

Chapter 2

ICA and ICAMM Methods

2.1 Introduction

The seminal work of the research in ICA was provided by Jutten in [1–4]. Independent component analysis (ICA) aims to separate hidden sources from their observed linear mixtures without any prior knowledge. The only assumption about the sources is that they are mutually independent [5]. Thus, the goal is blind source estimation; although it has been recently alleviated by incorporating prior knowledge about the sources into the ICA model in the so-called semi-blind source separation (see for instance [6–8]). This technique has been widely used in many fields of application such as telecommunications, bioengineering, and material testing [5]. There is extensive literature that reviews and provides taxonomies and comparisons about the large number of ICA algorithms that have been developed during the last two decades (see for example [5, 9–13]). Therefore, in this chapter, instead of undertaking an exhaustive review of the methods, we will focus on reviewing the following: the ICA basic concepts, some ICA algorithms that will be used for comparison with those proposed in this work, and existing ICAMM algorithms.

The standard noiseless instantaneous ICA formulates a $M \times 1$ random vector \mathbf{x} by linear mixtures of M random variables that are mutually independent s_1, \dots, s_M whose distributions are totally unknown. That is, for $\mathbf{s} = (s_1, \dots, s_M)^T$ and some matrix \mathbf{A}

$$\mathbf{x} = \mathbf{A}\mathbf{s} \quad (2.1)$$

The essential principle is to estimate the so-called mixing matrix \mathbf{A} , or equivalently $\mathbf{B} = \mathbf{A}^{-1}$ (the demixing matrix). The matrix \mathbf{A} contains the coefficients of the linear transformation that represents the transfer function from sources to observations. Thus, given N i.i.d. observations $(\mathbf{x}_1, \dots, \mathbf{x}_N)$ from the distribution of \mathbf{x} , \mathbf{A}^{-1} can be applied to separate each of the sources $\mathbf{s}_i = \mathbf{B}_i \mathbf{x}$, where \mathbf{B}_i is the i th

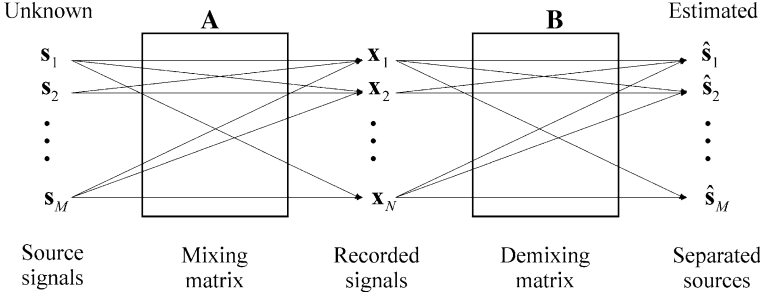


Fig. 2.1 The instantaneous mixing and unmixing model for BSS-ICA

row of **B**. This can be seen as a projection pursuit density estimation problem to find M directions such that the corresponding projections are the most mutually independent. For the sake of simplicity, we will assume the square problem (the same number of sources as mixtures, thus the order of **A** is $M \times M$). Figure 2.1 shows a schema that illustrates the instantaneous mixing and unmixing model for BSS-ICA.

Furthermore, the instantaneous linear model can be applied in the frequency domain for the analysis of convolutive mixtures. Applying the Fourier transform to both sides of Eq. (2.1), we obtain the following frequency expression

$$\mathbf{x}(\omega) = \mathbf{A}(\omega)\mathbf{s}(\omega) \quad (2.2)$$

where $\mathbf{x}(\omega) = \text{FT}\{\mathbf{x}\}$, $\mathbf{A}(\omega) = \text{FT}\{\mathbf{A}\}$, and $\mathbf{s}(\omega) = \text{FT}\{\mathbf{s}\}$ are the Fourier transforms of the observation vector, mixing matrix, and source vector, respectively. The time and frequency domain ICA models are equivalent, but the coefficients of the transfer matrix may vary with ω (see for instance [14] and the references within). An attempt to generalize the BSS algorithms for MIMO signal processing that exploits three signal properties nonwhiteness, nongaussianity, and nonstationarity in an information theoretic cost function has been recently formulated in [15, 16]. In some cases, the convolutive model can be solved as an “instantaneous” problem for selected frequencies. The frequency component permutation problem is thus avoided. The frequencies to be analyzed are selected according to the application; for instance, in a detection problem, the frequencies around the working frequency of the excitation sensor are in the band of interest. We include an example of this frequency ICA analysis applied in NDT in Chap. 5.

It is well-known that **A** is identifiable, up to scaling and permutation of columns, when **s** has at most one Gaussian component and **A** is assumed to be non-singular [17]. The restriction in Gaussian components is explained by the central limit theorem, considering that a linear mixture of independent random variables is more Gaussian than the original variables. Thus, to specify **B** uniquely, we need to put some scale and permutation constraints either on **s** or on **B**. Because of the ICA indeterminacies the sources are usually assumed to be unit variance. Also, it is

commonly assumed that both the observed variables and the independent components have zero mean.

The source independence is expressed as the joint probability, which is the product of the marginal densities $p(\mathbf{s}) = \prod_i p_i(\mathbf{s}_i)$. Since the source distribution is not available, the independence is represented in different ways, e.g., using the following statistics

$$E[g_i(\hat{\mathbf{s}}_i)g_j(\hat{\mathbf{s}}_j)] = 0 \quad (2.3)$$

for any non-linear function g_i , i.e., all the cross cumulants must be zero.

Most of the existing algorithms used to estimate the matrix \mathbf{A} can be organized in two categories. The first category of methods directly approximates the distributions of hidden sources within a specified class of distributions and minimizes a cost function the so-called contrast function, or simply contrast, which is generically denoted $\phi(\hat{\mathbf{s}})$ such as mutual information, likelihood function, or equivalents [5, 17–21]. The design of the ICA algorithms includes the formulation of a contrast function that has to be minimized through an optimization procedure. The contrast function is a real valued function of the estimated sources \mathbf{s} , which yields a minimum value when the independence is attained. The second category of methods optimizes other contrast functions without approximating distributions explicitly. These functions can be, for instance, nongaussianity (using neguentropy or kurtosis), and nonlinear correlation among estimated sources [2, 22].

In several ICA algorithms, the data are first whitened (also called sphering), which requires the covariance matrix of the data to be unity. It is well-known that the demixing matrix can be factorized as the product of a whitening and an orthogonal matrix, i.e., $\mathbf{B} = \mathbf{V}\mathbf{W}$, where \mathbf{V} is the whitening matrix and \mathbf{W} is the orthogonal one. The mixtures are first whitened in order to exhaust the second order moments (signals are forced to be uncorrelated). The whitened vector is expressed as $\mathbf{z} = \mathbf{V}\mathbf{A}\mathbf{s}$, with $E[\mathbf{z}\mathbf{z}^T] = \mathbf{I}$, and the whiteness constraint $E[\hat{\mathbf{s}}\hat{\mathbf{s}}^T] = \mathbf{I}$, with $\hat{\mathbf{s}}$ being the estimated sources. Thus, the ICA model, considering a prewhitening step, is expressed as

$$\hat{\mathbf{s}} = \mathbf{B}\mathbf{x} = \mathbf{W}\mathbf{V}\mathbf{x} \quad (2.4)$$

The orthogonal matrix \mathbf{W} is a rotation of the joint density, which has to maximize the nongaussianity of the marginal densities, thus maximizing a measure of independence. The rotation step keeps the covariance of $\hat{\mathbf{s}}$ equal to the identity, thus preserving the whiteness, hence, the decorrelation of the components. Prewhitening is an optional step to estimate the ICA parameters; in fact, recent methods avoid a prewhitening phase and directly attempt to compute a non-orthogonal diagonalizing congruence (see e.g., [23, 24]). A discussion about connections between mutual information, entropy, and non Gaussianity in a general framework without imposing whitening is presented in [25]. However, prewhitening in ICA algorithms has been reported to provide algorithmic computational advantages (see e.g., [26, 27]).

