $\boldsymbol{z}$ equals the identity matrix if $\boldsymbol{z}$ is sphered

$$\mathcal{E}\{\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}}\} = \boldsymbol{I}. \tag{5.22}$$

One can say that sphering removes the effects of first- and second-order statistics (Hyvärinen et al., 2001).

5.3.5.1 Sphering as a Preprocessing Step in ICA

For the sphered vector $\boldsymbol{z}$

$$\begin{aligned}\boldsymbol{z} &= \boldsymbol{V}\boldsymbol{x} \\ &= \boldsymbol{V}\boldsymbol{A}\boldsymbol{s},\end{aligned} \tag{5.23}$$

the Independent Component Analysis model still holds, albeit with a different mixing matrix given by $\boldsymbol{V}\boldsymbol{A}$. Hence, Independent Component Analysis can be performed on the sphered vector $\boldsymbol{z}$ just as well.

The sphering transformation is treated in this chapter not because is actually solves the problem of estimating the Independent Component Analysis model, but because it makes it a lot easier.

More specifically, using a sphering transformation as a preprocessing step in Independent Component Analysis significantly reduces the degrees of freedom of the estimation problem (Hyvärinen et al., 2001) because after the sphering transformation in Eq. (5.21), the unmixing matrix $\boldsymbol{W}$, which gives the estimates of the independent components in the end

$$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{z}, \tag{5.24}$$

has to be orthogonal. To see the reason for this, note that from Eq. (5.23) and the assumption that the independent components s_j are statistically independent and have unit variance it follows that

$$\begin{aligned}\mathcal{E}\left\{\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}}\right\} &= \mathcal{E}\left\{\left(\boldsymbol{V}\boldsymbol{A}\right)\boldsymbol{s}\boldsymbol{s}^{\mathrm{T}}\left(\boldsymbol{V}\boldsymbol{A}\right)^{\mathrm{T}}\right\} \\ &= \left(\boldsymbol{V}\boldsymbol{A}\right)\underbrace{\mathcal{E}\left\{\boldsymbol{s}\boldsymbol{s}^{\mathrm{T}}\right\}}_{=\boldsymbol{I}}\left(\boldsymbol{V}\boldsymbol{A}\right)^{\mathrm{T}}.\end{aligned} \tag{5.25}$$

Combining Eq. (5.25) with Eq. (5.22), we get

$$\left(\boldsymbol{V}\boldsymbol{A}\right)\left(\boldsymbol{V}\boldsymbol{A}\right)^{\mathrm{T}} = \boldsymbol{I}, \tag{5.26}$$

which shows that after the sphering transformation the mixing matrix $\boldsymbol{V}\boldsymbol{A}$ is indeed orthogonal.

Furthermore, since the unmixing matrix $\boldsymbol{W}$ ideally corresponds to the inverse of the mixing matrix $\boldsymbol{V}\boldsymbol{A}$, we derive from Eq. (5.26) that for sphered data the unmixing matrix $\boldsymbol{W}$ has to be orthogonal as well.

As pointed out in Hyvärinen et al. (2001), an orthogonal N-by-N matrix has

just

$$\frac{N\,(N-1)}{2} \tag{5.27}$$

degrees of freedom compared to the N^2 degrees of an arbitrary matrix of the same size. Note that in the twodimensional case ($N = 2$), an orthogonal matrix can be described by just a single parameter.

Geometrically, orthogonal matrices describe coordinate transformations preserving angles and distances, which corresponds to a rotation of the coordinate system in the multidimensional space.

5.3.5.2 Sphering Using Eigenvalue Decomposition

A sphering matrix $\boldsymbol{V}$ can always be found, e. g. from the well-known eigenvalue decomposition of the covariance matrix $\boldsymbol{C_x}$ of the zero-mean random vector $\boldsymbol{x}$

$$\boldsymbol{C_x} = \boldsymbol{E}\boldsymbol{D}\boldsymbol{E}^{\mathrm{T}}, \tag{5.28}$$

where the orthogonal matrix $\boldsymbol{E}$ contains on its columns the unit-norm eigenvectors of the covariance matrix $\boldsymbol{C_x}$ and the matrix $\boldsymbol{D}$ is a diagonal matrix whose entries are the corresponding eigenvalues, sorted in order of descending variances λ_i

$$\boldsymbol{D} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_N \end{bmatrix}, \qquad \lambda_1 > \lambda_2 > \cdots > \lambda_N. \tag{5.29}$$

Then, as can easily be verified, a suitable sphering matrix $\boldsymbol{V}$ is given by

$$\boldsymbol{V} = \boldsymbol{E}\boldsymbol{D}^{-1/2}\boldsymbol{E}^{\mathrm{T}}, \tag{5.30}$$

where

$$\boldsymbol{D}^{-1/2} = \begin{bmatrix} \frac{1}{\sqrt{\lambda_1}} & 0 & \cdots & 0 \\ 0 & \frac{1}{\sqrt{\lambda_2}} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{\sqrt{\lambda_N}} \end{bmatrix}. \tag{5.31}$$

Online sphering algorithms that avoid the direct eigenvalue decomposition of the correlation matrix are mentioned in Hyvärinen et al. (2001).

5.3.6 Constraint on Unmixing Matrix for Sphered Data

For a sphered random vector $\boldsymbol{z}$

$$\begin{aligned}\boldsymbol{z} &= \boldsymbol{V}\boldsymbol{x} \\ &= \boldsymbol{V}\boldsymbol{A}\boldsymbol{s},\end{aligned} \tag{5.32}$$

where $\boldsymbol{V}$ is the matrix of the sphering transformation, we have for the estimate of the independent component

$$y_i = \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{z} = \underbrace{\boldsymbol{w}_i^{\mathrm{T}}\left(\boldsymbol{V}\boldsymbol{A}\right)}_{\boldsymbol{q}_i^{\mathrm{T}}}\boldsymbol{s}, \tag{5.33}$$

where the vector $\boldsymbol{q}_i$ is given by

$$\boldsymbol{q}_i = \left(\boldsymbol{V}\boldsymbol{A}\right)^{\mathrm{T}}\boldsymbol{w}_i. \tag{5.34}$$

Substituting Eq. (5.34) in Eq. (5.20) yields

$$\begin{aligned}\|\boldsymbol{q_i}\|^2 &= \boldsymbol{q_i}^{\mathrm{T}}\boldsymbol{q_i} \\ &= \left[\boldsymbol{w}_i^{\mathrm{T}}\left(\boldsymbol{V}\boldsymbol{A}\right)\right]\left[\left(\boldsymbol{V}\boldsymbol{A}\right)^{\mathrm{T}}\boldsymbol{w_i}\right] \\ &= \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_i,\end{aligned} \tag{5.35}$$

where we have used the fact that after the sphering transformation, the mixing matrix $\boldsymbol{V}\boldsymbol{A}$ is orthogonal (cf. Section 5.3.5.1), i. e.

$$\left(\boldsymbol{V}\boldsymbol{A}\right)^{\mathrm{T}} = \left(\boldsymbol{V}\boldsymbol{A}\right)^{-1}. \tag{5.36}$$

Equating Eq. (5.35) with Eq. (5.20), we see that for sphered data, the constraint in Eq. (5.20) can equivalently be expressed in the much more useful form

$$\|\boldsymbol{w}_i\|^2 \stackrel{!}{=} 1. \tag{5.37}$$

In a word, we require that the norm of the vector $\boldsymbol{w}_i$ be unity (Hyvärinen et al., 2001).

5.3.7 Applications of ICA

It is of great interest to inspect for which classes of real-world signals the Independent Component Analysis model is valid in practice.

In this context, the most critical of the assumptions we made in Section 5.3.3 is the one requiring that the mixing is instantaneous. Actually, this requirement is approximately satisfied by electromagnetic signals that are allowed to propagate only over very short distances prior to being recorded as the mixtures $\boldsymbol{x}$. On the other hand, sound propagates so slowly that—depending on the sampling frequency—small differences in the distances between the sources and the recording devices cause considerable time delays, thus making it impossible to apply the instantaneous mixing model. In this case, the sensors generally pick up *convolved* mixtures of the source signals.

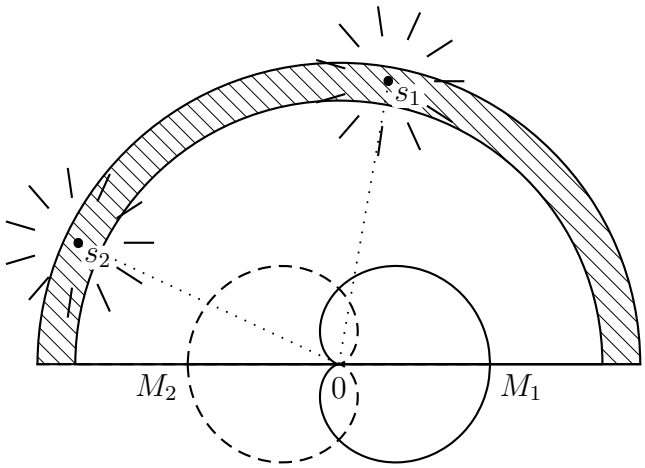


Figure 5.3: Two sources s_1 and s_2 are recorded by two cardioid microphones M_1 and M_2 positioned at 0. Note that the distances between the microphones and the two sources are approximately equal. In the cardioid-shaped curves, the distance from the origin 0 is a measure of the sensitivity of the microphones. (Based on Dickreiter, 1997.)

Nevertheless, under certain circumstances, there is a minor chance of succeeding in performing blind source separation on a two-channel stereo recording. To be precise, consider a high-quality record made using a coincident recording technique, where two artists s_1 and s_2 are geometrically arranged roughly on an arc not far away from the microphones, such that the diffuse sound field does not dominate, e. g. in the hatched area in Fig. 5.3 (Dickreiter, 1997). Then, the assumptions of the Independent Component Analysis model seem to be met quite well and the algorithms for Independent Component Analysis might lead to a satisfactory source separation even for real-world audio signals in a setup like the one described.

According to Hyvärinen et al. (2001), Independent Component Analysis has successfully been used in applications in the field of electroencephalography (EEG) and magnetoencephalography (MEG). Here, the sensors pick up electromagnetic

fields of signals emerging from neural currents in the brain, a process which can be modeled as an instantaneous mixing.

Moreover, in order to extract characteristic feature of signals, Independent Component Analysis can be used to design statistical generative models of observed data. This leads to a data representation that might be applicable in data compression, denoising, or pattern recognition, e. g. in computer vision (Hyvärinen et al., 2001).

Further fields of application include telecommunications (Hyvärinen et al., 2001) and financial market data analysis (Haykin, 2002, Hyvärinen et al., 2001).

5.3.8 Approaches to ICA Model Estimation

There exists a multitude of different approaches to the estimation of the Independent Component Analysis model, some of which will be covered in the next chapters. More specifically, we will develop algorithms based on

- maximization of non-Gaussianity,
- maximum likelihood estimation,
- minimization of mutual information, and
- tensorial methods.

6 Maximization of Non-Gaussianity

In this chapter, we describe a conceptually simple approach to estimating the Independent Component Analysis model parameters based on maximization of non-Gaussianity. First, the approach is intuitively justified by means of the central limit theorem. Then, we treat in detail two measures of non-Gaussianity and develop several algorithms for maximizing them.

Unless stated otherwise, our discussion follows the one in Hyvärinen et al. (2001).

6.1 Justification of Maximization of Non-Gaussianity

To show why it is possible to estimate the Independent Component Analysis model parameters through an optimization problem involving as a cost function a measure of non-Gaussianity, recall the central limit theorem, which we reviewed in Section 1.4. To repeat, the probability density function of a sum of mutually statistically independent random variables tends toward a p. d. f. with a Gaussian distribution. Now, according to the Independent Component Analysis model, the estimates of the independent components y_i can be written as a weighted sum of the (unknown) random variables s_j. In matrix notation, we have

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{B}\boldsymbol{x} && \text{(6.1a)} \\ &= \underbrace{\boldsymbol{B}\boldsymbol{A}}_{\boldsymbol{Q}}\boldsymbol{s} \\ &= \boldsymbol{Q}\boldsymbol{s} && \text{(6.1b)} \end{aligned}$$

with a matrix $\boldsymbol{Q}$ of suitable size. For a single estimate y_i, it follows from Eq. (6.1b) that

$$y_i = \sum_j q_{ij} s_j, \tag{6.1c}$$

where the weights are given by the coefficients q_{ij}. We see from Eq. (6.1c) that since by definition the random variables s_j are mutually statistically independent,

the central limit theorem does indeed apply to the estimates y_i.

Therefore, one can assume that the distribution of the estimate y_i is closer to a Gaussian distribution when the estimate y_i is the sum of *several* independent components s_j. On the other hand, y_i must be least Gaussian if it is the "sum" a of just *single* independent non-Gaussian random variable s_j. Then, in accordance with the ambiguities that remain in the model estimation (Section 5.3.4), only one q_{ij} in Eq. (6.1c) is nonzero and the estimate y_i approximates one independent component s_j (possibly with an inverted sign)

$$y_i = q_{ij} s_j \tag{6.2a}$$

$$= \pm s_j. \tag{6.2b}$$

Of course, we do not have direct access to the independent components as might seem required by Eq. (6.1c). Remember that all we have are the random variables x_i. But, as is evident from Eq. (6.1a), the estimate y_i is also a linear combination of the observable random variables x_i, with the weights given by the coefficients b_{ij} this time

$$y_i = \sum_j b_{ij} x_j \tag{6.3a}$$

$$= \boldsymbol{b}_i^{\mathrm{T}} \boldsymbol{x}. \tag{6.3b}$$

To conclude, a valid estimate of an independent component can be found by tuning the weights of the vector $\boldsymbol{b}_i$ in Eq. (6.3) so that the departure of the resulting distribution of the random variable $y_i = \boldsymbol{b}_i^{\mathrm{T}} \boldsymbol{x}$ from the Gaussian distribution is maximized. In other words, in the approach discussed in this chapter, we try to maximize the non-Gaussianity of the estimate y_i, where—as always in Independent Component Analysis—we fix its variance to unity for reasons explained in Section 5.3.6. For mathematical convenience, let us deal with a sphered input vector $\boldsymbol{z}$ in the rest of this chapter. Note that in this case, the estimate y_i is likewise obtained by

$$y_i = \boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{z} \tag{6.4}$$

under the constraint that the norm of $\boldsymbol{w}_i$ be unity.

In the next two chapters, we show how both the kurtosis of y_i and the negentropy of y_i can be used as cost functions measuring non-Gaussianity. For each of the two measures, we solve the optimization problem by methods explained in Chapter 4

so as to derive algorithms for estimating the Independent Component Analysis model parameters.

Firstly, we limit our discussion to the estimation of only one independent component y_i. At the end of this chapter, we finally show how to obtain the complete unmixing matrix.