The algorithms used in ICA can be deterministic or stochastic. The deterministic algorithms always produce the same results (usually exploiting the algebraic structure of the matrices involved) whereas the stochastic algorithms are adaptive starting from a random unmixing matrix that is updated iteratively. The updating can be made for every observation (on-line) or for the whole set of observations (off-line). Thus, the results of stochastic algorithms vary in different executions of the algorithm. The reliability of the results has to be studied since the algorithm may reach a local optimum (local consistency) instead of the unique global optimum (global consistency) of the contrast function. The convergence depends on statistical variables such as random sampling of the data. It is commonly accepted that the estimation results are robust to the details of knowledge about the distributions (super- or sub-gaussianity, and so on). It has also been demonstrated that incorrect assumptions on such distributions can result in poor estimation performance, and sometimes in a complete failure to obtain the source separation [28]. Local consistency of ICA methods that search for specified distributions and global consistency in the case of two sources with heavy-tail distributions has been studied [19, 26, 29]. Recently, the statistical reliability or “quality” of the parameters estimated by ICA has been analyzed using bootstrap resampling techniques and visualization of the cluster structure of the components [30, 31].

2.2 Standard ICA Methods

The ideal measure of independence is the “mutual information” that was proposed as a contrast function in [17]. It has been demonstrated that this function corresponds to the likelihood for a model of independent components that is optimized with respect to all its parameters. Thus, the likelihood in a given ICA model is the probability of a data set as a function of the mixing matrix and the component distributions [28]. Mutual information (I) is defined as the Kullback–Leibler (KL) divergence or relative entropy between the joint density and the product of the marginal distributions:

$$I(\hat{\mathbf{s}}) = KL\left(\hat{\mathbf{s}}; \prod_i p(\hat{s}_i)\right) = \int p(\hat{\mathbf{s}}) \log \frac{p(\hat{\mathbf{s}})}{\prod_i p(\hat{s}_i)} d\hat{\mathbf{s}} \quad (2.5)$$

It is non-negative and equals to zero only if the distributions are the same. The logarithm of the fraction in Eq. (2.5) can be transformed into a difference of logarithms, obtaining

$$I(\hat{\mathbf{s}}) = \sum_i H(\hat{s}_i) - H(\hat{\mathbf{s}}) \quad (2.6)$$

where $H(u)$ denotes Shannon’s differential entropy for a continuous random variable u , which can be seen as a measure of the randomness of the variable u .

$$H(u) = - \int p(u) \log p(u) du \quad (2.7)$$

The entropy of the estimated sources $H(\hat{\mathbf{s}})$ in Eq. (2.4) equals $H(\mathbf{x}) - \log |\det \mathbf{B}|$. If a step of prewhitening is considered, all the white versions of $\hat{\mathbf{s}}$ are rotated versions of each other and $\log |\det \mathbf{B}| = 0$ since \mathbf{B} is an orthogonal matrix. For this case, the entropy $H(\hat{\mathbf{s}})$ remains constant and, thus, the mutual information (or dependence) is equal to the sum of the marginal entropies of $\hat{\mathbf{s}}$ (up to the constant term $H(\hat{\mathbf{s}})$)

$$I(\mathbf{s}) = \sum_i H(\hat{s}_i) \Leftrightarrow E[\hat{\mathbf{s}}\hat{\mathbf{s}}^T] = \mathbf{I} \quad (2.8)$$

Thus, there is a connection between maximum independence and minimum entropy; the objective of maximizing the independence is equivalent to the objective of minimizing the sum of the entropies of all components. In this sense, ICA is a minimum entropy method under the whitening constraint $E[\hat{\mathbf{s}}\hat{\mathbf{s}}^T] = \mathbf{I}$. In addition, the entropy $-H(u)$ is equal to the Kullback–Leibler divergence between the random variable u and the zero mean unit variance Gaussian density (up to a constant). Hence, the mutual information contrast imposes finding marginal distributions as far as possible from Gaussianity. Furthermore, it has been demonstrated that the mutual information can be decomposed under linear transforms as the sum of two contributions: a contribution expressing the decorrelation of the components and a contribution expressing their non Gaussianity [25].

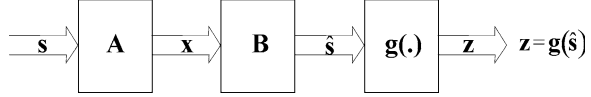
Unfortunately, the mutual information is difficult to approximate and optimize on the basis of a finite sample; thus, much research on ICA has focused on alternative solutions [17, 18, 20]. A popular approach for estimating the ICA model is the maximum likelihood (ML) estimation. The goal is to find the ICA parameters that give the highest probability for the observed data. The problem is formulated as $p_x(\mathbf{x}) = |\det \mathbf{B}| p_s(\mathbf{s}) = |\det \mathbf{B}| \prod_i p_i(s_i)$, where p_x is the density of the mixture vector, and p_i denotes the density of the independent components. Considering N samples available to evaluate the densities, the maximum log likelihood for the ICA model can be defined as

$$\frac{1}{N} \log L(\mathbf{B}) = E \left[\sum_i \log p_i(b_i^T \mathbf{x}) \right] + \log |\det \mathbf{B}| \quad (2.9)$$

where the expectation $E[\cdot]$ is the average computed from the observed samples.

Several methods to define the contrast function have been proposed in the literature. These methods are based on non Gaussianity, mutual information, higher order statistics (cumulants), and time structures [5]. The contrasts are closely connected, and have been implemented in different ICA algorithms for BSS with successful applications in many fields. The optimization techniques applied to the contrast function for adaptive algorithms are mainly based on gradient (natural, descent, etc.) and approximate Newton methods. The estimation

Fig. 2.2 InfoMax principle: mixing, unmixing, and nonlinear transformation



procedure of deterministic algorithms can exploit the algebraic structure of the matrices involved. The components are extracted using two methods. The first method consists of extracting sources source by source (deflation method), i.e., $\mathbf{s}_i = \mathbf{b}_i \mathbf{x}$; the second one consists of extracting all the sources simultaneously (symmetric method). The contrasts corresponding to these methods are called one-unit (one component) and multi-unit (several or all components) contrast functions.

We selected some of the most representative ICA algorithms (InfoMax [32, 23], JADE [33], FastIca [20, 22], and TDSEP [34]) derived from different perspectives of contrast design (entropy-, moment/cumulant-, and correlation-based methods). These algorithms will be used in comparisons with the techniques proposed in this work. A brief review of the selected ICA algorithms is included below.

2.2.1 InfoMax

The InfoMax algorithm was proposed in [32]. The InfoMax principle consists of maximizing the output entropy of a system $\mathbf{z} = \mathbf{g}(\hat{\mathbf{s}}) = \mathbf{g}(\mathbf{B}\mathbf{x})$ with respect to the demixing matrix \mathbf{B} , where \mathbf{g} is a nonlinear transformation (see Fig. 2.2).

The system shown in Fig. 2.2 can be considered as a neural network. The goal is to obtain the ICA parameters for an efficient flow of the information in the neural network. This requires maximizing the mutual information between the \mathbf{x} inputs and the $\hat{\mathbf{s}}$ outputs. It can be demonstrated that under no noise assumption, the maximization of this mutual information is equivalent to the maximization of the joint (output) entropy [35].

The transformation $\mathbf{g}(\cdot)$ is a $R^n \rightarrow R^n$ component-wise, non-linear function that operates on the sources estimated by the system linear part, i.e., $[\mathbf{g}(\hat{\mathbf{s}})]_i = g_i(\hat{s}_i)$ $1 \leq i \leq n$. Thus, the InfoMax contrast function is defined as

$$\phi_I(\mathbf{B}) = H(\mathbf{g}(\mathbf{B}\mathbf{x})) \quad (2.10)$$

where $H(\cdot)$ is the differential entropy. Scalar functions g_1, \dots, g_n are taken to be “squashing functions” that are capable of mapping a wide input domain to a narrow output domain (0, 1), and to be monotonously increasing. The entropy output entropy is estimated as [5]

$$H(\mathbf{g}(\mathbf{B}\mathbf{x})) = \sum_i E[\log g'_i(b_i^T \mathbf{x})] + \log |\det \mathbf{B}| \quad (2.11)$$

This expression can be matched with the expression of the likelihood in Eq. (2.9). If the nonlinearities g_i are chosen as the cumulative distribution functions corresponding to the densities p_i , i.e., $g'_i(\cdot) = p_i(\cdot)$, the output entropy is equal to the likelihood. Thus, InfoMax is equivalent to maximum likelihood estimation (see for instance [5, 36]).

The first implementation of InfoMax [32] employed a stochastic gradient algorithm. Afterwards, the algorithm convergence was accelerated using natural gradient [37]. InfoMax was extended in [23] (Extended InfoMax) for blind separation of mixed signals with sub- and super-gaussian source distributions. The optimization procedure uses stability analysis [38] to switch between sub- and super-gaussian regimes. The following is the algorithm learning rule

$$\Delta \mathbf{B} \propto (\mathbf{I} - E[\mathbf{g}(\hat{\mathbf{s}})\hat{\mathbf{s}}^T])\mathbf{B} \quad (2.12)$$

$g_i^+(\hat{\mathbf{s}}_i) = -2 \tanh(\hat{\mathbf{s}}_i)$ is usually used as component-wise nonlinearity for super-gaussian components and $g_i^-(\hat{\mathbf{s}}_i) = \tanh(\hat{\mathbf{s}}_i) - \hat{\mathbf{s}}_i$ for sub-gaussian components.

2.2.2 JADE

Joint Approximate Diagonalization of Eigen-matrices (JADE) is an algorithm that belongs to an approach derived from the theory of higher order cumulants [39]. This approach has been called higher-order cumulant tensor because its implementation is based on tensor algebra. The idea is to represent the fourth-order cumulant statistics of the data by a “quadricovariance tensor” and to compute its “eigenmatrices” to yield the desired components [40]. The tensor algebra enables the manipulation of the multidimensional higher-order cumulant matrices.

It can be shown that the second and third cumulants $cum(x_i, x_j)$ and $cum(x_i, x_j, x_k)$ are equal to the second and third moments $E[x_i, x_j]$ and $E[x_i, x_j, x_k]$. However, the fourth cumulant differs from the fourth moment of the random variables x_i, x_j, x_k , and x_l ; this is defined as

$$\begin{aligned} cum(x_i, x_j, x_k, x_l) &= C_{ijkl}(x_i, x_j, x_k, x_l) \\ &= E[x_i, x_j, x_k, x_l] - E[x_i, x_j]E[x_k, x_l] - E[x_i, x_k]E[x_j, x_l] - E[x_i, x_l]E[x_j, x_k] \end{aligned} \quad (2.13)$$

For independent variables $cum(x_i, x_j, x_k, x_l) = 0$. It means that $C_{ij}(\mathbf{s}) = \sigma_i^2 \delta_{ij}$, $C_{ijkl}(\mathbf{s}) = k_i \delta_{ijkl}$ with $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ for $i \neq j$, $\delta_{ijkl} = 1$ for $i = j = k = l$ and $\delta_{ijkl} = 0$ for $i \neq j \neq k \neq l$; where σ_i^2 is the variance, and k_i is the kurtosis of the source component s_i ($\sigma_i^2 = E[s_i^2]$, $K_i = E[s_i^4] - 3E^2[s_i^2]$) [5].

Thus, a measure of distance between the estimated and the source components can be stated as a distance between cumulants, obtaining the contrast under the whitening constraint

$$-\sum_i k_i C_{iiii}(\hat{\mathbf{s}}) = -E \left[\sum_i k_i (\hat{s}_i^4 - 3) \right] \quad (2.14)$$

If there is no prior knowledge about the sources in this case about the kurtosis, the contrast function is $-\sum_i k_i C_{iiii}^2(\hat{\mathbf{s}})$. This is equivalent to $\sum_{ijkl \neq iiii} C_{ijkl}^2(\hat{\mathbf{s}})$ since $E[\hat{\mathbf{s}}\hat{\mathbf{s}}] = \mathbf{I}$ [17] (up to a constant).

The JADE algorithm [33] approximates the independence by minimizing a smaller number of cross cumulants

$$\phi_{JADE} = \sum_{ijkl \neq ijjk} C_{ijkl}^2(\hat{\mathbf{s}}) \quad (2.15)$$

The optimization procedure of JADE tries to find the rotation matrix \mathbf{W} such that the cumulant matrices $\{\mathbf{Q}_i^z\}$ of the whitened data $\mathbf{z} = \mathbf{V}\mathbf{x}$ are as diagonal as possible. This solves

$$\arg \min \sum_i \text{off}(\mathbf{W}\mathbf{Q}_i^z\mathbf{W}^T) \quad (2.16)$$

where the operator $\text{off}(\mathbf{M}) = \sum_{i \neq j} \mathbf{M}_{ij}^2$ is the sum of the square of the off-diagonal

elements \mathbf{M} . This algorithm is based on the Jacobi method whose principle is that the rotation matrix \mathbf{Q} can be approximated by a sequence of elementary rotations $T_k(\phi_k)$ each of which try to minimize the off diagonal elements of the respective cumulant matrices. The rotation angle ϕ_k (Givens angles) can be calculated in closed form because fourth-order contrasts are polynomial in the parameters [41]. The rotation uses a small angle θ_{\min} , which controls the accuracy of the optimization. Thus, cumulant-based algebraic techniques avoid having to use gradient techniques for optimization. A comprehensive review about higher-order contrast used in ICA and comparison with gradient-based techniques is in [42].

2.2.3 FastIca

ICA methods have also been approached from the nongaussianity perspective. As stated above, without nongaussianity the estimation of the independent components is not possible. It is well-known from the central limit theorem that the distribution of a sum of independent random variables tends toward a Gaussian distribution, under certain conditions. The ICA estimation can be formulated as the search for directions that are maximally non-gaussian. Each local maximum gives one independent component [5]. In addition, the Gaussian variable has the maximum differential entropy (for unbounded variables with a common given variance). Thus, in order to find one independent component, we have to minimize entropy, i.e., we have to maximize the nongaussianity.

Two classical methods employed for measuring nongaussianity in ICA are kurtosis and negentropy. The kurtosis (fourth-order cumulant) of a random variable u is defined by $k(u) = E[u^4] - 3E^2[u^2]$. It is zero for Gaussian random variables and non zero for non Gaussian distributions. Random variables with negative kurtosis are called sub-gaussian or platykurtic, (e.g., the uniform random variable); and those with positive kurtosis are called super-gaussian or leptokurtic (e.g., the Laplacian random variable). Thus, functions such as $-\sum_i |k(\hat{s}_i)|$ and $-\sum_i |k^2(\hat{s}_i)|$ are appropriate contrasts. The gradient algorithm associated with the absolute value of the kurtosis is:

$$\Delta \mathbf{W} \propto \text{sign}(k(\mathbf{w}^T \mathbf{z})) E[\mathbf{z}(\mathbf{w}^T \mathbf{z})^3] \quad (2.17)$$

with the projection of \mathbf{W} on the unit sphere every step, i.e., it is normalized: $\frac{\mathbf{w}}{\|\mathbf{w}\|}$. This algorithm finds one component at a time, working with a whitened version of the mixed sources, $\mathbf{z} = \mathbf{V}\mathbf{x}$ by finding a column vector \mathbf{W} that maximizes the module of the kurtosis of $\hat{\mathbf{s}} = \mathbf{w}^T \mathbf{z}$.