6.2 Measuring Non-Gaussianity by Kurtosis

In Section 1.5.2.1, we introduced the kurtosis of a random variable y_i as its forth-order cumulant. For a zero-mean random variable, it is given by

$$\operatorname{kurt}(y_i) = \mathcal{E}\{y_i^4\} - 3\left(\mathcal{E}\{y_i^2\}\right)^2. \tag{6.5a}$$

On the assumption that the variance of the estimate of the independent component is unity, the definition in Eq. (6.5a) can be simplified to

$$\operatorname{kurt}(y_i) = \mathcal{E}\{y_i^4\} - 3. \tag{6.5b}$$

The suitability of kurtosis as a measure of non-Gaussianity was demonstrated in Section 1.6.2, where we found that kurtosis is necessarily zero for a Gaussian-distributed random variable

$$\operatorname{kurt}(y_{\text{Gauss}}) = 0 \tag{6.6}$$

and nonzero for most other random variables. More precisely, kurtosis is positive for super-Gaussian random variables and negative for sub-Gaussian ones, as illustrated in Fig. 1.2. Therefore, we choose the absolute value of the kurtosis[1] of the estimate of the independent component y_i as the cost function in maximization of non-Gaussianity

$$\mathcal{I}_{\text{kurt}}(\boldsymbol{w}_i) = |\operatorname{kurt}(y_i)| \tag{6.7}$$

$$= \left|\operatorname{kurt}\left(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{z}\right)\right|. \tag{6.8}$$

An advantage of measuring non-Gaussianity by the absolute value of kurtosis is obviously its computational simplicity. In particular, it can be estimated easily from a finite number of observations of a random variable as discussed in Section 1.2.2. On the other hand, this kind of sample estimate tends to be quite

[1] Obviously, we can take the square just as well.

sensitive to outliers due to the forth power in its definition in Eq. (6.5a). In a word, it lacks robustness.

Computer Experiment 6.1 (Maximization of Non-Gaussianity)
Fig. 6.1 and Fig. 6.2 summarize the outcomes of a computer experiment on the maximization of non-Gaussianity.

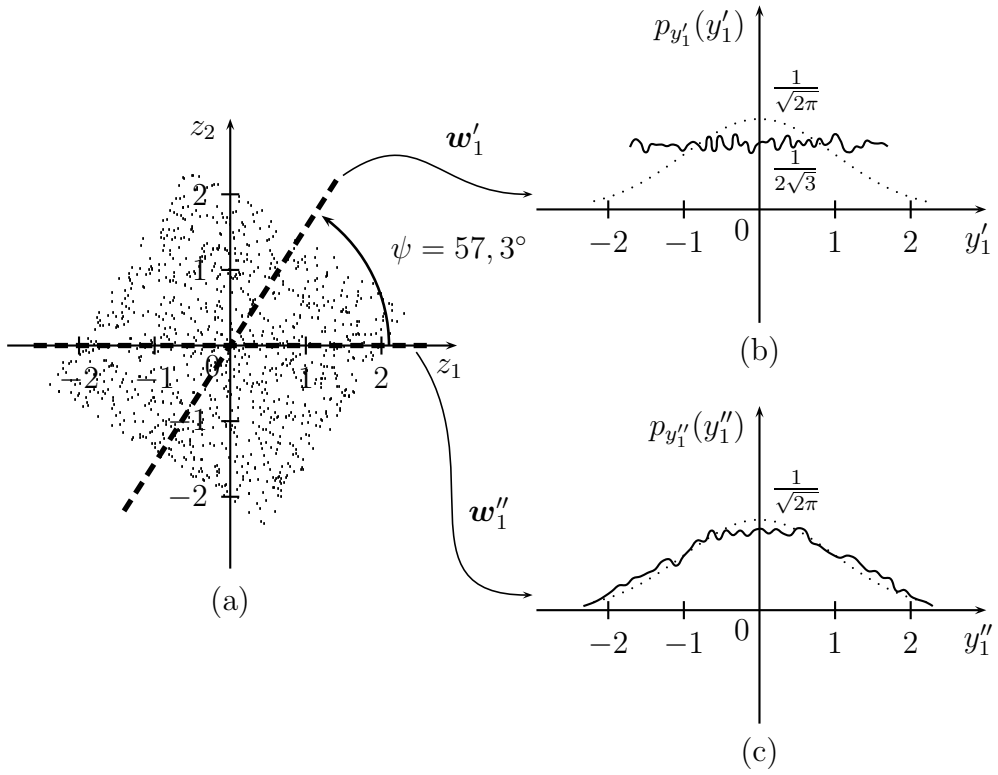


Figure 6.1: Histograms of the estimate of one independent component for two different vectors $\boldsymbol{w}_1'$ and $\boldsymbol{w}_1''$. (a) Samples of a sphered mixture. (b) Histogram of estimate of the independent component $y_1' = \boldsymbol{w}_1'^{\mathrm{T}}\boldsymbol{z}$. (c) Histogram of estimate of the independent component $y_1'' = \boldsymbol{w}_1''^{\mathrm{T}}\boldsymbol{z}$.

In Fig. 6.1(a), you see a number of samples drawn from a sphered mixture $\boldsymbol{z} = \begin{bmatrix} z_1 & z_2 \end{bmatrix}^{\mathrm{T}}$ of two independent components with uniform p. d. f.'s of zero mean and unit variance. As we know, the mixing matrix and with it the unmixing matrix are orthogonal in this case.

Let ψ denote the angle between the vector $\boldsymbol{w}_1$ and the positive x-axis. In this experiment, two specific values of ψ were considered, leading to the two vectors $\boldsymbol{w}_1'$ and $\boldsymbol{w}_1''$, respectively. Note that $\boldsymbol{w}_1'$ corresponds exactly to one of the vectors solving the Independent Component Analysis model. Accordingly, for the estimate of the independent component y_1' resulting from this vector, the histogram in Fig. 6.1(b) reveals the desired uniform structure.

Conversely, this is not true for the second vector $\boldsymbol{w}_1''$. More specifically, from the histogram of the estimate of the independent component y_1'' in Fig. 6.1(c) we can see that the shape of its distribution is almost triangular. Consequently, the similarity to the standardized Gaussian distribution (dotted line) is much more distinct.

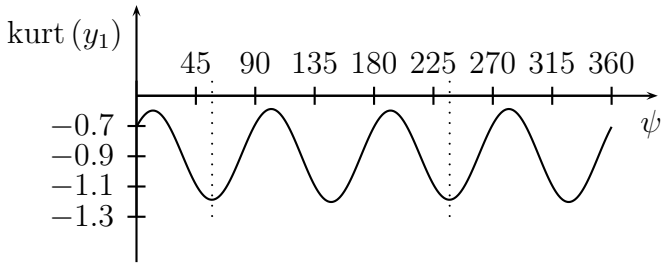


Figure 6.2: Kurtosis of the estimate of one independent component y_1 as a function of the angle ψ.

Similar insights can be gained from the inspection of Fig. 6.2. There, the kurtosis of the estimate of the independent component

$$y_1 = \boldsymbol{w}_1^{\mathrm{T}} \boldsymbol{z} = \begin{bmatrix} \cos\psi & \sin\psi \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \tag{6.9}$$

is plotted as a function of the angle ψ (in degrees). Kurtosis is always negative according to the fact that the independent components are sub-Gaussian. Hence, the maxima of the absolute value of kurtosis are obtained at the minima of kurtosis. For your convenience, the two minima whose values of ψ give the independent component in Eq. (6.9) are marked with dotted lines. Note that there are four local minima as was to be expected due to the sign ambiguity in the Independent Component Analysis model estimation. ■

6.2.1 Gradient Algorithms

We can optimize the cost function $\mathcal{I}_{\mathrm{kurt}}(\boldsymbol{w}_i)$ in Eq. (6.7) making use of the method of steepest descent, which we discussed in Section 4.5. Here, the gradient of the cost function can easily be computed according to the rules mentioned in the appendix of Chapter 4. In particular, we obtain

$$\frac{\partial \mathcal{I}_{\mathrm{kurt}}(\boldsymbol{w}_i)}{\partial \boldsymbol{w}_i} = 4 \operatorname{sign}\left(\operatorname{kurt}\left(\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{z}\right)\right) \left[\mathcal{E}\left\{ \boldsymbol{z} \left(\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{z}\right)^3 \right\} - 3 \boldsymbol{w}_i \underbrace{\|\boldsymbol{w}_i\|^2}_{=1} \right], \tag{6.10}$$

where in practice, the expected values—including the kurtosis of the estimate of the independent component—have to be estimated from the available samples as is typical of the batch algorithms treated in Section 1.2.2. Clearly, prior knowledge of whether the independent component we want to estimate is super-Gaussian or sub-Gaussian can be used right off by choosing the correct sign inside the signum function in Eq. (6.10).

Additionally, we have to take into account the constraint in Eq. (5.37). This can be done most simply by orthogonally projecting the updated vector on the constraint set after every iteration step (Section 4.6.2).

Consequently, an estimate y_i can be found by iteratively performing the following calculations, where the gradient of the cost function is given by Eq. (6.10):

$$\boldsymbol{w}_i'[n] = \boldsymbol{w}_i[n] + \alpha[n] \frac{\partial \mathcal{I}_{\text{kurt}}\left(\boldsymbol{w}_i\right)}{\partial \boldsymbol{w}_i}\bigg|_{\boldsymbol{w}_i=\boldsymbol{w}_i[n]} \tag{6.11a}$$

$$\boldsymbol{w}_i[n+1] = \frac{\boldsymbol{w}_i'[n]}{\|\boldsymbol{w}_i'[n]\|}. \tag{6.11b}$$

Note that here we use the method of steepest descent to maximize the cost function, hence the plus sign in front of the gradient in Eq. (6.11a).

A stochastic gradient algorithm (Section 4.5.2.2) is obtained from Eq. (6.11a) and Eq. (6.10) by omitting the expectation operator inside the parentheses in Eq. (6.10). Then it is possible to use each observation $\boldsymbol{z}[n]$ as soon as it becomes available. On the other hand, unless the kurtosis of the estimate of the independent component in Eq. (6.10) is known a priori, it still has to be estimated correctly. For instance, we can perform an online estimation according to the formula

$$\gamma_i[n+1] = \gamma_i[n] + \alpha_{\gamma_i}[n] \left\{ \left[\left(\boldsymbol{w}_i^{\mathrm{T}}[n] \boldsymbol{z}[n] \right)^4 - 3 \right] - \gamma_i[n] \right\}, \tag{6.12}$$

where $\gamma_i[n]$ denotes the estimate of the kurtosis of y_i in iteration step n and $\alpha_{\gamma_i}[n]$ is the sequence of the step-size parameter.

6.2.2 Fixed-Point Algorithm

From Example 4.4 we know that a vector $\boldsymbol{w}_i$ solving the optimization problem of maximizing a cost function $\mathcal{I}\left(\boldsymbol{w}_i\right)$ under the constraint that the norm of $\boldsymbol{w}_i$ be unity always points in the same direction as the gradient of the cost function.

Adapted to the problem considered here, Eq. (4.37) reads

$$\frac{\partial \mathcal{I}_{\text{kurt}}(\boldsymbol{w}_i)}{\partial \boldsymbol{w}_i} + 2\lambda \boldsymbol{w}_i = \boldsymbol{0}, \tag{6.13}$$

where the gradient is given by Eq. (6.10). To be able to solve Eq. (6.13) by a fixed-point iteration (Section 4.3), we rewrite it in the form

$$\frac{\partial \mathcal{I}_{\text{kurt}}(\boldsymbol{w}_i)}{\partial \boldsymbol{w}_i} = -2\lambda \boldsymbol{w}_i. \tag{6.14}$$

Due to the mandatory orthogonal projection of the solution $\boldsymbol{w}_i$ on the constraint set, the expression -2λ on the right-hand side of Eq. (6.14) has no effect whatsoever and can therefore be omitted. This finally leads to the so-called *FastICA-algorithm* for maximization of non-Gaussianity measured by the absolute value of kurtosis

$$\boldsymbol{w}_i'[n] = \mathcal{E}\left\{\boldsymbol{z}\left(\boldsymbol{w}_i^{\mathrm{T}}[n]\boldsymbol{z}\right)^3\right\} - 3\boldsymbol{w}_i[n] \tag{6.15}$$

$$\boldsymbol{w}_i[n+1] = \frac{\boldsymbol{w}_i'[n]}{\|\boldsymbol{w}_i'[n]\|}. \tag{6.16}$$

6.2.3 Comparison between Gradient-Based Algorithms and the Fixed-Point Algorithm

One remarkable virtue of the FastICA algorithm over the methods of steepest descent is clear from Eq. (6.15). More precisely, note that the FastICA algorithm does not have a step-size parameter as the gradient-based methods do. This results in increased reliability and ease of use. In addition, one can show that the FastICA algorithm generally converges of second order, which means very fast convergence (cf. Section 4.2.3). For independent components with a symmetric distribution, convergence is even cubic.

Conversely, the convergence speed of gradient-based methods is only linear. Furthermore, convergence is significantly influenced by the choice of a suitable learning-rate sequence. On the other hand, especially the stochastic gradient algorithms allow fast adaptation in nonstationary environments.

6.3 Measuring Non-Gaussianity by Negentropy

As a second possibility of measuring non-Gaussianity, let us consider negentropy, which we treated in detail in Section 3.3. We saw there that the negentropy of

a random variable y_i is zero if and only if the random variable has a Gaussian distribution

$$\mathcal{N}(y_i) = 0 \iff y_i = y_{\text{Gauss}}, \tag{6.17}$$

while in all other cases, the negentropy is positive due to the maximum entropy property of Gaussian-distributed random variables (cf. Section 3.1.4). (In contrast, kurtosis is not necessarily nonzero for random variables with a non-Gaussian distribution.)

However, we need complete knowledge of the p. d. f. of the random variable under consideration in order to be able to compute its negentropy exactly. In this sense, negentropy is computationally much more demanding than kurtosis. Fortunately, there exist several approximations of negentropy that are helpful in Independent Component Analysis by maximization of non-Gaussianity. In particular, two different kinds of negentropy approximation were mentioned in Section 3.3.2:

1. approximation of negentropy by higher-order cumulants
2. approximation of negentropy by nonpolynomial functions

We have already shown in Section 3.3.2.1 that for symmetric distributions, which are often encountered in practice, the first kind of approximation actually leads to an approximation of negentropy proportional to the square of kurtosis. Since maximization of the square of an expression leads to the same extrema as maximization of its absolute value, from this measure of non-Gaussianity we get conceptually the algorithms from Section 6.2.

However, a much more appealing measure of non-Gaussianity can be obtained from the approximation of negentropy by nonpolynomial functions. In the simplest case, i. e. if we decide to take just one nonlinear function $G(\cdot)$, the negentropy $\mathcal{N}(y_i)$ of a random variable y_i can be approximated by (cf. Section 3.3.2.2)

$$\mathcal{N}(y_i) = k_1\big(\mathcal{E}\{G(y_i)\} - \mathcal{E}\{G(\nu)\}\big)^2, \tag{6.18}$$

where ν is a standardized Gaussian-distributed random variable. Since the positive constant k_1 is just a scaling factor, it can be omitted in optimization problems. The approximation of negentropy in Eq. (6.18) is designed so that it is necessarily zero for a Gaussian-distributed random variable and positive otherwise. Accordingly, we can use it as a cost function for maximization of non-Gaussianity

$$\mathcal{I}_{\mathcal{N}}(\boldsymbol{w}_i) = \big(\mathcal{E}\big\{G\left(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{z}\right)\big\} - \mathcal{E}\{G(\nu)\}\big)^2. \tag{6.19}$$

In order to obtain measures of non-Gaussianity that are more robust than kurtosis, we can use the nonlinear functions proposed in Section 3.3.2.2, reproduced here for convenience

$$G_1(y_i) = \frac{1}{a_1} \ln \cosh(a_1 y_i), \qquad 1 \leq a_1 \leq 2, \tag{6.20a}$$

$$G_2(y_i) = -\mathrm{e}^{-\frac{y_i^2}{2}} \tag{6.20b}$$

$$G_3(y_i) = \frac{1}{4} y_i^4. \tag{6.20c}$$

As mentioned in Hyvärinen (1999), $G_1(\cdot)$ is a good general-purpose nonlinear function for the optimization problem considered in this section. On the other hand, $G_2(\cdot)$ might be better for highly super-Gaussian independent components, or when robustness is very important.

Interestingly enough, when we use the function in Eq. (6.20c), the resulting cost function $\mathcal{I}_{\mathcal{N}}(\boldsymbol{w}_i)$ in Eq. (6.19) resembles the approximation of negentropy by cumulants.

Measuring non-Gaussianity by the cost function in Eq. (6.19) using nonlinear functions like the first and the second in Eq. (6.20) provides estimates of independent components that are considerably more robust than those obtained by the maximization of kurtosis without being computationally more demanding.

6.3.1 Gradient Algorithms

The gradient of the cost function $\mathcal{I}_{\mathcal{N}}(\boldsymbol{w}_i)$ in Eq. (6.19) with respect to the weight vector $\boldsymbol{w}_i$ is given by

$$\frac{\partial \mathcal{I}_{\mathcal{N}}(\boldsymbol{w}_i)}{\partial \boldsymbol{w}_i} = \rho \, \mathcal{E}\left\{\boldsymbol{z} g\left(\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{z}\right)\right\}, \qquad \rho = \mathcal{E}\left\{G\left(\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{z}\right)\right\} - \mathcal{E}\{G(\nu)\}, \tag{6.21}$$

where the function $g(\cdot)$ is the derivative of the nonpolynomial function $G(\cdot)$ used in the approximation of negentropy in Eq. (6.19)

$$g(y_i) = \frac{\mathrm{d}G(y_i)}{\mathrm{d}y_i}. \tag{6.22}$$

In Fig. 6.3, the derivatives $g_i(y_i)$ corresponding to the functions in Eq. (6.20) are

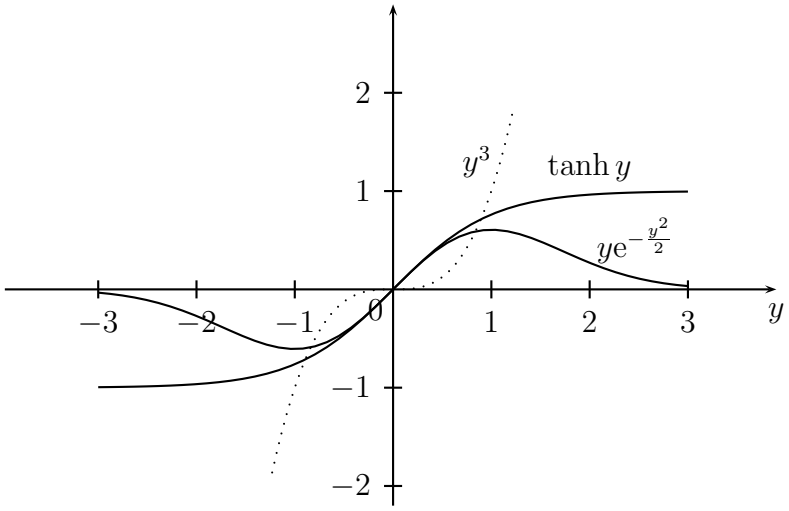


Figure 6.3: Derivatives of two functions suitable for approximating negentropy by one nonlinear function. For reference, the derivative of the forth power of x corresponding to the kurtosis measure is also plotted.

plotted. They read

$$g_1(y_i) = \tanh(a_1 y_i), \qquad 1 \leq a_1 \leq 2 \tag{6.23a}$$

$$g_2(y_i) = y_i \mathrm{e}^{-\frac{y_i^2}{2}} \tag{6.23b}$$

$$g_3(y_i) = y_i^3. \tag{6.23c}$$

Actually, as far as the stability of the resulting algorithm is concerned, the choice of the function $G(\cdot)$ is not critical as long as the input data are sphered and the solution vectors $\boldsymbol{w}_i$ are constrained to unit norm. Thus, the method based on the cost function in Eq. (6.19) is rather general and allows one to tailor algorithms for Independent Component Analysis maximizing non-Gaussianity to the special requirements of the task at hand. More information on this issue can be found in Hyvärinen and Oja (1998).

Note that the term $\mathcal{E}\{G(\nu)\}$ involved in the computation of ρ in Eq. (6.21) can be computed exactly from the definition of the mathematical expectation in Eq. (1.10), whereas the other expectations have to be estimated from the available data samples.

Taking into account the constraint in Eq. (5.37), we obtain an estimate y_i by

iterating

$$\boldsymbol{w}_i'[n] = \boldsymbol{w}_i[n] + \alpha[n] \frac{\partial \mathcal{I}_{\mathcal{N}}(\boldsymbol{w}_i)}{\partial \boldsymbol{w}_i}\bigg|_{\boldsymbol{w}_i = \boldsymbol{w}_i[n]} \tag{6.24a}$$

$$\boldsymbol{w}_i[n+1] = \frac{\boldsymbol{w}_i'[n]}{\|\boldsymbol{w}_i'[n]\|}, \tag{6.24b}$$

where the gradient is given by Eq. (6.21).

Again, we can replace the expectation $\mathcal{E}\{\boldsymbol{z} g(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{z})\}$ in Eq. (6.21) by its current value $\boldsymbol{z}[n] g(\boldsymbol{w}_i^{\mathrm{T}}[n]\boldsymbol{z}[n])$ in order to obtain a stochastic gradient algorithm. On the other hand, the parameter ρ still has to be estimated online, e. g. along the lines of the estimate of kurtosis in Eq. (6.12).

6.3.2 Fixed-Point Algorithm

An algorithm having as favorable properties as the fixed-point iteration in Section 6.2.2 can be designed for negentropy as a measure of non-Gaussianity as well. Details on the derivation of the algorithm can be found in the reference mentioned at the beginning of this chapter. In principle, the constrained optimization problem of maximizing the non-Gaussianity measured by an approximation of negentropy like the one in Eq. (6.18) is solved by the method of Lagrange multipliers (cf. Section 4.6.1). The corresponding adjoint equation is in turn solved by Newton's method (cf. Section 4.4), where the particular structure of the sphered input data is exploited in order to avoid the compulsory matrix inversion of the Jacobian of the adjoint equation. The resulting FastICA algorithm has the form of a fixed-point iteration (Section 4.3) and has to be followed by projection on the constraint set

$$\boldsymbol{w}_i'[n] = \mathcal{E}\{\boldsymbol{z} g(\boldsymbol{w}_i^{\mathrm{T}}[n]\boldsymbol{z})\} - \mathcal{E}\{g'(\boldsymbol{w}_i^{\mathrm{T}}[n]\boldsymbol{z})\}\boldsymbol{w}_i[n] \tag{6.25}$$

$$\boldsymbol{w}_i[n+1] = \frac{\boldsymbol{w}_i'[n]}{\|\boldsymbol{w}_i'[n]\|}. \tag{6.26}$$

Here, $g'(\cdot)$ denotes the derivative of the learning function $g(\cdot)$. For the functions in Eq. (6.23a), we have

$$g_1'(y) = a_1\left(1 - \tanh^2(a_1 y)\right), \qquad 1 \le a_1 \le 2, \tag{6.27a}$$

$$g_2'(y) = \left(1 - y^2\right)\mathrm{e}^{-\frac{y^2}{2}} \tag{6.27b}$$

$$g_3'(y) = 3y^2. \tag{6.27c}$$

6.3.3 Comparison between Gradient-Based Algorithms and the Fixed-Point Algorithm

As for the advantages and drawbacks of the algorithms based on the steepest-descent method and the FastICA algorithm measuring non-Gaussianity by an approximation of negentropy, the same can be said as in Section 6.2.3. In addition, note that the FastICA algorithm based on the approximation of negentropy is even better than the FastICA algorithm maximizing kurtosis because of the desirable statistical properties of negentropy or its approximation.

6.4 Estimating the Complete Unmixing Matrix

So far, we treated the estimation of only a single independent component y_i, on the assumption that the input data have been sphered in a preprocessing step. The complete set of estimates of the independent components can simply be obtained as described in the following.

First, recall that the independent components y_i are by definition mutually statistically independent. As pointed out in Section 1.3.2, this implies their uncorrelatedness

$$\mathcal{E}\{y_i y_j\} = \mathcal{E}\{y_i\}\,\mathcal{E}\{y_j\} \tag{6.28}$$

for every two different indices $i \neq j$. Moreover, for zero-mean random variables, Eq. (6.28) reduces to

$$\mathcal{E}\{y_i y_j\} = 0. \tag{6.29}$$

Considering that the input data $\boldsymbol{z}$ are sphered, i. e.

$$\mathcal{E}\{\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}}\} = \boldsymbol{I}, \tag{6.30}$$

where $\boldsymbol{I}$ denotes the identity matrix, we get from Eq. (6.29)

$$\begin{aligned}
\mathcal{E}\{y_i y_j\} &= \mathcal{E}\{(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{z})\,(\boldsymbol{w}_j^{\mathrm{T}}\boldsymbol{z})\} \\
&= \mathcal{E}\{(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{z})\,(\boldsymbol{z}^{\mathrm{T}}\boldsymbol{w}_j)\} \\
&= \boldsymbol{w}_i^{\mathrm{T}} \underbrace{\mathcal{E}\{\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}}\}}_{=\boldsymbol{I}} \boldsymbol{w}_j \\
&= \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{w}_j = 0.
\end{aligned} \tag{6.31}$$

In a word, the vectors $\boldsymbol{w}_i$ have to be mutually orthogonal. Consequently, we run an algorithm of our choice to estimate the independent components as we have done in this chapter and prevent the vectors $\boldsymbol{w}_i$ from converging onto the same extrema by means of orthogonalization. More precisely, two different orthogonalization schemes are usually used in this context: (1) the well-known Gram-Schmidt orthogonalization and (2) a symmetric orthogonalization. Furthermore, Hyvärinen et al. (2001) present algorithms for performing online orthogonalization.

6.4.1 Gram-Schmidt Orthogonalization

The Gram-Schmidt orthogonalization allows to compute a vector $\tilde{\boldsymbol{w}}_p$ that is orthogonal to a set of mutually orthogonal vectors $\tilde{\boldsymbol{w}}_j, j = 1, \ldots, p-1$ and an arbitrary vector $\boldsymbol{w}_p$, where the norm of each is unity

$$\tilde{\boldsymbol{w}}_p = \boldsymbol{w}_p - \sum_{j=1}^{p-1} \left(\boldsymbol{w}_p^{\mathrm{T}} \tilde{\boldsymbol{w}}_j\right) \tilde{\boldsymbol{w}}_j, \qquad p = 2, \ldots, N. \tag{6.32}$$

In our optimization problem, we successively compute vectors $\boldsymbol{w}_p$ that are then orthogonalized with respect to the vectors previously found by Eq. (6.32) as a last step in the update procedure. This will of course be followed by normalizing the new vector $\tilde{\boldsymbol{w}}_p$ by its norm in order to satisfy the constraint in Eq. (5.37).

Let us compile the set of orthogonal vectors $\tilde{\boldsymbol{w}}_i, i = 1, \ldots, N$ in a matrix

$$\tilde{\boldsymbol{W}} = \begin{bmatrix} \tilde{\boldsymbol{w}}_1^{\mathrm{T}} \\ \vdots \\ \tilde{\boldsymbol{w}}_N^{\mathrm{T}} \end{bmatrix}. \tag{6.33}$$

Then, the complete set of independent component estimates $\boldsymbol{y} = \begin{bmatrix} y_1 & \cdots & y_N \end{bmatrix}^{\mathrm{T}}$ can be obtained by

$$\boldsymbol{y} = \tilde{\boldsymbol{W}} \boldsymbol{z}. \tag{6.34}$$

From a numerical point of view, we have to note that estimation errors made during the computations of the first vectors accumulate in successive vectors. In addition, it is not possible to compute the estimates of several independent components in parallel. The symmetric orthogonalization discussed next does not have these drawbacks.

6.4.2 Symmetric Orthogonalization by Eigenvalue Decomposition

As an alternative to the Gram-Schmidt orthogonalization, we can first estimate the vectors $\boldsymbol{w}_i$ separately by a suitable algorithm. From the matrix

$$\boldsymbol{W} = \begin{bmatrix} \boldsymbol{w}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{w}_N^{\mathrm{T}} \end{bmatrix}, \tag{6.35}$$

an orthogonal matrix $\tilde{\boldsymbol{W}}$ can be obtained by the formula

$$\tilde{\boldsymbol{W}} = \left(\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\right)^{-1/2} \boldsymbol{W}. \tag{6.36}$$

Here, the so-called inverse square root $\left(\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\right)^{-1/2}$ is computed from the eigenvalue decomposition of the matrix $\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}$

$$\begin{aligned} \boldsymbol{W}\boldsymbol{W}^{\mathrm{T}} &= \boldsymbol{E}\,\mathrm{diag}\left(\lambda_1, \ldots, \lambda_N\right)\boldsymbol{E}^{\mathrm{T}} && \text{(6.37a)} \\ &= \boldsymbol{E}\boldsymbol{D}\boldsymbol{E}^{\mathrm{T}} && \text{(6.37b)} \end{aligned}$$

according to

$$\begin{aligned} \left(\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\right)^{-1/2} &= \boldsymbol{E}\,\mathrm{diag}\left(\lambda_1^{-1/2}, \ldots, \lambda_N^{-1/2}\right)\boldsymbol{E}^{\mathrm{T}} && \text{(6.38a)} \\ &= \boldsymbol{E}\boldsymbol{D}^{-1/2}\boldsymbol{E}^{\mathrm{T}}. && \text{(6.38b)} \end{aligned}$$

Using Eq. (6.37), Eq. (6.38), and the fact that the matrices $\boldsymbol{E}$ are orthogonal, we can easily show that the matrix $\tilde{\boldsymbol{W}}$ is indeed orthogonal. In particular, we have

$$\begin{aligned} \tilde{\boldsymbol{W}}\tilde{\boldsymbol{W}}^{\mathrm{T}} &= \left[\left(\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\right)^{-1/2}\boldsymbol{W}\right]\left[\left(\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\right)^{-1/2}\boldsymbol{W}\right]^{\mathrm{T}} \\ &= \left[\boldsymbol{E}\boldsymbol{D}^{-1/2}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{W}\right]\left[\boldsymbol{E}\boldsymbol{D}^{-1/2}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{W}\right]^{\mathrm{T}} \\ &= \boldsymbol{E}\boldsymbol{D}^{-1/2}\boldsymbol{E}^{\mathrm{T}}\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}\boldsymbol{E}\boldsymbol{D}^{-1/2}\boldsymbol{E}^{\mathrm{T}} \\ &= \boldsymbol{E}\boldsymbol{D}^{-1/2}\underbrace{\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}}_{=\boldsymbol{I}}\boldsymbol{D}\underbrace{\boldsymbol{E}^{\mathrm{T}}\boldsymbol{E}}_{=\boldsymbol{I}}\boldsymbol{D}^{-1/2}\boldsymbol{E}^{\mathrm{T}} \\ &= \boldsymbol{E}\underbrace{\boldsymbol{D}^{-1/2}\boldsymbol{D}\boldsymbol{D}^{-1/2}}_{=\boldsymbol{I}}\boldsymbol{E}^{\mathrm{T}} \\ &= \boldsymbol{E}\boldsymbol{E}^{\mathrm{T}} \\ &= \boldsymbol{I}. \end{aligned}$$

Here again, the independent components are obtained like in Eq. (6.34).