The FastICA algorithm uses estimates of negentropy based on the maximum entropy principle, which requires the use of appropriate nonlinearities for the learning rule of the neural network [20, 22]. Separation is performed by the minimization of the negentropy of the mixture in order to obtain uncorrelated and independent sources whose amplitude distributions are as non Gaussian as possible. The non Gaussianity is measured with the differential entropy j , called negentropy [17], which is defined as the difference between the entropy of a Gaussian random variable u_{gauss} and the differential entropy of a random variable u , which are both variables of the same correlation (and covariance) matrix

$$J(u) = H(u_{\text{gauss}}) - H(u) \quad (2.18)$$

where the differential entropy H is defined by $H(u) = -\int f(u) \log f(u) du$. Since Gaussian random variables have the largest entropy H among all random variables having equal variance, maximizing $J(u)$ leads to the separation of independent source signals.

The use of negentropy has an advantage that is well justified by statistical theory. However, entropy estimation is computationally very difficult. Thus, several methods of approximation have been proposed [5]. One successful approximation consists of using a nonquadratic function G , which becomes

$$J(u) \propto [E\{G(u)\} - E\{G(v)\}]^2 \quad (2.19)$$

For optimization, the following algorithm can be obtained

$$\Delta \mathbf{w} \propto \gamma E\{\mathbf{z}g(\mathbf{w}^T \mathbf{z})\} \quad (2.20)$$

with the projection of \mathbf{w} on the unit sphere every step and where $\gamma = E\{G(\mathbf{w}^T \mathbf{z})\} - E\{G(v)\}$ and v is a standardized Gaussian random variable. The normalization is necessary to project \mathbf{w} to keep the variance of $\mathbf{w}^T \mathbf{z}$ constant. The non-linearity $g(\cdot)$ is the derivative of the function G used in the approximation. It can be chosen from $g_1(\hat{s}) = \tanh(a_1 \hat{s})$ where $1 \leq a_1 \leq 2$, $g_2(\hat{s}) = \hat{s} \exp(-\hat{s}^2/2)$, or $g_3(\hat{s}) = \hat{s}^3$ [20, 22].

2.2.4 TDSEP

Temporal decorrelation source separation (TDSEP) is one of the ICA algorithms that exploit the time structure of the signals. It is based on the simultaneous diagonalization of several time-delayed correlation matrices. The approach relies on second-order statistics by assuming distinctive spectral/temporal characteristics of the sources [34, 43, 44]. These algorithms have been successfully applied in biosignal processing given the inherent time structure of the signals and their capability to separate signals whose amplitude distribution is near Gaussian.

The TDSEP algorithm uses the property that the cross-correlation functions vanish for mutually independent signals. It assumes that the signals $\mathbf{s}(t)$ have temporal structure (“non delta” autocorrelation function). All time delayed correlation matrices $\mathbf{R}_{\tau(\mathbf{s})}$ should be diagonal. This knowledge is used to calculate the unknown mixing matrix \mathbf{A} by a simultaneous diagonalization of a set of correlated matrices $\mathbf{R}_{\tau(\mathbf{x})} = \langle \mathbf{x}(t) \mathbf{x}(t - \tau)^T \rangle$ for different choices of τ , where τ , is a lag constant, $\tau = 1, 2, 3, \dots$. The diagonal elements of these matrices are formed by the values of the autocorrelation functions and the off-diagonal elements are the respective cross correlations,

$$\mathbf{R}_{\tau(\mathbf{x})} = \begin{bmatrix} \varphi_{x_1, x_1}(\tau) & \dots & \varphi_{x_1, x_n}(\tau) \\ \varphi_{x_1, x_2}(\tau) & \dots & \varphi_{x_2, x_n}(\tau) \\ \vdots & & \ddots \\ \varphi_{x_n, x_1}(\tau) & \dots & \varphi_{x_n, x_n}(\tau) \end{bmatrix} \quad (2.21)$$

where φ denotes the correlation function. If the signals were independent over time, all time-delayed correlation matrices should be diagonal because the cross-correlations of independent signals vanish.

The contrast consists of finding a matrix \mathbf{B} (considering whitening) so that in addition to making the instantaneous covariances of $\hat{\mathbf{s}}(t) = \mathbf{B}\mathbf{x}(t)$ go to zero, the lagged covariances are made zero as well:

$$E[\hat{s}_i(t) \hat{s}_j(t - \tau)] = 0, \quad \text{for all } i, j, \tau \quad \text{with } i \neq j \quad (2.22)$$

For the independent components $S_i(t)$, the lagged covariances are all zero due to independence, without the need for higher-order information to estimate the model.

The optimization procedure has to minimize the sum of the off-diagonal elements (diagonalize) of several lagged covariances of $\hat{\mathbf{s}} = \mathbf{w}\mathbf{z}$. Considering the symmetric version $\bar{\mathbf{C}}_{\tau(\mathbf{z})} = \frac{1}{2} [\mathbf{C}_{\tau(\mathbf{z})} + (\mathbf{C}_{\tau(\mathbf{z})})^T]$ of the covariance matrix and a set of chosen lags τ denoted by \mathbf{s} , the objective function can be written as

$$\sum_{\tau \in \mathbf{s}} \text{off}(\mathbf{W}\bar{\mathbf{C}}_{\tau(\mathbf{z})}\mathbf{W}^T) \quad (2.23)$$

The minimization of Eq. (2.23) can be accomplished by a gradient descent algorithm. Another alternative is to adapt the existing methods for eigenvalue decomposition for this simultaneous approximate diagonalization of several matrices. The SOBI algorithm (Second-Order Blind Identification) and TDSEP use Jacobi-like algorithms for optimization [43, 44].

The set of time delays τ can be arbitrarily selected or manually given with prior knowledge. The advantage of second-order methods is their computational simplicity and efficiency. Furthermore, for a reliable estimate of covariances only comparatively few samples are needed.

2.3 Non-Parametric ICA

The estimation of the densities is, in general, a non-parametric problem. This means that the number of parameters is infinite, or, in practice, very large. The non-parametric problems are the most difficult to estimate. As was reviewed in Sect. 2.1, most known methods for solving the ICA problem involve specification of the parametric form of the latent components densities p_i and estimation of \mathbf{B} together with parameters of p_i using maximum likelihood or minimization of the empirical versions of various divergence criteria between densities. In practical applications, the distributions p_i of the independent components are generally unknown, and thus ICA can be considered as a semi-parametric method in which these distributions are left unspecified.

Conventional ICA techniques have used two methods to avoid non-parametric estimation. The first method consists of using prior available knowledge about the densities. The results of the estimator would depend on the specification of the priors. By including these priors in the likelihood, the likelihood would really be a function of \mathbf{B} only. A second method is to approximate the densities of the independent components by a family of densities that are specified by a limited number of parameters. For instance, a simple parameterization of the p_i is a single binary parameters, i.e., the choice between two densities [5].

Nowadays, there seem to be two research directions in ICA modelling: the first is motivated to design a signal separation algorithm that is “truly blind” to the

particular underlying distributions of the mixed signals (any information about the sources is completely unknown), (see for instance [45]); the second consists of including the maximum number of priors available in the cost function in order to guide the algorithm to find particular sources (blind source extraction, semi-blind source separation, etc.), (see for instance [46, 47]). Some new methods that use non-parametric (NP) density estimation have been recently developed from the first direction in ICA research.

The new non-parametric ICA methods use techniques such as: minimization of a kernel canonical correlation or a kernel generalized variance among recovered sources (the so-called Kernel-ICA) [48]; maximum likelihood estimation (MLE) by using spline-based density approximations [49]; MLE by using Gaussian kernel density estimates (the so-called Npica) [45]; and minimization of the entropy of the marginals by estimating their order statistics (the so-called Radical) [50]. These methods have shown good performance in simulations, but there are no references about their performance in real applications. Theoretical analyses (convergence, consistency, and other issues) of non-parametric density estimation in the framework of ICA are found in [29, 26, 51]. We include a review of the Npica, Radical, and Kernel-ICA algorithms in the following section.