6.5 Summary and Outlook

In this chapter, we showed that the Independent Component Analysis model can be solved by maximizing the non-Gaussianity of the estimates of the independent components. For measuring non-Gaussianity, a simple approximation of negentropy is to be preferred to the classical measure of kurtosis from a statistical point of view. Moreover, we developed several algorithms based on the method of steepest descent and on the fixed-point iteration.

Remarkably, the algorithms introduced in this chapter allow to estimate only a part of the independent components as a special feature, which is not possible for most other algorithms (Hyvärinen and Oja, 1997).

Semi-adaptive versions, e. g. of the FastICA algorithms, can be obtained by estimating the expectations not over the whole data set, but over blocks of samples only. This way, even the batch algorithms allow an adaptation in a nonstationary environment (Hyvärinen and Oja, 1997).

7 Maximum Likelihood Estimation

A popular and efficient method for solving the Independent Component Analysis model is based on the maximum likelihood method, which we discussed in Section 2.2. Roughly speaking, the concept of the maximum likelihood method is to find those parameter values that are most likely responsible for having generated some observations. Here, the parameters are the elements of the unmixing matrix.

Among all unbiased point estimates, the one obtained from the maximum likelihood method is the most efficient.

7.1 Log-Likelihood Function of the ICA Model

First, let us derive the likelihood function $\ell(\cdot)$ of the Independent Component Analysis model as in Hyvärinen et al. (2001), where we have made some minor corrections, though. To this end, we assume that the Independent Component Analysis model from Eq. (5.7)

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s} \tag{7.1}$$

possesses the unique solution

$$\begin{aligned} \boldsymbol{s} &= \boldsymbol{A}^{-1}\boldsymbol{x} \\ &= \boldsymbol{B}\boldsymbol{x} = \begin{bmatrix} \boldsymbol{b}_1^{\mathrm{T}} \\ \vdots \\ \boldsymbol{b}_N^{\mathrm{T}} \end{bmatrix} \boldsymbol{x}, \end{aligned} \tag{7.2}$$

where $\boldsymbol{B} = \boldsymbol{A}^{-1}$. Then, proceeding along the lines of Example 1.2, we get from Eq. (1.9) for the p. d. f. $p_{\boldsymbol{x}}(\boldsymbol{x})$ of the observable random variables $\boldsymbol{x}$

$$\begin{aligned} p_{\boldsymbol{x}}(\boldsymbol{x}) &= \frac{1}{|\det \boldsymbol{A}|} p_{\boldsymbol{s}}(\boldsymbol{s}) \\ &= |\det \boldsymbol{B}|\, p_{\boldsymbol{s}}(\boldsymbol{s}) \\ &= |\det \boldsymbol{B}| \prod_{i=1}^{N} p_{s_i}(s_i) \\ &= |\det \boldsymbol{B}| \prod_{i=1}^{N} p_{s_i}\left(\boldsymbol{b}_i^{\mathrm{T}} \boldsymbol{x}\right). \end{aligned} \tag{7.3}$$

Here, $p_{s_i}(s_i)$ denotes the marginal p. d. f. of the independent component s_i. Note that in the last steps, we considered the statistical independence of the independent components s_i and used the identity (Bartsch, 1999)

$$\frac{1}{\det \boldsymbol{A}} = \det \boldsymbol{A}^{-1} = \det \boldsymbol{B}. \tag{7.4}$$

Finally, the likelihood function $\ell(\boldsymbol{B})$ for K observations $\boldsymbol{x}[k]$ of the random vector $\boldsymbol{x}$ is given by

$$\ell(\boldsymbol{B}) = \prod_{k=1}^{K} |\det \boldsymbol{B}| \prod_{i=1}^{N} p_{s_i}\left(\boldsymbol{b}_i^{\mathrm{T}} \boldsymbol{x}[k]\right) \tag{7.5}$$

$$= |\det \boldsymbol{B}|^K \prod_{k=1}^{K} \prod_{i=1}^{N} p_{s_i}\left(\boldsymbol{b}_i^{\mathrm{T}} \boldsymbol{x}[k]\right) \tag{7.6}$$

and the corresponding log-likelihood function $\mathcal{L}(\boldsymbol{B})$ by

$$\mathcal{L}(\boldsymbol{B}) = \ln \ell(\boldsymbol{B}) \tag{7.7}$$

$$= K \ln |\det \boldsymbol{B}| + \sum_{k=1}^{K} \sum_{i=1}^{N} \ln p_{s_i}\left(\boldsymbol{b}_i^{\mathrm{T}} \boldsymbol{x}[k]\right). \tag{7.8}$$

As suggested in Hyvärinen et al. (2001), we can take the sum over the K observations in Eq. (7.8) for a corresponding expected value—similar to Eq. (1.14)—in order to obtain the cost function for maximum likelihood estimation

$$\mathcal{I}_{\mathrm{ML}}(\boldsymbol{B}) = \frac{1}{K} \mathcal{L}(\boldsymbol{x}) \tag{7.9}$$

$$= \ln |\det \boldsymbol{B}| + \mathcal{E}\left\{ \sum_{i=1}^{N} \ln p_{s_i}\left(\boldsymbol{b}_i^{\mathrm{T}} \boldsymbol{x}\right) \right\}. \tag{7.10}$$

7.1.1 Nonparametric Density Estimation

It should be noted that the cost function $\mathcal{I}_{\mathrm{ML}}$ in Eq. (7.10) depends on several quantities:

1. the elements of the unmixing matrix $\boldsymbol{B}$
2. the unknown p. d. f.'s of the independent components

As for the unknown p. d. f.'s, one would usually prefer a nonparametric estimation. Now, such a nonparametric estimation of the p. d. f.'s of the independent components would be statistically difficult and require a huge amount of input data (Hyvärinen et al., 2001). Therefore, we prefer to avoid this kind of estimation and try designing suitable density approximations that can be parameterized by as few parameters as possible.

7.1.2 Binary Density Approximation for the ICA Cost Function

Fortunately, in order to be able to solve the Independent Component Analysis model, an approximation by just two different probability density functions is sufficient, as shown in Hyvärinen et al. (2001). More specifically, let $\tilde{p}_{s_j}(s_j)$ denote *assumed* p. d. f.'s of the independent components s_j. In addition, constrain the estimates $y_i = \boldsymbol{b}_i^{\mathrm{T}}\boldsymbol{x}$ of the independent components to be sphered (Section 5.3.5), such that

$$\mathcal{E}\{\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\} = \boldsymbol{I}. \tag{7.11}$$

Then, the estimate obtained from optimization of the likelihood function is locally consistent, i. e. it locally converges onto an independent component for a large number of observations K, if the nonpolynomial moment

$$\mathcal{E}\Big\{s_j g_{s_j}(s_j) - g'_{s_j}(s_j)\Big\} > 0, \tag{7.12}$$

is positive for all independent components s_j. Here, g_{s_j} is computed from the assumed p. d. f. $\tilde{p}_{s_j}(s_j)$ of the independent component s_j according to

$$g_{s_j}(s_j) = \frac{\mathrm{d}}{\mathrm{d}s_j} \ln \tilde{p}_{s_j}(s_j) = \frac{1}{\tilde{p}_{s_j}(s_j)} \frac{\mathrm{d}\tilde{p}_{s_j}(s_j)}{\mathrm{d}s_j}, \tag{7.13}$$

and $g'_{s_j}(s_j)$ is the derivative of $g_{s_j}(s_j)$ with respect to s_j.

Hyvärinen et al. (2001) propose a set of two p. d. f.'s for one of which the nonpolynomial moment in Eq. (7.12) is always positive, so that the maximum-likelihood

estimate converges to one of the true independent components. More precisely, consider the two log-densities

$$\ln \tilde{p}^{+}_{s_j}(s_j) = \varpi_1 - 2 \ln \cosh s_j \tag{7.14a}$$

and

$$\ln \tilde{p}^{-}_{s_j}(s_j) = \varpi_2 - \left(\frac{s_j^2}{2} - \ln \cosh s_j \right), \tag{7.14b}$$

where ϖ_1 and ϖ_2 are scaling factors that turn the functions into proper p. d. f.'s. $\tilde{p}^{+}_{s_j}(s_j)$ is the p. d. f. of a super-Gaussian random variable, whereas $\tilde{p}^{-}_{s_j}(s_j)$ is the p. d. f. of a sub-Gaussian random variable. For the densities in Eq. (7.14a) and Eq. (7.14b), the nonpolynomial moment in Eq. (7.12) is given by

$$2\mathcal{E}\left\{ -s_j \tanh s_j + \left(1 - \tanh^2 s_j\right) \right\} \tag{7.15a}$$

and by

$$\mathcal{E}\left\{ s_j \tanh s_j - \left(1 - \tanh^2 s_j\right) \right\}, \tag{7.15b}$$

respectively. Obviously, the signs of the nonpolynomial moments in Eq. (7.15) are always opposite. As a consequence, only one of the p. d. f.'s used in Eq. (7.14a) and Eq. (7.14b) gives the correct sign as required in Eq. (7.12) and can therefore be used in the maximum likelihood estimation of the Independent Component Analysis model instead of the true but unknown p. d. f. of the independent component.

In practice, to find out which of the two densities to take, we perform an online computation of the nonpolynomial moments in Eq. (7.15) along the lines of the online estimation of kurtosis in Eq. (6.12), using an estimate of the independent component y_i. We then choose the p. d. f. for which the corresponding nonpolynomial moment is positive as required by Eq. (7.12).

Once a suitable p. d. f. is found, the problems associated with the nonparametric density estimation are solved. Then, the cost function $\mathcal{I}_{\mathrm{ML}}(\boldsymbol{B})$ in Eq. (7.10) is a function of the unmixing matrix $\boldsymbol{B}$ only.

Note finally that if knowledge of the p. d. f. of the independent components to estimate is available, it can of course be used in the log-likelihood function. From Eq. (7.12) we conclude that the p. d. f. need not be known exactly. More specifically, small misspecifications can be tolerated as long as the nonpolynomial moment in Eq. (7.12) has the correct positive sign (Hyvärinen et al., 2001).

7.2 The Bell-Sejnowski Algorithm

In this section, we solve the problem of optimizing the log-likelihood-based cost function by the method of steepest descent. Using Eq. (4.47), we may express the gradient of the cost function $\mathcal{I}_{\mathrm{ML}}(\boldsymbol{B})$ in Eq. (7.10) as

$$\frac{\partial \mathcal{I}_{\mathrm{ML}}(\boldsymbol{B})}{\partial \boldsymbol{B}} = \left(\boldsymbol{B}^{\mathrm{T}}\right)^{-1} + \mathcal{E}\left\{\boldsymbol{g}(\boldsymbol{B}\boldsymbol{x})\,\boldsymbol{x}^{\mathrm{T}}\right\}, \tag{7.16}$$

where

$$\boldsymbol{g}(\boldsymbol{B}\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{y}) \tag{7.17}$$

$$= \begin{bmatrix} g_{s_1}(s_1) \\ \vdots \\ g_{s_N}(s_N) \end{bmatrix} \tag{7.18}$$

and the functions $g_{s_j}(s_j)$ are determined from the estimates y_i as discussed in Section 7.1.2. Thus, if the nonpolynomial moment in Eq. (7.15a) is positive for an independent component s_j, the nonlinear function $g^{+}_{s_j}(s_j)$ to be employed in Eq. (7.17) is

$$g^{+}_{s_j}(s_j) = -2\tanh s_j, \tag{7.19a}$$

whereas if it is negative, the nonlinear function

$$g^{-}_{s_j}(s_j) = \tanh s_j - s_j \tag{7.19b}$$

should be used instead.

(7.19c)

To summarize, we get as an iterative algorithm for maximization of the log-likelihood function of the observations (Hyvärinen et al., 2001)

$$\boldsymbol{B}[n+1] = \boldsymbol{B}[n] + \alpha[n] \frac{\partial \mathcal{I}_{\mathrm{ML}}(\boldsymbol{B})}{\partial \boldsymbol{B}}\bigg|_{\boldsymbol{B}=\boldsymbol{B}[n]}, \tag{7.20}$$

where the gradient is given by Eq. (7.16) and $\alpha[n]$ denotes a suitable step-size parameter sequence. Note that according to Eq. (7.11), the estimates of the independent components should be sphered before estimating the nonpolynomial

moment for the binary density approximation to work correctly for Independent Component Analysis.

Once more, an online stochastic gradient algorithm commonly known as the *Bell-Sejnowski algorithm* can be obtained from the update rule in Eq. (7.20) by substituting $\boldsymbol{g}\left(\boldsymbol{B}[n]\boldsymbol{x}[n]\right)\boldsymbol{x}^{\mathrm{T}}[n]$ for the expectation $\mathcal{E}\left\{\boldsymbol{g}\left(\boldsymbol{B}\boldsymbol{x}\right)\boldsymbol{x}^{\mathrm{T}}\right\}$, thus making it possible to use every observation $\boldsymbol{x}[n]$ as soon as it becomes available (Hyvärinen et al., 2001). This yields

$$\boldsymbol{B}[n+1] = \boldsymbol{B}[n] + \alpha[n]\left[\left(\boldsymbol{B}^{\mathrm{T}}[n]\right)^{-1} + \boldsymbol{g}\left(\boldsymbol{B}[n]\boldsymbol{x}[n]\right)\boldsymbol{x}^{\mathrm{T}}[n]\right]. \tag{7.21}$$

7.2.1 Derivation from Infomax Principle

Actually, the Bell-Sejnowski algorithm in Eq. (7.21) was first derived not from the maximum likelihood approach, but using the principle of *information maximization* (Bell and Sejnowski, 1995).

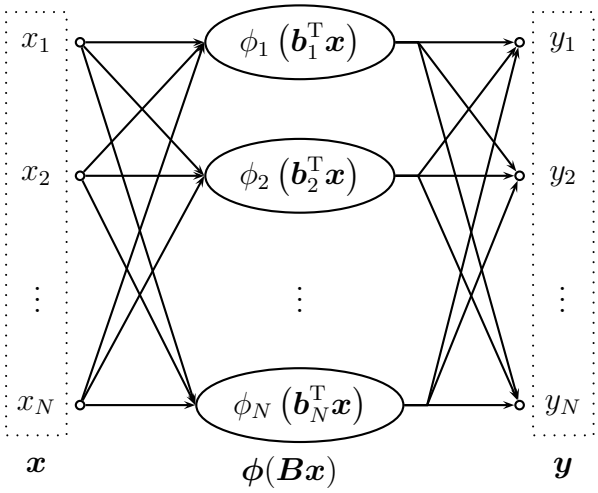


Figure 7.1: Neural network structure, $N \to N$ mapping.

As an illustration of this principle, consider the network of nonlinear units depicted in Fig. 7.1, where each unit, which is represented by an oval node, computes its output according to a nonlinear *sigmoid function* $\phi_i(\cdot)$ that depends on the input vector $\boldsymbol{x}$ and a vector of adjustable weights $\boldsymbol{b}_i$. Note that in general, the network will also contain some noise sources.

Now, the principle of information maximization aims at maximizing the information transmission from the input $\boldsymbol{x}$ to the output $\boldsymbol{y}$ of the network by adjusting the weights $\boldsymbol{B}$ so that the mutual information $I(\boldsymbol{x}, \boldsymbol{y})$ between the inputs and the outputs (cf. Section 3.2) is maximized. It can be shown that for the zero-noise

limit, maximization of the mutual information $I(\boldsymbol{x}, \boldsymbol{y})$ between the inputs and the outputs is equivalent to maximization of the differential entropy $h(\boldsymbol{y})$ of the outputs alone.

From Eq. (3.6), the differential entropy $h(\boldsymbol{y})$ of the outputs can be computed as

$$h(\boldsymbol{y}) = h(\boldsymbol{x}) + \mathcal{E}\{\log|\det \boldsymbol{J}_{\boldsymbol{\phi}}(\boldsymbol{x})|\}, \tag{7.22}$$

where $\boldsymbol{J}_{\boldsymbol{\phi}}(\boldsymbol{x})$ is the Jacobian matrix (cf. Eq. (1.8)) of the vector function

$$\boldsymbol{\phi}(\boldsymbol{B}\boldsymbol{x}) = \begin{bmatrix} \phi_1(u_1) \\ \vdots \\ \phi_N(u_N) \end{bmatrix}, \qquad u_i = \boldsymbol{b}_i^{\mathrm{T}}\boldsymbol{x}. \tag{7.23}$$

Since

$$\begin{aligned}
\det \boldsymbol{J}_{\boldsymbol{\phi}}(\boldsymbol{x}) &= \det \begin{bmatrix} \frac{\partial \phi_1}{\partial x_1} & \cdots & \frac{\partial \phi_1}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial \phi_N}{\partial x_1} & \cdots & \frac{\partial \phi_N}{\partial x_N} \end{bmatrix} \\
&= \det \begin{bmatrix} \frac{\mathrm{d}\phi_1}{\mathrm{d}u_1}\frac{\partial u_1}{\partial x_1} & \cdots & \frac{\mathrm{d}\phi_1}{\mathrm{d}u_1}\frac{\partial u_1}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\mathrm{d}\phi_N}{\mathrm{d}u_N}\frac{\partial u_N}{\partial x_1} & \cdots & \frac{\mathrm{d}\phi_N}{\mathrm{d}u_N}\frac{\partial u_N}{\partial x_N} \end{bmatrix} \\
&= \left(\prod_{i=1}^{N} \frac{\mathrm{d}\phi_i}{\mathrm{d}u_i}\right) \det \begin{bmatrix} \frac{\partial u_1}{\partial x_1} & \cdots & \frac{\partial u_1}{\partial x_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial u_N}{\partial x_1} & \cdots & \frac{\partial u_N}{\partial x_N} \end{bmatrix} \\
&= \left(\prod_{i=1}^{N} \frac{\mathrm{d}\phi_i}{\mathrm{d}u_i}\right) \det \boldsymbol{B},
\end{aligned} \tag{7.24}$$

we get for the second term in Eq. (7.22)

$$\mathcal{E}\{\log|\det \boldsymbol{J}_{\boldsymbol{\phi}}(\boldsymbol{x})|\} = \mathcal{E}\left\{\sum_{i=1}^{N} \log \frac{\mathrm{d}\phi_i\left(\boldsymbol{b}_i^{\mathrm{T}}\boldsymbol{x}\right)}{\mathrm{d}u_i}\right\} + \log|\det \boldsymbol{B}|. \tag{7.25}$$

Comparing Eq. (7.25) with Eq. (7.10) and matching the bases of the logarithms, we see that maximization of mutual information between the inputs and the outputs of the neural network as suggested by the principle of information maximization is indeed equivalent to optimizing our cost function $\mathcal{I}_{\mathrm{ML}}$ based on the maximum likelihood method if the derivative of the sigmoids in the neural network equals

the densities of the independent components

$$\frac{\mathrm{d}\phi_i}{\mathrm{d}u_i} = p_{s_i}. \tag{7.26}$$

For a discussion of the Bell-Sejnowski algorithm in the light of neural networks, see the excellent paper by Bell and Sejnowski (1995).

7.2.2 The Natural Gradient Algorithm

Consider the parameter space of the Independent Component Analysis model, whose coordinates are given by the *synaptic weights* b_{ij}. According to Amari's *information geometry*, this parameter space is not Euclidean, i. e. its basis vectors do not form an orthonormal coordinate system. In fact, the parameter space of a neural network has a *Riemannian structure*. In such a space the direction of steepest descent or steepest ascent is not given by the conventional gradient. Rather, the so-called *natural gradient* should be used (Amari, 1997). Without going into detail here, we mention that for a single-layer neural network, the natural gradient can be obtained from the conventional gradient in the update rule by post-multiplying it by the product of matrices

$$\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B}. \tag{7.27}$$

In the case of the update rule of the gradient algorithm in Eq. (7.20), this yields

$$\begin{aligned}\left(\frac{\partial \mathcal{I}_{\mathrm{ML}}(\boldsymbol{B})}{\partial \boldsymbol{B}}\right)\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B} &= \left[\left(\boldsymbol{B}^{\mathrm{T}}\right)^{-1} + \mathcal{E}\left\{\boldsymbol{g}\left(\boldsymbol{B}\boldsymbol{x}\right)\boldsymbol{x}^{\mathrm{T}}\right\}\right]\boldsymbol{B}^{\mathrm{T}}\boldsymbol{B} \\ &= \left(\boldsymbol{I} + \mathcal{E}\left\{\boldsymbol{g}\left(\boldsymbol{y}\right)\boldsymbol{y}^{\mathrm{T}}\right\}\right)\boldsymbol{B},\end{aligned} \tag{7.28}$$

where $\boldsymbol{I}$ denotes the identity matrix.

Note that the algorithm derived using the natural gradient avoids the matrix inversion present in the algorithm in Eq. (7.20). As a consequence, we expect improved stability in cases where the mixing matrix or unmixing matrix is close to singularity (Hyvärinen et al., 2001).

Furthermore, Amari (1997) shows that an online algorithm that uses the natural gradient instead of the conventional gradient is asymptotically as efficient (Section 2.1) as the corresponding batch algorithm, which is able to extract information from all samples in each iteration step (cf. Section 4.5.2).

7.3 Fixed-Point Iteration

Gradient algorithms are not the only way to maximizing the log-likelihood function in Eq. (7.10). In fact, a faster and more reliable algorithm similar to the FastICA algorithms in Section 6.2.2 and in Section 6.3.2 can be derived as follows.

Consider again the cost function $\mathcal{I}_{\mathrm{ML}}$ in Eq. (7.10). If we constrain the estimates of the independent components y_i to be sphered according to Eq. (7.11), it can be shown (Hyvärinen et al., 2001) that the term $\ln|\det \boldsymbol{B}|$ in Eq. (7.10) has to be constant.[1] What remains is a sum of N terms of the form optimized by the fixed-point iteration in Section 6.3.2, when we set $G(\cdot) = \ln p_{s_j}(s_j)$. Accordingly, taking into account the fact that in Chapter 6 the data we used were sphered, an analogous derivation yields (Hyvärinen et al., 2001)

$$\tilde{\boldsymbol{B}}[n] = \boldsymbol{B}[n] + \operatorname{diag}(\zeta_i)\left(\operatorname{diag}(\xi_i) + \mathcal{E}\left\{\boldsymbol{g}(\boldsymbol{y})\,\boldsymbol{y}^{\mathrm{T}}\right\}\right)\boldsymbol{B}[n] \tag{7.31a}$$

$$\boldsymbol{B}[n+1] = \left(\tilde{\boldsymbol{B}}[n]\boldsymbol{C}_{\boldsymbol{x}}\tilde{\boldsymbol{B}}^{\mathrm{T}}[n]\right)^{-1/2}\tilde{\boldsymbol{B}}[n], \tag{7.31b}$$

where

$$\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x}, \qquad \xi_i = -\mathcal{E}\{y_i g(y_i)\}, \qquad \zeta_i = \frac{1}{\mathcal{E}\{y_i g(y_i) - g'(y_i)\}} \tag{7.32}$$

and $\boldsymbol{C}_{\boldsymbol{x}}$ denotes the correlation matrix of the random vector $\boldsymbol{x}$. Note that the update rule in Eq. (7.31a) must be followed by projection on the set of sphering matrices as in Eq. (7.31b) because we required the output $\boldsymbol{y}$ to be sphered.

The discussion of the nonlinear function $\boldsymbol{g}$ in Eq. (7.31) is deferred to the next section.

[1] Taking the determinant on both sides of Eq. (7.11) yields

$$\det \boldsymbol{I} = \det \mathcal{E}\left\{\boldsymbol{B}\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}\right\} \tag{7.29}$$

$$= \det \boldsymbol{B} \det \mathcal{E}\left\{\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}\right\} \det \boldsymbol{B}^{\mathrm{T}}, \tag{7.30}$$

because the determinant of a product of matrices equals the product of the individual determinants. Since the determinant of the identity matrix $\det \boldsymbol{I}$ is unity (Bartsch, 1999), this implies that $\det \boldsymbol{B}$ can be regarded as constant in the optimization problem if $\boldsymbol{y}$ is constrained to be sphered.

7.4 Comparison between Gradient-Based Algorithms and the Fixed-Point Algorithm

Comparing the fixed-point algorithm in Eq. (7.31a) to the natural gradient algorithm in Eq. (7.20), we conclude that the matrices $\operatorname{diag}(\zeta_i)$ and $\operatorname{diag}(\xi_i)$ obviously play the role of the step-size parameter sequence $\alpha[n]$ in the gradient algorithm in Eq. (7.20), thus providing for an optimized automatic adjustment of the step size (Hyvärinen et al., 2001).

Furthermore, the presence of the scalars ζ_i in the FastICA algorithm in Eq. (7.31) makes it unnecessary to separately estimate the nature of the p. d. f.'s of the independent components. In fact, the nonpolynomial moment in Eq. (7.12), which is used in the gradient algorithm to discriminate between the densities $\tilde{p}_{s_j}^{+}$ and $\tilde{p}_{s_j}^{-}$, can also be found in the denominator of the factors ζ_i, where it fulfills an analogous task. As a consequence, the FastICA algorithm maximizing the likelihood of the observations can be run with one fixed nonlinear function, e. g. the hyperbolic tangent (Hyvärinen et al., 2001):

$$\boldsymbol{g}(\boldsymbol{y}) = \begin{bmatrix} \tanh y_1 \\ \vdots \\ \tanh y_N \end{bmatrix} = \begin{bmatrix} \tanh\left(\boldsymbol{b}_1^{\mathrm{T}}\boldsymbol{x}\right) \\ \vdots \\ \tanh\left(\boldsymbol{b}_N^{\mathrm{T}}\boldsymbol{x}\right) \end{bmatrix}. \tag{7.33}$$

7.5 Summary and Outlook

In this chapter, we tried to solve the problem of estimating the Independent Component Analysis model by the maximum likelihood method. After deriving the likelihood function, we found a way to deal with the problem of nonparametric approximation of p. d. f.'s. More specifically, two densities were shown to be sufficient for the purposes of Independent Component Analysis.

From a computational point of view, two approaches to maximizing the likelihood function were followed. The first leads to the well-known Bell-Sejnowski algorithm, while the second yields an efficient algorithm of the fixed-point form.

8 Minimization of Mutual Information

One approach to solving the Independent Component Analysis model is to minimize the statistical dependence of linear combinations of the observable random variables x_i (Comon, 1994, Amari et al., 1996, Hyvärinen et al., 2001), as expressed by

$$\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x}, \tag{8.1}$$

where $\boldsymbol{B}$ is the unmixing matrix and $\boldsymbol{y}$ is the vector of linear combinations whose dependence is to be minimized.

According to what was said in Section 3.2, it is natural to quantify the statistical independence of $\boldsymbol{y}$ by the information-theoretic measure of mutual information $I(\boldsymbol{y})$. To repeat, mutual information $I(\boldsymbol{y})$ of the components of the random vector $\boldsymbol{y}$ is always nonnegative and zero if and only if the components y_i are mutually statistically independent, as is clear from its definition in Eq. (3.15a).

Note that if the observable data $\boldsymbol{x}$ follow the linear generative model of Independent Component Analysis introduced in Section 5.3, the methods derived from the approach discussed in this chapter are obviously able to provide estimates of the original independent components $\boldsymbol{s}$. On the other hand, even in cases where it might not be reasonable or realistic to hypothesize any underlying model in data generation, minimization of mutual information still yields a valuable decomposition of the data (Hyvärinen et al., 2001).

In this chapter, we reveal the connection between the minimization of mutual information and the two approaches discussed in the previous chapters, namely maximization of non-Gaussianity and the maximum-likelihood approach. Since they turn out to be equivalent under certain circumstances, no new algorithms are introduced in this chapter.

8.1 Connection with Maximization of Non-Gaussianity

The approach whereby the ICA model is estimated maximizing non-Gaussianity (cf. Chapter 6) can be justified more rigorously than we did in Section 6.1 by showing that this approach is intimately connected with minimization of mutual information.

More specifically, using the formula for the differential entropy of a linear transformation in Eq. (3.8) and the definition of negentropy in Eq. (3.16), we can express the mutual information $I(\boldsymbol{y})$ of the random vector $\boldsymbol{y}$ as

$$I(\boldsymbol{y}) = \sum_{i=1}^{N} h(y_i) - h(\boldsymbol{y}) \tag{8.2}$$

$$= \sum_{i=1}^{N} h(y_i) - \underbrace{\big(h(\boldsymbol{x}) + \log|\det \boldsymbol{B}|\big)}_{h(\boldsymbol{y})} \tag{8.3}$$

$$= \sum_{i=1}^{N} h(y_i) - h(\boldsymbol{x}) - \log|\det \boldsymbol{B}| \tag{8.4}$$

$$= \sum_{i=1}^{N} \underbrace{\big(h(y_{i,\,\mathrm{Gauss}}) - \mathcal{N}(y_i)\big)}_{h(y_i)} - h(\boldsymbol{x}) - \log|\det \boldsymbol{B}|\,. \tag{8.5}$$

Moreover, for a sphered random vector $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{x}$, the term $\log|\det \boldsymbol{B}|$ in Eq. (8.5) is constant, as was shown in the footnote in Section 7.3. In addition, neither the differential entropy $h(\boldsymbol{x})$ of the mixtures, nor the (fixed) differential entropy of the standardized Gaussian-distributed random variables $y_{i,\,\mathrm{Gauss}}$ depends on $\boldsymbol{B}$:

$$I(\boldsymbol{y}) = -\sum_{i=1}^{N} \mathcal{N}(y_i) + \underbrace{\sum_{i=1}^{N} \underbrace{h(y_{i,\,\mathrm{Gauss}})}_{\frac{1}{2}\log(2\pi \mathrm{e})} - h(\boldsymbol{x}) - \log|\det \boldsymbol{B}|}_{\text{const.}}\,. \tag{8.6}$$

Thus, since negentropy $\mathcal{N}$ is always nonnegative and well suited for measuring non-Gaussianity (cf. Section 3.3), Eq. (8.6) clearly shows that minimization of mutual information $I(\boldsymbol{y})$ is equivalent to maximization of the sum of non-Gaussianities of the estimates of the independent components y_i, if the estimates are constrained to be *uncorrelated* and of unit variance, i. e. sphered (Hyvärinen et al., 2001).