2.3.1 Npica

The Npica algorithm [45] is a maximum loglikelihood ICA method that solves the Eq. (2.9). It uses a non-parametric estimation for the probability density function p_i , which is directly estimated from the data using a kernel density estimation technique [52].

Given a batch of sample data of size N , the marginal distribution of an arbitrary reconstructed signal is approximated as follows:

$$p_i(\hat{s}_i) = \frac{1}{Nh} \sum_{l=1}^N \kappa\left(\frac{\hat{s}_i - \hat{s}_{il}}{h}\right), \quad i = 1, \dots, M \quad (2.24)$$

where h is the kernel bandwidth and κ is the Gaussian kernel $\kappa(u) \triangleq \frac{1}{\sqrt{2\pi}} e^{-u^2/2}$.

The kernel centroids \hat{s}_{il} are equal to $\hat{s}_{il} = \mathbf{w}_i \mathbf{x}^{(l)} = \sum_{l=1}^N w_{il} X_{li}$, where $\mathbf{x}^{(l)}$ is the l th column of the mixture matrix \mathbf{X} .

The expectation of the maximum loglikelihood solution is approximated by the following cost function

$$L(\mathbf{W}) = -L_0(\mathbf{W}) - \log(\det \mathbf{W}) \quad (2.25)$$

where $L_0(\mathbf{W})$ is obtained by replacing the marginal pdf's p_i with their kernel density estimates

$$\begin{aligned}
L_0(\mathbf{W}) &= \sum_{i=1}^M E \log \left[\frac{1}{Nh} \sum_{l=1}^M \kappa \left(\frac{\hat{s}_i - \hat{s}_{il}}{h} \right) \right] \\
&\approx \frac{1}{N} \sum_{i=1}^M \sum_{k=1}^M \log \left[\frac{1}{Nh} \sum_{l=1}^M \kappa \left(\frac{\mathbf{w}_i(\mathbf{x}^{(k)} - \mathbf{x}^{(l)})}{h} \right) \right]
\end{aligned} \tag{2.26}$$

The overall optimization problem can thus be posed as

$$\min_{\mathbf{w}} -\frac{1}{N} \sum_{i=1}^M \sum_{k=1}^M \log \left[\frac{1}{Nh} \sum_{l=1}^M \kappa \left(\frac{\mathbf{w}_i(\mathbf{x}^{(k)} - \mathbf{x}^{(l)})}{h} \right) \right] - \log |\det \mathbf{W}| \tag{2.27}$$

$$\text{s.t.} \|\mathbf{w}_i\| = 1, \quad i = 1, \dots, M \tag{2.28}$$

Given the sample data $\mathbf{x}^{(k)}$, $k = 1, \dots, N$, the objective of Eq. (2.27) is a smooth nonlinear function of the elements of the matrix \mathbf{W} . The additional constraints of Eq. (2.28) restrict the space of possible solutions of the problem to a finite set. The optimization technique applied is the quasi-Newton method.

2.3.2 Radical

The Radical algorithm [50] uses entropy minimization, i.e., it must estimate the entropy of each marginal for each possible \mathbf{W} matrix. The Radical marginal entropy estimates are functions of the order statistics of those marginals.

The order statistics are estimated using *spacings* estimates of entropy. Consider a one-dimensional random variable Z , and a random sample of Z denoted by Z^1, Z^2, \dots, Z^N . The order statistics of a random sample of Z are simply the elements of the sample rearranged in non-decreasing order: $Z^{(1)} \leq Z^{(2)} \leq \dots \leq Z^{(N)}$. A spacing of order m , or m -*spacings* is then defined to be $Z^{(i+m)} - Z^{(i)}$, for $1 \leq i \leq i+m \leq N$. Finally, if m is a function of N , a m_N -*spacings* such as $Z^{(i+m)} - Z^{(i)}$, can be defined.

For any random variable Z with an impulse-free density $p(\cdot)$ and continuous distribution function $p(\mathbf{x}/C_k) = |\det \mathbf{A}_k^{-1}| p(\mathbf{s}_k)$, the following holds. Let p^* be the Z -way product density $p^*(Z^1, Z^2, \dots, Z^N) = p(Z^1)p(Z^2) \dots p(Z^N)$. Then

$$E_{p^*} \left[P \left(Z^{(i+1)} \right) - P \left(Z^{(i)} \right) \right] = \frac{1}{N+1}, \quad \forall i, 1 \leq i \leq N-1 \tag{2.29}$$

Using these ideas, the following simple entropy estimator can be derived.

$$\hat{H}_{m\text{-spacings}}(Z^1, \dots, Z^N) \equiv \frac{m}{N-1} \sum_{i=0}^{\frac{N-1}{m}-1} \log \left(\frac{N+1}{m} \left(Z^{(m(i+1)+1)} - Z^{(mi+1)} \right) \right) \quad (2.30)$$

Under the condition that $m, N \rightarrow \infty$, $\frac{m}{N} \rightarrow 0$, this estimator is consistent; typically $m = \sqrt{N}$. The intuition behind this estimator is that by considering m -spacings with larger and larger values of m , the variance of the probability mass of these spacings relative to their expected values gets smaller and smaller. In fact, the probability mass of m -spacings is distributed according to a beta distribution with parameters m and $N+1$ [50]. Thus, a modification of Eq. (2.30) in which the m -spacings overlap is used in Radical. The final contrast consists of an entropy estimator that is used to minimize Eq. (2.8),

$$\hat{H}_{\text{Radical}}(Z^1, \dots, Z^N) \equiv \frac{1}{N-m} \sum_{i=1}^{N-m} \log \left(\frac{N+1}{m} \left(Z^{(i+m)} - Z^{(i)} \right) \right) \quad (2.31)$$

The optimization method of the algorithm for cost function minimization is exhaustive search. It is assumed that the data are first pre-whitened and augmented with a number of synthetic replicates of each of the original N sample points with additive spherical Gaussian noise to make a surrogate data set. This is done in order to obtain a smoother version of the estimator in an attempt to remove false minima. Afterwards, for each angle θ , the data are rotated ($\hat{\mathbf{s}} = \mathbf{W}(\theta) \cdot \mathbf{x}$) using a pair-wise Jacobi rotation and the cost function evaluated. The output is the \mathbf{W} corresponding to the optimal θ . There are $M(M-1)/2$ distinct Jacobi rotations parameterized by θ (for a M -dimensional ICA). Optimizing over a set of these rotations is known as a sweep. Empirically, performing multiple sweeps improves the estimate of \mathbf{W} for some number of iterations. In [50], good results were reported in simulations for $S \approx M$ (S is the number of sweeps).

2.3.3 Kernel-ICA

The Kernel-ICA algorithm [48] uses contrast functions based on canonical correlations in a reproducing kernel Hilbert space. This approach is not based on a single nonlinear function, but rather on an entire function space of candidate nonlinearities. The contrast function is a rather direct measure of the dependence of a set of random variables. Considering the case of two univariate random variables x_1 and x_2 , and letting F be a vector space of functions from \mathbb{R} to \mathbb{R} , the F -correlation ρ_F is defined as the maximal correlation between the random variables $f_1(x_1)$ and $f_2(x_2)$, where f_1 and f_2 range over F :

$$\rho_F = \max_{f_1, f_2 \in F} \text{corr}(f_1(x_1), f_2(x_2)) = \max_{f_1, f_2 \in F} \frac{\text{cov}(f_1(x_1), f_2(x_2))}{(\text{var}f_1(x_1))^{1/2} (\text{var}f_2(x_2))^{1/2}} \quad (2.32)$$

Clearly, if the variables x_1 and x_2 are independent, then the F - *correlation* is equal to zero. It can be shown that F - *correlation* is the maximal possible correlation between one-dimensional linear projections $\Phi(x_1)$ and $\Phi(x_2)$, with $\Phi(x) = K(\cdot, x)$ being the feature map, where the kernel $K(\cdot, x)$ is a function in F for each x . This is the definition of the first “canonical correlation” between $\Phi(x_1)$ and $\Phi(x_2)$.