Note, however, that as opposed to the methods that maximize non-Gaussianity, approaches that minimize mutual information always estimate the whole set of

independent components at the same time (Hyvärinen et al., 2001).

8.2 Connection with Maximum Likelihood Estimation

Recall the cost function that we derived in Chapter 7 in the context of estimating the Independent Component Analysis model by the maximum likelihood method, reproduced at this point for convenience of presentation

$$\mathcal{I}_{\mathrm{ML}}(\boldsymbol{B}) = \ln|\det \boldsymbol{B}| + \mathcal{E}\left\{\sum_{i=1}^{N} \ln p_{s_i}\left(\boldsymbol{b}_i^{\mathrm{T}}\boldsymbol{x}\right)\right\}. \tag{8.7}$$

If the unknown functions $p_{s_i}(s_i)$ corresponded to the true p. d. f.'s of the (estimates of the) independent components, we could rewrite the maximum-likelihood cost function in Eq. (8.7) as

$$\mathcal{I}_{\mathrm{ML}}(\boldsymbol{B}) = \log|\det \boldsymbol{B}| - \sum_{i=1}^{N} \underbrace{\mathcal{E}\left\{-\log p_{y_i}\left(\boldsymbol{b}_i^{\mathrm{T}}\boldsymbol{x}\right)\right\}}_{h(y_i)} \tag{8.8}$$

$$= -\left(\sum_{i=1}^{N} h(y_i) - \log|\det \boldsymbol{B}|\right), \tag{8.9}$$

where we used the definition of differential entropy in Eq. (3.3b) and again matched the bases of the logarithms.

Since the maximum-likelihood cost function $\mathcal{I}_{\mathrm{ML}}$ in Eq. (8.9) and the mutual information $I(\boldsymbol{y})$ in Eq. (8.4) differ only in the sign and in the constant term involving the differential entropy $h(\boldsymbol{x})$ of the input vector $\boldsymbol{x}$, which cannot be affected by the optimization anyway, we conclude that the approach to estimating the Independent Component Analysis model by maximization of the log-likelihood $\mathcal{I}_{\mathrm{ML}}$ is equivalent to the minimization of the mutual information $I(\boldsymbol{y})$ of the estimates of the independent components $\boldsymbol{y}$ (Hyvärinen et al., 2001).

8.3 Summary and Outlook

Using the definitions of the various information-theoretic quantities, we showed in this chapter the equivalence of the minimization of mutual information and respectively the maximization of non-Gaussianity and the maximization-likelihood estimation approach, especially if $\boldsymbol{y}$ is constrained to be sphered.

Because of this close connection, the algorithms devised in the two previous chapters can be used for minimization of mutual information as well.

Among the algorithms which maximize non-Gaussianity, though, those with symmetric orthogonalization are the ones to be used, because in minimization of mutual information no order is defined between the individual components. Therefore, in the estimation process there is no reason to prefer one component to another (Hyvärinen et al., 2001).

Alternatively, remember that mutual information $I(\cdot)$ is computed from differential entropies, as seen from Eq. (3.15b). Consequently, it is possible to approximate mutual information $I(\cdot)$ in the same way that we approximated negentropy $\mathcal{N}$ in Section 3.3.2. For an example of such a proceeding consult the paper by Amari et al. (1996); they employ the Gram-Charlier density expansion (cf. Section 3.3.2.1), of course substituting Amari's natural gradient (cf. Section 7.2.2) for the conventional gradient. Conversely, Comon (1994) uses another type of density approximation, namely the so-called *Edgeworth expansion*. Details on both approaches can be found in the respective paper.

9 Tensorial Methods

In this chapter, we discuss ICA methods based on joint forth-order cumulants. (Cumulants are discussed in Section 1.5.2.)

First, the notion of cumulant tensor and cumulant matrix is introduced. Then, we show that diagonalization of cumulant matrices of sphered mixtures yields estimates of the unmixing matrix of the ICA model. In this context, we treat the JADE algorithm and the algorithm of Forth-Order Blind Identification. To conclude, we point out the connection between tensorial methods and the FastICA algorithm devised earlier.

9.1 Cumulant Tensor and Cumulant Matrix

Let us denote the set of all joint forth-order cumulants of the components of the random vector $\boldsymbol{x} = \begin{bmatrix} x_1 & \cdots & x_N \end{bmatrix}^{\mathrm{T}}$ by (Cardoso and Souloumiac, 1993)

$$\boldsymbol{\mathcal{Q}}_{\boldsymbol{x}} = \{\mathrm{cum}\,(x_i, x_j, x_k, x_l) \mid i, j, k, l = 1, \ldots, N\}\,. \tag{9.1}$$

The cumulant set $\boldsymbol{\mathcal{Q}}_{\boldsymbol{x}}$ is in fact a tensor, but for our purposes it suffices to apply index-free notations as an extension of matrix-vector notations familiar from linear algebra (Cardoso, 1990). Note that the cumulants are symmetric in their arguments (Mathews and Sicuranza, 2002).

Then, the ijth entry of the $(N \times N)$-*cumulant matrix* $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{M})$ associated with an arbitrary $(N \times N)$-matrix $\boldsymbol{M}$ with respect to the forth-order cumulants $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}$ of the random vector $\boldsymbol{z}$ is defined by (Cardoso and Souloumiac, 1993)

$$\Big[\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{M})\Big]_{ij} = \sum_{k=1}^{N}\sum_{l=1}^{N} \mathrm{cum}\,(z_i, z_j, z_k, z_l)\, m_{kl}, \qquad i, j = 1, \ldots, N, \tag{9.2}$$

where m_{kl} denotes the klth element of the matrix $\boldsymbol{M}$. Note that the transformation $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{M})$ in Eq. (9.2) is an extension of the usual multiplication of a vector $\boldsymbol{q}$ by a matrix $\boldsymbol{T}$, where in a completely analogous fashion, the ith entry of the resulting

vector $\boldsymbol{p} = \boldsymbol{T}\boldsymbol{q}$ is given by

$$p_i = \sum_{k=1}^{N} t_{ik} q_k. \tag{9.3}$$

For sphered data $\boldsymbol{z}$, the cumulant matrix in Eq. (9.2) can be expressed in an algebraic index-free notation. More specifically, with the definition of forth-order cumulants in terms of moments, we can write the cumulant matrix in the equivalent form

$$\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{M}) = \mathcal{E}\left\{\left(\boldsymbol{z}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{z}\right)\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}}\right\} - \boldsymbol{M} - \boldsymbol{M}^{\mathrm{T}} - \boldsymbol{I}\,\mathrm{tr}\left(\boldsymbol{M}\right), \tag{9.4}$$

where $\boldsymbol{I}$ denotes the identity matrix and $\mathrm{tr}\left(\boldsymbol{M}\right)$ is the trace of the matrix $\boldsymbol{M}$, i. e. the sum of the elements on the diagonal of the matrix $\boldsymbol{M}$.

9.2 Eigenstructure of the Cumulant Tensor

If the ICA model holds, the sphered mixtures can be expressed as $\boldsymbol{z} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{s}$, where the matrix $\boldsymbol{W}^{\mathrm{T}}$ is the inverse of the orthogonal unmixing matrix. Taking into account the properties of cumulants (cf. Section 1.5.3) and the statistical independence of the components s_p, by which $\mathrm{cum}\left(s_i, s_j, s_k, s_l\right) = \mathrm{kurt}\left(s_i\right)$ for $i = j = k = l$ and zero otherwise (Cardoso, 1999), we have

$$\begin{aligned}\left[\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{M})\right]_{ij} &= \sum_{k=1}^{N}\sum_{l=1}^{N}\sum_{p=1}^{N} \mathrm{kurt}\left(s_p\right) w_{pi} w_{pj} w_{pk} w_{pl} m_{kl} \\ &= \sum_{p=1}^{N} \underbrace{w_{pi} w_{pj}}_{\left[\boldsymbol{w}_p \boldsymbol{w}_p^{\mathrm{T}}\right]_{ij}} \mathrm{kurt}\left(s_p\right) \underbrace{\sum_{k=1}^{N}\sum_{l=1}^{N} w_{pk} m_{kl} w_{pl}}_{\boldsymbol{w}_p^{\mathrm{T}}\boldsymbol{M}\boldsymbol{w}_p,\ \text{quadratic form}},\end{aligned} \tag{9.5}$$

where $\boldsymbol{w}_p^{\mathrm{T}}$ denotes the pth row of the unmixing matrix $\boldsymbol{W}$ and $\left[\boldsymbol{w}_p \boldsymbol{w}_p^{\mathrm{T}}\right]_{ij}$ stands for the ijth element of the dyadic product of $\boldsymbol{w}_p$. Clearly, Eq. (9.5) can be expressed in matrix form as

$$\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{M}) = \sum_{p=1}^{N} \mathrm{kurt}\left(s_p\right)\left(\boldsymbol{w}_p^{\mathrm{T}}\boldsymbol{M}\boldsymbol{w}_p\right)\left(\boldsymbol{w}_p \boldsymbol{w}_p^{\mathrm{T}}\right) \tag{9.6a}$$

$$= \boldsymbol{W}^{\mathrm{T}}\,\mathrm{diag}\left(\mathrm{kurt}\left(s_p\right)\left(\boldsymbol{w}_p^{\mathrm{T}}\boldsymbol{M}\boldsymbol{w}_p\right)\right)\boldsymbol{W}. \tag{9.6b}$$

From Eq. (9.6) we see that for sphered data $\boldsymbol{z}$, any cumulant matrix $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{M})$ is diagonalized by the unmixing matrix $\boldsymbol{W}$ (Cardoso and Souloumiac, 1993), i. e. the matrix

$$\boldsymbol{W}\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{M})\boldsymbol{W}^{\mathrm{T}} \tag{9.7}$$

is diagonal when $\boldsymbol{W}$ is an unmixing matrix of the underlying ICA model.

Eq. (9.6) suggests some kind of eigenvalue decomposition of cumulant matrices $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{M})$ so as to get the unmixing matrix $\boldsymbol{W}$ (Cardoso and Souloumiac, 1993). In the following we first show how this can be done using just a single cumulant matrix. Then, we present an algorithm that jointly diagonalizes several cumulant matrices, thereby getting rid of the problem of potential degeneracy of the spectrum[1] of a single cumulant matrix.

9.2.1 Diagonalization of a Single Cumulant Matrix

9.2.1.1 Identity Matrix

In the simplest case, we can exploit the eigenstructure of the cumulant set by diagonalizing a single cumulant matrix. Choosing the cumulant matrix of the identity matrix $\boldsymbol{I}$, we get from Eq. (9.4)

$$\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{I}) = \mathcal{E}\left\{\|\boldsymbol{z}\|^2\,\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}}\right\} - (N+2)\,\boldsymbol{I}. \tag{9.8}$$

Since addition of a scaled identity matrix does not affect the eigenvalue decomposition in any significant way (Hyvärinen et al., 2001), diagonalization of Eq. (9.8) essentially leads to the FOBI algorithm described in Section 9.3. Note that in this case, the cumulant set $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}$ need not even be estimated as a whole (Cardoso and Souloumiac, 1993).

For the choice of the cumulant matrix considered in this subsection, Eq. (9.6) reads

$$\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{I}) = \boldsymbol{W}^{\mathrm{T}}\,\mathrm{diag}\left(\mathrm{kurt}\left(s_p\right)\right)\boldsymbol{W}. \tag{9.9}$$

Thus, the spectrum of the cumulant matrix $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{I})$ is degenerate when the kurtoses of all sources are not distinct. As a consequence, the eigenvalue decomposition does not yield a suitable unmixing matrix (Cardoso and Souloumiac, 1993). These issues will be treated in more detail in Section 9.3.

[1] The set of eigenvalues of a matrix is also referred to as the spectrum of the matrix.

9.2.1.2 One Arbitrary Matrix

As an alternative to diagonalizing the cumulant matrix of the identity matrix like in the previous subsection, consider the diagonalization of a single arbitrary cumulant matrix $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{M})$.

From Eq. (9.6), we get

$$\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{M}) = \boldsymbol{W}^{\mathrm{T}} \operatorname{diag}\left(\operatorname{kurt}(s_p)\left(\boldsymbol{w}_p^{\mathrm{T}} \boldsymbol{M} \boldsymbol{w}_p\right)\right) \boldsymbol{W}, \tag{9.10}$$

i. e. the eigenvalues of the cumulant matrix $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{M})$ are $\operatorname{kurt}(s_p)\left(\boldsymbol{w}_p^{\mathrm{T}} \boldsymbol{M} \boldsymbol{w}_p\right)$. Thus, according to Cardoso and Souloumiac (1993), the eigenvalues are distinct with a high probability.

However, there is no guideline on how to choose the matrix $\boldsymbol{M}$ in order to guarantee nondegeneracy of the spectrum (Cardoso and Souloumiac, 1993, Cardoso, 1999). Moreover, the eigenvalue decomposition of a single cumulant matrix takes into account only a part of the information contained in the whole cumulant set (Cardoso, 1999). These drawbacks are overcome by the diagonalization of more than one matrix described next.

9.2.2 Joint Diagonalization of Several Matrices

In order to extent the notion of diagonalization of a single matrix, consider the cost function (Cardoso and Souloumiac, 1993)

$$\mathcal{I}_{\mathrm{JADE}}(\boldsymbol{\mathcal{M}}, \boldsymbol{W}) = \sum_{\boldsymbol{M}_i \in \boldsymbol{\mathcal{M}}} \left\| \operatorname{diag}\left(\boldsymbol{W} \boldsymbol{M}_i \boldsymbol{W}^{\mathrm{T}}\right) \right\|^2. \tag{9.11}$$

Here, $\boldsymbol{\mathcal{M}}$ is a set of K given matrices $\boldsymbol{M}_i, i = 1, \ldots, K$, and the optimization is with respect to the *orthogonal* matrix $\boldsymbol{W}$. Clearly, $\mathcal{I}_{\mathrm{JADE}}$ is a measure of how well the matrix $\boldsymbol{W}$ is able to jointly diagonalize the matrices in the set of matrices $\boldsymbol{\mathcal{M}}$.

When the set of matrices $\boldsymbol{\mathcal{M}}$ involves cumulant matrices estimated from a finite data set,[2] there is no matrix $\boldsymbol{W}$ that could be able to diagonalize all matrices in the matrix set $\boldsymbol{\mathcal{M}}$ exactly (Cardoso and Souloumiac, 1993). Nevertheless, optimization of the cost function $\mathcal{I}_{\mathrm{JADE}}$ yields a joint *approximative* diagonalization of the matrix set (Cardoso and Souloumiac, 1993). Moreover, there is a connection between $\mathcal{I}_{\mathrm{JADE}}$ and another popular ICA cost function based on autocumulants[3] of the

[2] We will treat this case in a moment.