Canonical correlation analysis (CCA) is a multivariate statistical technique similar to PCA. While PCA works with a single random vector and maximizes the variance of projections of the data, CCA works with a pair of random vectors (or in general with a set of m random vectors) and maximizes correlation between sets of projections. While PCA leads to an eigenvector problem, CCA leads to a generalized eigenvector problem.

Kernel-ICA employs a “kernelized” version of CCA to compute a flexible contrast function for ICA. The following definitions are considered. Let $\{x_1^1, \dots, x_1^N\}$ and $\{x_2^1, \dots, x_2^N\}$ denote sets of N observations of x_1 and x_2 , respectively, and let $\{\Phi(x_1^1), \dots, \Phi(x_1^N)\}$ and $\{\Phi(x_2^1), \dots, \Phi(x_2^N)\}$ denote the corresponding images in feature space. Let S_1 and S_2 represent the linear spaces spanned by the α_i -images of the data points. Thus, $f_1 = \sum_{k=1}^N \alpha_1^k \Phi(x_1^k) + f_1^\perp$ and $f_2 = \sum_{k=1}^N \alpha_2^k \Phi(x_2^k) + f_2^\perp$, where f_1^\perp and f_2^\perp are orthogonal to S_1 and S_2 , respectively.

Considering that K_1 and K_2 are the Gram matrices associated with the data sets $\{x_1^i\}$ and $\{x_2^i\}$, respectively, the following variance estimates are obtained: $\hat{\text{var}}(\langle \Phi(x_1), f_1 \rangle) = \frac{1}{N} \alpha_1^T K_1 K_1 \alpha_1$ and $\hat{\text{var}}(\langle \Phi(x_2), f_2 \rangle) = \frac{1}{N} \alpha_2^T K_2 K_2 \alpha_2$. Thus, the kernelized CCA problem for two variables becomes that of performing the following maximization:

$$\rho_F(K_1, K_2) = \max_{\alpha_1, \alpha_2 \in \mathbb{R}^N} \frac{\alpha_1^T K_1 K_2 \alpha_2}{(\alpha_1^T K_1^2 \alpha_1)^{1/2} (\alpha_2^T K_2^2 \alpha_2)^{1/2}} \quad (2.33)$$

The formulation as a generalized and regularized eigenvalue problem to m variables is the following:

$$\begin{aligned} & \begin{pmatrix} (K_1 + \frac{N_k}{2} I)^2 & K_1 K_2 & \dots & K_1 K_m \\ K_2 K_1 & (K_2 + \frac{N_k}{2} I)^2 & \dots & K_2 K_m \\ \vdots & \vdots & \ddots & \vdots \\ K_m K_1 & K_m K_2 & \dots & (K_m + \frac{N_k}{2} I)^2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix} \\ &= \lambda \begin{pmatrix} (K_1 + \frac{N_k}{2} I)^2 & 0 & \dots & 0 \\ 0 & (K_1 + \frac{N_k}{2} I)^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & (K_1 + \frac{N_k}{2} I)^2 \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{pmatrix} \end{aligned} \quad (2.34)$$

where N_k is a small positive constant used for regularization. The minimal value of this problem is called the first kernel canonical correlation.

The Kernel-ICA algorithm proceeds as follows. Given a set of data vectors $\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N$, and given a parameter matrix \mathbf{W} , we set $\hat{\mathbf{s}}^i = \mathbf{W}\mathbf{x}^i$, for each i , and thereby form a set of estimated source vectors $\hat{\mathbf{s}}^1, \hat{\mathbf{s}}^2, \dots, \hat{\mathbf{s}}^N$. The m components of these vectors yield a set of m Gram matrices, K_1, K_2, \dots, K_m , and these Gram matrices (which depend on \mathbf{W}) define the contrast function $C(\mathbf{W}) = I\lambda F(K_1, \dots, K_m)$. The ICA algorithm minimizes this function with respect to \mathbf{W} .

The optimization technique used for Kernel-ICA is gradient descent (with line search) on an almost-everywhere differentiable function $C(\mathbf{W})$. The algorithm converges to a local minimum of $C(\mathbf{W})$ for any starting point. However, the ICA contrast functions have multiple local minima, and restarts are generally necessary if we are to find the global optimum. Empirically, the number of restarts was found to be small when the number of samples was sufficiently large so as to make the problem well-defined [48].

2.4 ICA Mixture Modelling

ICAMM is proposed in the framework of pattern recognition, considering that the observed data come from a mixture model and they can be categorized into several mutually exclusive classes. ICAMM assumes the underlying process that generated observed data is composed by multiple ICA models (data of each class are modelled as an ICA, i.e., linear combinations of independent non-gaussian sources). This modelling has been proposed in order to deal with the problems of the widely used mixture of Gaussians (MoG)-based modelling [53]. The principal limitations of MoG are: (i) the size (M^2) of each covariance matrix becomes extremely large when the dimension (M) of the problem increases; and (ii) each component is a Gaussian, which is a condition that is rarely found in real data sets. The antecedents of ICAMM can be found in [54] where each Gaussian of the mixture was replaced with a probabilistic principal component analysis (PPCA), allowing the covariance matrix dimension to be reduced, preserving the representation of the data. This PCA-based method was modified in [55] using variational Bayesian inference to infer the optimum number of analysers, obtaining the so-called Mixture of Factor Analysers. Afterwards, a robust approach for PPCA that exploits the adaptive distribution tails of the Student- t was proposed [56, 57]. This last allows the performance of the method is not spoiled by non-gaussian noise (e.g., outliers). Thus, ICA mixture modelling has been the natural evolution from these antecedents.

ICAMM was introduced in [58] considering a source model switching between Laplacian and bimodal densities. Afterwards, the model was extended using generalized exponential sources [59], self-similar areas such as mixtures of Gaussians sub-features using variational Bayesian inference [53], and sources with

non-gaussian structures recovered by a learning algorithm using Beta divergence [56]. In addition, the automatic estimation of the number of ICA mixtures has been approached by variational Bayesian learning [60, 61] and on-line adaptive estimation of the clusters comparing log-likelihood of the data [62]. An alternative to the simultaneous estimation of all the ICAMM parameters is the performing of segmented and repeated ICAs. This strategy has been recently applied for the extraction of neural activity from large-scale optical recordings [63]. Ultimately, computational optimization of gradient techniques used in ICAMM algorithm was proposed applying Newton's method in the [64, 60].

The general formulation of ICAMM is:

$$\mathbf{x}_t = \mathbf{A}_k \mathbf{s}_k + \mathbf{b}_k, \quad k = 1, \dots, K \quad (2.35)$$

where C_k denotes the class k , and each class is described by an ICA model with a mixing matrix \mathbf{A}_k , and a bias vector \mathbf{b}_k . Essentially, \mathbf{b}_k determines the location of the cluster and $\mathbf{A}_k \mathbf{s}_k$ its shape. The goal of an ICA mixture model algorithm is to determine the parameters for each class. Figure 2.3 shows the model of ICA mixtures.

There are a few methods proposed in the ICAMM framework. They can be grouped as follows: maximum-likelihood based, iterative-based on a distance measure, and variational Bayesian learning methods. We include a review of three representative ICAMM techniques: the first proposed method for unsupervised classification and automatic context switching [58], the Beta-divergence method [65], and a variational Bayesian method [53].

2.4.1 Unsupervised Classification Using ICAMM

In [58], an unsupervised classification maximum-likelihood-based algorithm for modelling classes with non-gaussian densities (ICA structures) is proposed.

The likelihood of the data is given by the joint density $p(\mathbf{X}|\Theta) = \prod_{i=1}^T p(\mathbf{x}_i|\Theta)$, with t being the data index $t = 1, \dots, T$. The mixture density is $p(\mathbf{x}_t|\Theta) = \prod_{k=1}^K p(\mathbf{x}_t|C_k, \theta_k) p(C_k)$, where $\Theta = (\theta_1, \dots, \theta_K)$ are the unknown parameters for each of the component densities $p(\mathbf{x}|C_k, \theta_k)$, and C_k denotes the class k , $k = 1, \dots, K$. The data within each class k are described by Eq. (2.35).