[3] An autocumulant is a cumulant involving only one index. In the case of forth-order cumulants, this is equal to the kurtosis (Cardoso, 1999). On the other hand, cumulants with at least two different indices are called cross-cumulants

estimates of the independent components (cf. Section 9.2.2.3).

Joint diagonalization of the cost function $\mathcal{I}_{\text{JADE}}$ in Eq. (9.11) reliably solves the problem of Independent Component Analysis if the set of matrices $\boldsymbol{\mathcal{M}}$ satisfies certain requirements. More specifically, suitable choices for the set of matrices are (1) the parallel set and (2) the eigenset associated with the cumulant set of sphered data. In both cases the information of the *whole* cumulant set is involved in the estimation problem (Cardoso and Souloumiac, 1993).

The appeal of joint diagonalization comes from the fact that it can be realized efficiently by an extension of the iterative Jacobi algorithm as described by Cardoso (1999). Then, the computational complexity of diagonalizing N matrices is roughly N times the complexity of diagonalizing a single matrix (Cardoso and Souloumiac, 1993).

9.2.2.1 Joint Diagonalization of the Parallel Set

Cardoso and Souloumiac (1993) show that optimization of $\mathcal{I}_{\text{JADE}}$ by joint diagonalization yields a solution to the problem of Independent Component Analysis model estimation when the set of matrices $\boldsymbol{\mathcal{M}}$ is equal to the so-called *parallel set* (Cardoso and Souloumiac, 1993)

$$\boldsymbol{\mathcal{M}}^{\text{P}} = \left\{ \boldsymbol{\mathcal{Q}}_{\boldsymbol{z}} \left(\mathbf{1}_k \mathbf{1}_l^{\text{T}} \right) \mid k, l = 1, \ldots, N \right\}, \tag{9.12}$$

where $\mathbf{1}_k$ denotes a column vector of length N where all elements are zero except for the kth, which is unity. From the definition of cumulant matrices in Eq. (9.2) we see that $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}} \left(\mathbf{1}_k \mathbf{1}_l^{\text{T}} \right)$ is in fact an $N \times N$-matrix whose ijth entry is given by $\text{cum}\left(z_i, z_j, z_k, z_l\right)$. Therefore, the complete set of N^2 matrices constituting the parallel set $\boldsymbol{\mathcal{M}}^{\text{P}}$ contains the information of the full cumulant set $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}$.

The algorithm is summarized in the following box.

1. Sphere the observations $\boldsymbol{x}$ by a sphering matrix $\boldsymbol{V}$ to get $\boldsymbol{z} = \boldsymbol{V}\boldsymbol{x}$ (cf. Section 5.3.5)
2. Estimate the cumulant set $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}$ in Eq. (9.1)
3. Jointly diagonalize the parallel set $\boldsymbol{\mathcal{M}}^{\text{P}}$ by optimizing the cost function $\mathcal{I}_{\text{JADE}}$, i. e. obtain an orthogonal unmixing matrix $\boldsymbol{W}$ as

$$\boldsymbol{W} = \arg\max_{\boldsymbol{W}} \; \mathcal{I}_{\text{JADE}} \left(\boldsymbol{\mathcal{M}}^{\text{P}}, \boldsymbol{W} \right) \tag{9.13}$$

9.2.2.2 Joint Diagonalization of the Eigenset

Alternatively, the joint diagonalization can be done on the so-called eigenset $\boldsymbol{\mathcal{M}}^{\mathrm{e}}$ (Cardoso and Souloumiac, 1993) stemming from the concept of eigenmatrices.

According to Cardoso (1990), an *eigenmatrix* associated with the cumulant set $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}$ is a matrix $\boldsymbol{M}$ such that

$$\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{M}) = \lambda \boldsymbol{M}, \tag{9.14}$$

where λ denotes the scalar *eigenvalue* corresponding to the eigenmatrix $\boldsymbol{M}$. In other words, an eigenmatrix is not changed by an application of the transformation defined by the cumulant matrix, except for a scaling by λ. Then, the *eigenset*

$$\boldsymbol{\mathcal{M}}^{\mathrm{e}} = \{\lambda_i \boldsymbol{M}_i \mid \boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}(\boldsymbol{M}_i) = \lambda_i \boldsymbol{M}_i, i = 1, \ldots, N\}. \tag{9.15}$$

Cardoso (1990) states that among the N^2 eigenvalues associated with the cumulant set $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}$, only N eigenvalues are nonzero. Consequently, there exist exactly N nontrivial eigenmatrices. Thus, once all eigenmatrices corresponding to nonzero eigenvalues are found,[4] the joint diagonalization of the cost function $\mathcal{I}_{\mathrm{JADE}}$ involves just N matrices, instead of the N^2 matrices needed when the parallel set $\boldsymbol{\mathcal{M}}^{\mathrm{p}}$ is used (Cardoso and Souloumiac, 1993).

In the literature the resulting algorithm is called the JADE algorithm (Joint Approximative Diagonalization of eigenmatrices). We summarize the necessary steps in the following box.

1. Sphere the observations $\boldsymbol{x}$ by a sphering matrix $\boldsymbol{V}$ to get $\boldsymbol{z} = \boldsymbol{V}\boldsymbol{x}$ (cf. Section 5.3.5)
2. Estimate the cumulant set $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}$ in Eq. (9.1)
3. Compute the eigenmatrices corresponding to nonzero eigenvalues
4. Jointly diagonalize the eigenset $\boldsymbol{\mathcal{M}}^{\mathrm{e}}$ by optimizing the cost function $\mathcal{I}_{\mathrm{JADE}}$, i. e. obtain an orthogonal unmixing matrix $\boldsymbol{W}$ as

$$\boldsymbol{W} = \arg\max_{\boldsymbol{W}} \mathcal{I}_{\mathrm{JADE}}(\boldsymbol{\mathcal{M}}^{\mathrm{e}}, \boldsymbol{W}) \tag{9.16}$$

[4] To compute the N eigenmatrices in a straightforward way, one can rearrange the elements of the cumulant set $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}$ in an $N^2 \times N^2$ matrix and then perform an ordinary eigenvalue decomposition of this $N^2 \times N^2$ matrix (Cardoso and Souloumiac, 1993).

9.2.2.3 Connection to Related Approaches

To illustrate the connection between the joint diagonalization methods and other approaches, consider a random vector $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{z}$, where $\boldsymbol{z}$ is sphered and the unmixing matrix $\boldsymbol{W}$ is orthogonal. If the variables of such a random vector $\boldsymbol{y}$ are mutually statistically independent, their joint forth-order cumulants are given by

$$\operatorname{cum}(y_i, y_j, y_k, y_l) = \begin{cases} \operatorname{kurt}(y_i) & \text{for } i = j = k = l, \\ 0 & \text{otherwise.} \end{cases} \tag{9.17}$$

Moreover, note that the total sum of squares of the joint forth-order cumulants is not changed by an orthogonal transformation $\boldsymbol{W}$ (Cardoso and Souloumiac, 1993). As a consequence, a suitable cost function for ICA model estimation could be to maximize the sum of squares of the autocumulants, i. e.

$$\boldsymbol{W} = \arg\max_{\boldsymbol{W}} \sum_{i=1}^{N} \big(\operatorname{cum}(y_i, y_i, y_i, y_i)\big)^2, \tag{9.18}$$

or, equivalently, to minimize the sum of squares of the cross-cumulants, i. e.

$$\boldsymbol{W} = \arg\min_{\boldsymbol{W}} \sum_{ijkl \neq iiii} \big(\operatorname{cum}(y_i, y_j, y_k, y_l)\big)^2, \tag{9.19}$$

where the summation is over all cumulants with different indices.

Cardoso and Souloumiac (1993) prove that joint diagonalization of both the parallel set $\boldsymbol{\mathcal{M}}^{\mathrm{p}}$ and the eigenset $\boldsymbol{\mathcal{M}}^{\mathrm{e}}$ is equivalent to optimizing the cost function

$$\mathcal{I}_{\mathrm{JADE}} = \sum_{i=1}^{N} \sum_{k=1}^{N} \sum_{l=1}^{N} \big(\operatorname{cum}(y_i, y_i, y_k, y_l)\big)^2 \tag{9.20}$$

when the unmixing matrix $\boldsymbol{W}$ is constrained to be orthogonal. Note that the JADE cost function in Eq. (9.20) is similar to Eq. (9.18), but—as we have seen in this Chapter—efficient methods can be devised for its optimization in terms of joint approximative diagonalization using the Jacobi technique (Cardoso and Souloumiac, 1993).

9.3 Forth-Order Blind Identification

The Forth-order blind identification (FOBI) algorithm, introduced by Cardoso (1989), provides a computationally very simple approach to solving the Indepen-

dent Component Analysis model. Its connection to the principle of diagonalization of a single cumulant matrix was already discussed in Section 9.2.1.1.

In the FOBI algorithm, an estimate of the unmixing matrix is directly obtained from the eigenvalue decomposition of the so-called *weighted correlation matrix* $\boldsymbol{\Omega}$ (Cardoso, 1989) of the sphered mixtures $\boldsymbol{z}$, given by

$$\boldsymbol{\Omega} = \mathcal{E}\left\{\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}} \left\|\boldsymbol{z}\right\|^2\right\}. \tag{9.21}$$

In fact, assuming the ICA model (cf. Section 5.3) and exploiting the properties of the expectation operator (cf. Section 1.2), we can express the weighted correlation matrix $\boldsymbol{\Omega}$ of a sphered vector $\boldsymbol{z}$ as

$$\begin{aligned}
\boldsymbol{\Omega} &= \mathcal{E}\left\{\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}} \left\|\boldsymbol{z}\right\|^2\right\} \\
&= \mathcal{E}\left\{\boldsymbol{V}\boldsymbol{A}\boldsymbol{s}\boldsymbol{s}^{\mathrm{T}} \left(\boldsymbol{V}\boldsymbol{A}\right)^{\mathrm{T}} \left\|\boldsymbol{V}\boldsymbol{A}\boldsymbol{s}\right\|^2\right\} \\
&= \mathcal{E}\left\{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{s}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{W} \left\|\boldsymbol{W}^{\mathrm{T}}\boldsymbol{s}\right\|^2\right\} \\
&= \mathcal{E}\left\{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{s}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{W} \left(\boldsymbol{s}^{\mathrm{T}} \underbrace{\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}}}_{\boldsymbol{I}} \boldsymbol{s}\right)\right\} \\
&= \mathcal{E}\left\{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{s}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{W} \left\|\boldsymbol{s}\right\|^2\right\} \\
&= \mathcal{E}\left\{\boldsymbol{W}^{\mathrm{T}}\boldsymbol{s}\boldsymbol{s}^{\mathrm{T}} \left\|\boldsymbol{s}\right\|^2 \boldsymbol{W}\right\} \\
&= \boldsymbol{W}^{\mathrm{T}}\mathcal{E}\left\{\boldsymbol{s}\boldsymbol{s}^{\mathrm{T}} \left\|\boldsymbol{s}\right\|^2\right\} \boldsymbol{W}.
\end{aligned} \tag{9.22}$$

Moreover, using the mutual statistical independence of the components s_i and the fact that the independent components s_i have zero-mean and unit variance (cf. Section 5.3.4), we can transform the expected value of the weighted dyadic product of $\boldsymbol{s}$ to yield

$$\boldsymbol{\Omega} = \boldsymbol{W}^{\mathrm{T}} \operatorname{diag}\left(\mathcal{E}\left\{s_i^2 \left\|\boldsymbol{s}\right\|^2\right\}\right) \boldsymbol{W} \tag{9.23a}$$

$$= \boldsymbol{W}^{\mathrm{T}} \operatorname{diag}\left(\mathcal{E}\left\{s_i^4\right\} + N - 1\right) \boldsymbol{W}. \tag{9.23b}$$

But Eq. (9.23b) is exactly of the form of the eigenvalue decomposition of the weighted correlation matrix $\boldsymbol{\Omega}$, the expressions $(\mathcal{E}\{s_i^4\} + N - 1)$ and the matrix $\boldsymbol{W}^{\mathrm{T}}$ corresponding to its eigenvalues and its eigenvectors, respectively. As a consequence, if all eigenvalues of the weighted correlation matrix $\boldsymbol{\Omega}$ are distinct, the eigenvectors of $\boldsymbol{\Omega}$ put along the rows of the orthogonal unmixing matrix $\boldsymbol{W}$ solve the ICA model estimation problem (Cardoso, 1989, Hyvärinen et al., 2001).

On the other hand, consider the case when several independent components s_i

possess identical forth-order moments $\mathcal{E}\{s_i^4\}$, so that the associated eigenvalues of the decomposition in Eq. (9.23b) are identical as well. Then, the eigenvalue decomposition is no longer unique (Cardoso, 1989). As a consequence, the FOBI algorithm fails to separate those independent components which have the same forth-order moment, while the components with different forth-order moments can still be estimated (Cardoso, 1989).

As a possible remedy, Cardoso (1989) generalizes the weighted correlation matrix introducing a scalar nonlinear function $f(\cdot)$ and with it, higher than forth-order information on the data:

$$\boldsymbol{\Omega}_f = \mathcal{E}\left\{\boldsymbol{z}\boldsymbol{z}^{\mathrm{T}} f\left(\|\boldsymbol{z}\|^2\right)\right\}. \tag{9.24a}$$

As one can show, for symmetrically distributed independent components, the eigenvalue decomposition of this generalized weighted correlation matrix $\boldsymbol{\Omega}_f$ again yields an unmixing matrix, since

$$\boldsymbol{\Omega}_f = \boldsymbol{W}^{\mathrm{T}} \operatorname{diag}\left(\mathcal{E}\left\{s_i^2 f\left(\|\boldsymbol{s}\|^2\right)\right\}\right) \boldsymbol{W}. \tag{9.24b}$$

Note that for the linear function $f(u) = u$, the generalized FOBI algorithm in Eq. (9.24) reduces to the form in Eq. (9.23).

However, not even the generalized algorithm is able to separate independent components whose *distributions* are completely identical, since then the eigenvalues of the decomposition in Eq. (9.24b) will always be equal, no matter which function $f(\cdot)$ we choose.

As opposed to most other algorithms presented in this thesis, FOBI is not an iterative algorithm. Rather, the eigenvalue decomposition of the weighted correlation matrix is computed only once, immediately yielding the estimate of the unmixing matrix.

From a computational point of view, the complexity of the FOBI algorithm is essentially that of a sphering transformation.

Computer Experiment 9.1 (Forth-Order Blind Identification)
In this computer experiment, we investigate the performance and the limitations of the FOBI algorithm in Eq. (9.23) for the two-dimensional case.

To this end, we consider two different mixing environments. More specifically, in Case 1, two independent components $\boldsymbol{s}$ with different p. d. f.'s were involved (standardized uniform density and a standardized trapezoidal p. d. f., respectively). On the other hand, the independent components in Case 2 are both uniformly

distributed and therefore have identical forth-order moments.