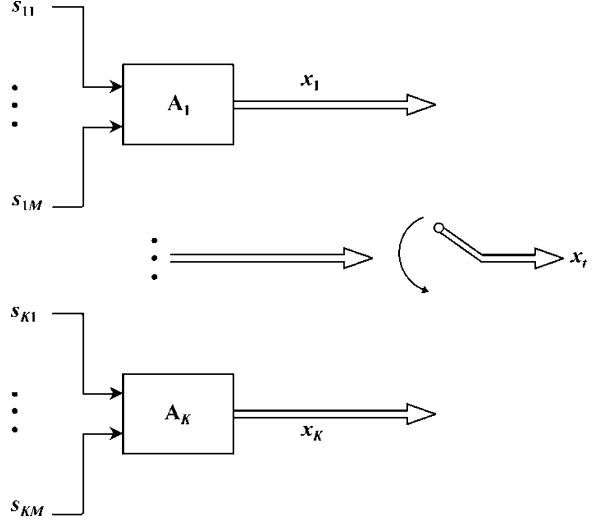
The log-likelihood of the data for each class is defined as

$$\log p(\mathbf{x}_t|C_k, \theta_k) = \log p(\mathbf{s}_k) - \log(\det |\mathbf{A}_k|) \quad (2.36)$$

and the probability for each class given the data vector \mathbf{x}_t is:

$$p(C_k|\mathbf{x}_t, \Theta) = \frac{p(\mathbf{x}_t|\theta_k, C_k)p(C_k)}{\sum_{k=1}^K p(\mathbf{x}_t|\theta_k, C_k)p(C_k)}.$$

Fig. 2.3 Outline of the ICA mixture model



The Extended InfoMax algorithm [23] is used for adapting the basis functions (mixture matrix) in the ICA model. The gradient ascent technique is used to maximize the log-likelihood function. The rules to update the basis functions \mathbf{A}_k and the bias vectors \mathbf{b}_k for every class are the following

$$\Delta \mathbf{A}_k \propto -p(C_k|\mathbf{x}_t, \Theta) \mathbf{A}_k [\mathbf{I} - \mathbf{K} \tanh(\mathbf{s}_k) \mathbf{s}_k^T - \mathbf{s}_k \mathbf{s}_k^T] \quad (2.37)$$

$$\mathbf{b}_k = \frac{\sum_{t=1}^T \mathbf{x}_t p(C_k|\mathbf{x}_t, \Theta)}{\sum_{k=1}^K p(C_k|\mathbf{x}_t, \Theta)} \quad (2.38)$$

For the automatic switching between super-gaussian and sub-gaussian source distributions, a switching matrix $O_{k,l}$ is used. Super-Gaussian ($O_{k,l} = 1$) : $\log p(\mathbf{s}_k) \propto -\sum_{l=1}^n |s_{k,l}|$, and Sub-Gaussian ($O_{k,l} = -1$) : $\log p(\mathbf{s}_k) \propto -\sum_{l=1}^n (\log(\cosh(s_{k,l})) - \frac{s_{k,l}^2}{2})$. where n is the dimensions of the source, $s_{k,l}$ is the l th dimension of the source in the k th class, and $O_{k,l}$ is an index which allows for automatic switching between super-gaussian and sub-gaussian models [23] $O_{k,l} = \text{sign} \left[E \{ \text{sech}^2(s_{k,l}) \} E \{ s_{k,l}^2 \} - E \{ (\tanh(s_{k,l})) s_{k,l} \} \right]$.

The algorithm was tested to automatically identify different contexts in BSS (each context featured by the parameters of an ICA model), assuming the number of classes K to be known. An extension was made in [61] where the number of clusters and the intrinsic dimension of each cluster were determined by a variational Bayesian method similar to the method proposed in [59]. Recently, an on-line version for partitioning the input-output space for fuzzy neural networks was proposed in [62]. In this algorithm, one cluster is generated for the first data vector. For new data, a decision is made to generate or not generate new clusters

depending on the degree to which the new incoming pattern \mathbf{x}_t belongs to the j th cluster, which is defined as $F^j(\mathbf{x}_t) = \log p(\mathbf{x}_t|C_j)$. The maximum log-likelihood value ($F^{J_{\max}}(\mathbf{x}_t)$) among all log-likelihood values estimated for the existing J clusters at time t is selected. If $F^{J_{\max}}(\mathbf{x}_t) \geq F$, the corresponding new incoming pattern is added to the existing cluster with index J_{\max} , and the parameters of this cluster are updated properly (F is a given negative threshold value obtained empirically). In this case, no new cluster is generated. If $F^{J_{\max}}(\mathbf{x}_t) < F$, a new cluster is generated to accommodate this new pattern.

2.4.2 β -Divergence Method Applied to ICAMM

This algorithm is based on the minimum β -divergence distance [56, 65]. The β -divergence between two probability density functions $p(\mathbf{x})$ and $q(\mathbf{x})$ is defined as

$$D_\beta(p, q) = \int \left[\frac{1}{\beta} \{p^\beta(\mathbf{x}) - q^\beta(\mathbf{x})\} p(\mathbf{x}) - \frac{1}{\beta+1} \{p^{\beta+1}(\mathbf{x}) - q^{\beta+1}(\mathbf{x})\} \right] d\mathbf{x}, \quad \text{for } \beta > 0 \quad (2.39)$$

which is non-negative and equal to zero if and only if $p(\mathbf{x}) = q(\mathbf{x})$. The β -divergence reduces to Kullback–Leibler divergence when $\beta \rightarrow 0$.

There exists a matrix \mathbf{W} and a shifting parameter vector $\boldsymbol{\mu}$ such that the components of $\hat{\mathbf{s}} = \mathbf{W}\mathbf{x} - \boldsymbol{\mu}$. Thus, the joint density of $\hat{\mathbf{s}}$ can be expressed as the product of marginal density functions q_1, \dots, q_m by $q(\hat{\mathbf{s}}) = \prod_{i=1}^m q_i(\hat{s}_i)$, and the joint

density function of \mathbf{x} can be expressed as $r(\mathbf{x}, \mathbf{W}, \boldsymbol{\mu}) = |\det(\mathbf{W})| \prod_{i=1}^m q_i(\mathbf{w}_i\mathbf{x} - \mu_i)$, where \mathbf{W}_i is the i th row vector of \mathbf{W} , and μ_i is the i th component of $\boldsymbol{\mu}$.

The algorithm explores the recovering matrix of each class in the ICA mixture on the basis of the initial condition of a shifting parameter $\boldsymbol{\mu}$. If the initial value of the shifting parameter is close to the mean of the k th class, then the estimates for the recovering matrix \mathbf{W}_k and the shifting parameter $\boldsymbol{\mu}_k$ can be obtained for this class by considering the data in other classes as outliers. Thus, $\{(\mathbf{W}_k, \boldsymbol{\mu}_k); k = 1, \dots, c\}$ can be estimated by the repeated application of the β -divergence method to recover all hidden classes that are sequentially based on a rule for the step-by-step change of the shifting parameter $\boldsymbol{\mu}$. In order to create a rule for the sequential change of $\boldsymbol{\mu}$, the weight function ϕ is defined

$$\phi(\mathbf{x}, \mathbf{W}, \boldsymbol{\mu}) = \prod_{i=1}^m p_i^\beta(\mathbf{w}_i\mathbf{x} - \mu_i) \quad (2.40)$$