In both cases, $K = 30\,000$ data samples $\boldsymbol{x}$ are generated according to the Independent Component Analysis model $\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}$. The entries of the mixing matrix $\boldsymbol{A}$ are uniformly distributed around the origin. The mixtures are then sphered using the method described in Section 5.3.5.2, and estimates of the independent components $\boldsymbol{y}$ are computed with the FOBI algorithm.

For both cases, the experiment is repeated 20 times, each time with new samples of the components $\boldsymbol{s}$ and a new mixing matrix $\boldsymbol{A}$.

We rate the separating algorithm by means of the performance index $\mathcal{P}$ proposed, e. g., in Hyvärinen et al. (2001)

$$\mathcal{P} = \sum_{i=1}^{N} \left(\sum_{j=1}^{N} \frac{|p_{ij}|}{\max_k |p_{ik}|} - 1 \right) + \sum_{j=1}^{N} \left(\sum_{i=1}^{N} \frac{|p_{ij}|}{\max_k |p_{kj}|} - 1 \right). \tag{9.25}$$

Here, the p_{ij} are the coefficients of the matrix

$$\boldsymbol{P} = \boldsymbol{W}\boldsymbol{V}\boldsymbol{A}, \tag{9.26}$$

where $\boldsymbol{W}$ denotes the estimated unmixing matrix and $\boldsymbol{V}$ is the matrix of the sphering transformation. Remember from Section 5.3.4 that the matrix $\boldsymbol{P}$ ideally consists of only one entry per row and column, in which case $\mathcal{P}$ is identically zero. Otherwise, $\mathcal{P}$ is positive.

	$\operatorname{kurt}(s_1)$	$\operatorname{kurt}(s_2)$	$\bar{\mathcal{P}}$	$\max_k \mathcal{P}_k$
Case 1	-1.200	-0.816	0.096	0.233
Case 2	-1.202	-1.202	2.297	3.947

Table 9.1: Performance of FOBI algorithm operating in two different mixing environments.

The results of the computer experiment are summarized in Tab. 9.1, were $\bar{\mathcal{P}}$ denotes the performance index averaged over the twenty repetitions. In Case 1, since the kurtoses of the independent components are different, the FOBI algorithm is able to yield acceptable estimates, which is confirmed by an average performance index of small absolute value. Conversely, in Case 2 both kurtoses are equal, and the algorithm fails completely. ■

9.4 Modified Power Method for Diagonalization of Eigenmatrices

Let us consider in more detail the concept of eigenmatrices introduced in Section 9.2.2.2. In fact, the rows of the unmixing matrix can be obtained directly from the eigenvalue decomposition of eigenmatrices.

Using Eq. (9.6), we immediately verify that the eigenmatrices of the cumulant set $\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}$ are in fact given by the rank-one matrices $\boldsymbol{w}_i \boldsymbol{w}_i^{\mathrm{T}}$ constructed from the ith row $\boldsymbol{w}_i^{\mathrm{T}}$ of the unmixing matrix $\boldsymbol{W}$. The corresponding eigenvalues are given by the kurtoses $\operatorname{kurt}(s_i)$ of the independent components s_i. Thus, (Hyvärinen et al., 2001, Cardoso, 1990)

$$\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}} \left(\boldsymbol{w}_i \boldsymbol{w}_i^{\mathrm{T}} \right) = \operatorname{kurt}(s_i) \, \boldsymbol{w}_i \boldsymbol{w}_i^{\mathrm{T}}, \qquad i = 1, \ldots, N. \tag{9.27}$$

Remember that only exactly N eigenvalues of the cumulant set are nonzero. Therefore, when we have found a matrix to be an eigenmatrix $\boldsymbol{w}_i \boldsymbol{w}_i^{\mathrm{T}}$ of the cumulant tensor, an eigenvalue decomposition of this eigenmatrix yields an estimate for one independent component.

If all eigenvalues of the cumulant set are distinct, every eigenmatrix corresponds to a different row of the unmixing matrix. On the other hand, let us inspect the case that the algebraic multiplicity $\mathcal{A}_i$ of an eigenvalue λ_i is greater than unity. Then, there are $\mathcal{A}_i$ eigenmatrices $\boldsymbol{M}_i$ associated with the eigenvalue λ_i. These eigenmatrices $\boldsymbol{M}_i$ are in general linear combinations of $\mathcal{A}_i$ vectors $\boldsymbol{w}_{i(j)}$, $j = 1, \ldots, \mathcal{A}_i$, that belong to this eigenvalue λ_i. Thus,

$$\boldsymbol{M}_i = \sum_{j=1}^{\mathcal{A}_i} c_j \boldsymbol{w}_{i(j)} \boldsymbol{w}_{i(j)}^{\mathrm{T}}. \tag{9.28}$$

Consequently, also in the case of equal eigenvalues, we get the desired projectors $\boldsymbol{w}_{i(j)}$ by an eigenvalue decomposition of the matrices $\boldsymbol{M}_i$ (Hyvärinen et al., 2001).

Interestingly enough, Hyvärinen et al. (2001) show that a power method[5] tailored to the specific structure of the problem at hand leads exactly to the FastICA algorithm with the cubic function that maximizes the absolute value of the non-Gaussianity of the projection y_i (cf. Section 6.2.2). Consult the reference for more details.

[5] The power method is a simple iterative method for finding the eigenvector of a matrix corresponding to the eigenvalue with the greatest absolute value. An arbitrary starting vector is iteratively transformed by the matrix whose eigenvector we wish to compute and normalized to unit length until convergence (Kreyszig, 1999).

9.5 Summary

In this chapter, we discussed methods for solving the Independent Component Analysis model estimation problem that rely on direct evaluation of forth-order cumulants.

We showed that the eigenvalue decomposition of cumulant matrices directly gives the desired unmixing matrix. Note that throughout this chapter, only sphered mixtures were considered, which once more shows the utility of sphering as a preprocessing step in ICA. Different algorithms result from the specific choice of which cumulant matrix or which cumulant matrices to take in the diagonalization.

More specifically, choosing the cumulant matrix of the identity matrix leads to the computationally simple FOBI algorithm. However, care must be taken when several independent components have similar distributions.

On the other hand, no such problems arise when we use joint approximative diagonalization of several cumulant matrices. This approach, known as the JADE algorithm, has a computationally efficient implementation in terms of Jacobi rotations.

To conclude, we saw that successive eigenvalue decomposition of cumulant eigenmatrices leads to the FastICA algorithm derived earlier, which gives another justification of the intuitive principle of maximization of non-Gaussianity.

List of Symbols

$\boldsymbol{A}$	Square, non-singular mixing matrix
$\mathcal{A}$	Algebraic multiplicity
α	Step-size parameter in gradient algorithms
$\boldsymbol{B}$	Nonsingualar unmixing matrix
b	Bias of an estimate
β	Positive number in step-size parameter sequence
C_{ij}	Covariance of the random variables x_i and x_j
$\boldsymbol{C}_{\boldsymbol{x}}$	Covariance matrix of the random vector $\boldsymbol{x}$
$\mathrm{cum}\,(x)$	Cumulant of the random variable x
$\boldsymbol{D}$	Diagonal matrix of variances
$\Delta\boldsymbol{w}[n]$	Newton correction
$\delta_{\mathrm{abs}},\ \delta_{\mathrm{rel}}$	Accuracy based on the absolute respectively the relative error
det	Determinant
$\mathrm{diag}\,(\cdots)$	Diagonal matrix consisting of the values given as arguments
$\boldsymbol{E}$	Orthogonal matrix of Eigenvectors
$\mathcal{E}\{\cdot\}$	Mathematical expectation
$\boldsymbol{\varepsilon}[n]$	Departure of the solution in iteration step n from the true solution
η_x	Mean value of the random variable x
$\boldsymbol{\eta}_{\boldsymbol{x}}$	Mean vector of the random vector $\boldsymbol{x}$
$\mathcal{G}_{\boldsymbol{x}}$	Group of random variables $\boldsymbol{x}$
grad	Vector of partial derivatives of multivariate function
γ	Estimate of the kurtosis
$H(x)$	Entropy of the random variable x
$\boldsymbol{H}_{\mathcal{I}}(\cdot)$	Hessian matrix of the function $\mathcal{I}(\cdot)$
$h(\boldsymbol{x})$	Differential entropy of the random vector $\boldsymbol{x}$
$I(\boldsymbol{x})$	Mutual information of the random vector $\boldsymbol{x}$
$\mathcal{I}$	Cost function in optimization problems
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{J}_{\boldsymbol{g}}(\boldsymbol{x})$	Jacobian matrix of the vector function $\boldsymbol{g}$
j	Imaginary unit
K	Number of available samples
$\mathrm{kurt}\,(x)$	Kurtosis of the random variable x
$\mathcal{L}\,(\boldsymbol{\vartheta})$	Log-likelihood function
$\ell\,(\boldsymbol{\vartheta})$	Likelihood function
λ	Eigenvalue
λ_{MP}	Lagrange multiplier
ln	Natural logarithm

M	Number of unknown sources in general unmixing environment
$\boldsymbol{\mathcal{M}}$	Set of matrices
$\boldsymbol{\mathcal{M}}^{\mathrm{p}},\ \boldsymbol{\mathcal{M}}^{\mathrm{e}}$	Parallel set and Eigenset, respectively
$\mathrm{mom}\,(x)$	Moment of the random variable x
$\mathcal{N}\,(\boldsymbol{x})$	Negentropy of the random vector $\boldsymbol{x}$
N	Number of independent component estimates in ICA
n	Iteration step in iterative algorithms
ν	Standardized Gaussian-distributed random variable
$\mathcal{P}$	Performance index used in ICA algorithm rating
$p_{\boldsymbol{x}}(\boldsymbol{x})$	Probability Density Function of the random vector $\boldsymbol{x}$
$\boldsymbol{\mathcal{Q}}_{\boldsymbol{x}}$	Set of joint forth-order cumulants of the random vector $\boldsymbol{z}$
$\boldsymbol{\mathcal{Q}}_{\boldsymbol{z}}\,(\boldsymbol{M})$	Cumulant matrix, transformation of matrix $\boldsymbol{M}$ with respect to the cumulants of $\boldsymbol{z}$
R_{ij}	Correlation between the random variables x_i and x_j
$\boldsymbol{R}_{\boldsymbol{x}}$	Correlation matrix of the random vector $\boldsymbol{x}$
$\mathbb{R}_{\geq 0}$	Set of real nonnegative numbers
ρ	Nonpolynomial moment in maximization of non-Gaussianity
$\boldsymbol{s}$	Vector of mutually statistically independent random variables
σ_x^2	Variance of the random variable x
$\mathrm{sign}\,(x)$	Absolute value of x
$\mathrm{tr}\,(M)$	Trace of matrix $\boldsymbol{M}$, i. e. sum of elements on the diagonal of $\boldsymbol{M}$
$\boldsymbol{\vartheta}$	Parameter vector
$\hat{\boldsymbol{\vartheta}}$	Estimate of a parameter vector
$\boldsymbol{V}$	Sphering transformation matrix
$\boldsymbol{W}$	Orthogonal unmixing matrix
$\boldsymbol{w}^*$	Fixed-point in fixed-point iteration
$\boldsymbol{x}$	Vector of linear combinations of mutually statistically independent random variables
$\boldsymbol{y}$	Estimates of independent components
$\Phi(\cdot)$	Characteristic function
$\boldsymbol{\phi}(\cdot)$	Vector function of nonlinear sigmoid functions in Bell-Sejnowski algorithm
$\phi(\cdot)$	Nonlinear sigmoid function in Bell-Sejnowski algorithm
$\boldsymbol{\varphi}(\cdot)$	Update rule
ψ	Angle of a unit-norm vector and the positive x-axis in the twodimensional plane
χ	Nonzero real number to show an ambiguity in ICA model estimation
ζ_i, ξ_i	Scalars in fixed-point iteration, maximum likelihood method
$\boldsymbol{z}$	Vector of sphered combinations of mutually statistically independent random variables, i. e. the components of $\boldsymbol{z}$ are mutually uncorrelated and have unit variance
$\boldsymbol{\Omega}$	Weighted correlation matrix in FOBI algorithm
$\boldsymbol{\Omega}_f$	Weighted correlation matrix in generalized FOBI algorithm
ω_i	Independent variables in characteristic function

Bibliography

Amari, S.-i. (1997). "Neural Learning in Structured Parameter Spaces — Natural Riemannian Gradient", *in Advances in Neural Information Processing Systems*, eds. M. C. Mozer, M. I. Jordan, and T. Petsche, vol. 9, The MIT Press, pp. 127–133.

Amari, S.-i., A. Cichocki, and H. H. Yang (1996). "A New Learning Algorithm for Blind Signal Separation", *in Advances in Neural Information Processing Systems*, eds. D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, vol. 8, The MIT Press, Cambridge, pp. 757–763.

Bartsch, H.-J. (1999). *Taschenbuch mathematischer Formeln*, 19th ed., Fachbuchverlag Leipzig im Carl Hanser Verlag.

Bell, A. J. and T. J. Sejnowski (1995). "An Information-Maximisation Approach to Blind Separation and Blind Deconvolution", *Neural Computation*, vol. 7, no. 6, pp. 1129–1159.

Cardoso, J.-F. (1989). "Source separation using higher order moments", *in Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, Glasgow, UK, pp. 2109–2112.

Cardoso, J.-F. (1990). "Eigen-structure of the fourth-order cumulant tensor with application to the blind source separation problem", *in Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing*, Albuquerque, New Mexico, pp. 2655–2658.

Cardoso, J.-F. (1999). "High-Order Contrasts for Independent Component Analysis", *Neural Computation*, vol. 11, no. 1, pp. 157–192.

Cardoso, J.-F. and A. Souloumiac (1993). "Blind beamforming for non Gaussian signals", *IEE Proceedings-F*, vol. 140, no. 6, pp. 362–370.

Comon, P. (1994). "Independent Component Analysis, a New Concept?", *Signal Processing*, vol. 36, no. 3, pp. 287–314.

Deuflhard, P. and A. Hohmann (2002). *Numerische Mathematik I*, 3rd ed., Walter de Gruyter, Berlin, New York.

Dickreiter, M. (1997). *Handbuch der Tonstudiotechnik*, 6th ed., K. G. Saur, München.

Haykin, S. (2002). *Adaptive Filter Theory*, 4th ed., Prentice Hall.

Hyvärinen, A. (1999). "Fast and Robust Fixed-Point Algorithms for Independent Component Analysis", *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634.

Hyvärinen, A. and E. Oja (1997). "A Fast Fixed-Point Algorithm for Independent Component Analysis", *Neural Computation*, vol. 9, pp. 1483–1492.

Hyvärinen, A. and E. Oja (1998). "Independent Component Analysis by General Non-linear Hebbian-like Learning Rules", *Signal Processing*, vol. 64, no. 3, pp. 301–313.

Hyvärinen, A., J. Karhunen, and E. Oja (2001). *Independent Component Analysis*, John Wiley & Sons.

Kreyszig, E. (1999). *Advanced Engineering Mathematics*, 8th ed., John Wiley & Sons.

Mathews, V. J. and G. L. Sicuranza (2002). *Polynomial Signal Processing*, John Wiley & Sons, New York.

Papoulis, A. (1991). *Probability, Random Variables and Stochastic Processes*, 3th ed., McGraw-Hill, Inc.