The minimum β -divergence method finds the minimizer of the empirical β -divergence $\widehat{D}_\beta(\tilde{r}, r_0(\cdot, \mathbf{W}, \boldsymbol{\mu}))$, where \tilde{r} is the empirical distribution of \mathbf{x} , and r_0

corresponds to a nonlinearity with density p_i (e.g., $p_i(z) = c_2 / \cosh(z)$ for super-gaussian signals) that allows switching between sub-gaussian and super-gaussian densities as in the Extended InfoMax algorithm [23]. This minimization is equivalent to maximizing the following quasi β -likelihood function:

$$L_\beta(\mathbf{w}, \boldsymbol{\mu}) = \frac{1}{n} \sum_{t=1}^n l_\beta(\mathbf{x}_t; \mathbf{W}, \boldsymbol{\mu}) \quad (2.41)$$

where
$$l_\beta(\mathbf{x}; \mathbf{w}, \boldsymbol{\mu}) = \begin{cases} \log(r_0(\mathbf{x}, \mathbf{w}, \boldsymbol{\mu})), & \text{for } \beta = 0 \\ \frac{1}{\beta} r_0^\beta(\mathbf{x}, \mathbf{w}, \boldsymbol{\mu}) - b_\beta(\mathbf{w}) - \frac{1-\beta}{\beta}, & \text{for } 0 < \beta < 1 \end{cases}, \quad \text{and}$$

$$b_\beta(\mathbf{w}) = \frac{1}{\beta+1} \int r_0^{\beta+1}(\mathbf{x}, \mathbf{w}, \boldsymbol{\mu}) d\mathbf{x} = \frac{|\det(\mathbf{w})|^\beta}{\beta+1} \int \prod_{i=1}^m p_i^{\beta+1}(z_i) dz$$

2.4.3 Variational Mixture of Bayesian ICAs

Bayesian inference and variational learning were introduced in the estimation of the ICAMM parameters in [53]. Mixture of Gaussians was used as source model. The generative model for a data vector \mathbf{x} in this approach is shown in Fig. 2.4.

The probability of generating a data vector \mathbf{x}^n from a C -component mixture model given assumptions \mathcal{M} is:

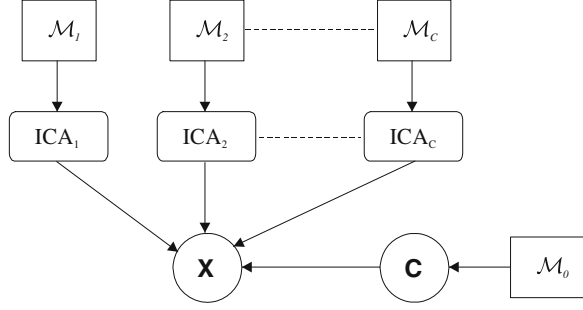
$$p(\mathbf{x}^n | \mathcal{M}) = \sum_{c=1}^C p(c | \mathcal{M})_0 p(\mathbf{x}^n | \mathcal{M}_c, c) \quad (2.42)$$

A data vector is generated by choosing one of the C components stochastically under $p(c | \mathcal{M})_0$ and then drawing from $p(\mathbf{x}^n | \mathcal{M}_c, c)$; where $\mathcal{M} = \{\mathcal{M}_0, \mathcal{M}_1, \dots, \mathcal{M}_C\}$ is the vector of component model assumptions, \mathcal{M}_c , and assumptions about the mixture process, \mathcal{M}_0 . The assumptions represent everything that essentially defines the model (values of fixed parameters, model structure, details of the component switching method, any prior information, etc.).

The probability of observing data vector \mathbf{x}^n under component c th ICA model ($\mathbf{x} = \mathbf{A}_c \mathbf{s}_c + \mathbf{y}_c + \mathbf{e}_c$, \mathbf{s}_c are the sources of dimension L_c , \mathbf{y}_c is an S -dimensional bias vector, and \mathbf{x} is S -dimensional additive noise) is given by

$$p(\mathbf{x}^n | \theta_c, c) = \left(\frac{\lambda_c}{2\pi} \right)^{\frac{S}{2}} \exp[-E_c] \quad (2.43)$$

where $\theta_c = \{\mathbf{A}_c, \mathbf{s}_c^n \lambda_c\}$, $E_c = \frac{\lambda_c}{2} (\mathbf{x}_n - \mathbf{A}_c \mathbf{s}_c^n - \mathbf{y}_c)^T (\mathbf{x}_n - \mathbf{A}_c \mathbf{s}_c^n - \mathbf{y}_c)$, and λ_c is related with the variance of the noise considered zero-mean Gaussian and isotropic.

Fig. 2.4 ICA mixture for variational learning

The source model is MoG, which is a factorized mixture of 1-dimensional Gaussians with L_c factors (i.e., sources) and L_c components per source. This model is defined as (subscript c has been dropped for brevity),

$$\begin{aligned}
 p(\mathbf{s}_c^n | \varphi_c, c) &= \prod_{i=1}^{L_c} \sum_{q_i=1}^{m_i} p(q_i^n = q_i | \pi_i, c) p(\mathbf{s}_{c,i}^n | \varphi_{c,i}, c) \\
 &= \prod_{i=1}^{L_c} \sum_{q_i=1}^{m_i} \pi_{i,q_i} \mathcal{N}(\mathbf{s}_{c,i}^n; \mu_{i,q_i}, \beta_{i,q_i})
 \end{aligned} \tag{2.44}$$

where μ_{i,q_i} is the position of feature q_i w.r.t. the cluster centre, β_{i,q_i} is its size, and π_{i,q_i} its “prominence” w.r.t. other features. The mixture proportions $\pi_{i,q_i} = p(q_i^n = q_i | \pi_i)$ are the prior probabilities of choosing component q_i of the i th source (of the c th ICA model etc.). q_i^n is a variable indicating which component of the i th source is chosen for generating $s_{c,i}^n$ and takes on values of $\{q_i = 1, \dots, q_i = m_i\}$ (where m_i depends on ICA model c). The parameters of source i are $\varphi_{c,i} = \{\pi_{c,i}, \mu_{c,i}, \beta_{c,i}\}$. The complete parameter set of the source model is $\varphi_c = \{\varphi_{c,1}, \varphi_{c,2}, \dots, \varphi_{c,L_c}\}$. The complete collection of possible source states is denoted as $\mathbf{q}_c = \{\mathbf{q}_{c,1}, \mathbf{q}_{c,2}, \dots, \mathbf{q}_{c,\mathbf{m}}\}$ and runs over all $\mathbf{m} = \prod im_i$ possible combinations of source states.

It can be shown that the likelihood of the i.i.d. data $\mathbf{X} = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N\}$ given the model parameters $\Theta_c = \{\mathbf{A}_c, \mathbf{y}_c, \lambda_c, \varphi_c\}$ can be written as

$$p(\mathbf{X} | \Theta_c, c) = \prod_{n=1}^N \sum_{\mathbf{q}=1}^{\mathbf{m}} \int p(\mathbf{x}^n, \mathbf{s}_c^n, \mathbf{q}_c^n | \Theta_c, c) d\mathbf{s}_c \tag{2.45}$$

where $d\mathbf{s}_c = \prod id s_{c,i}$. Thus the probability of generating a data vector from a C -component mixture model can be written as

$$p(\mathbf{X} | \mathcal{M}) = \sum_{c=1}^C p(c | \mathbf{k}) p(\mathbf{x} | \Theta_c, c) \tag{2.46}$$

where $p(c|\mathbf{k}) = \{p(c=1)=k_1, p(c=2)=k_2, \dots, p(c=C)=k_c\}$. $p(\mathbf{x}|\mathcal{M})$ is known as the evidence for model \mathcal{M} and quantifies the likelihood of the observed data under model \mathcal{M} . A Bayesian solution can be obtained by integrating out the parameters $\{\mathbf{k}, \Theta_c\}$ and hidden variables $\{\mathbf{s}_c, \mathbf{q}_c\}$. A set of prior distributions is assumed over all possible parameter values. For instance, the prior over the source model (MoG) parameters is defined as a product of priors over π_c, μ_c, β_c , thus $p(\varphi) = \prod_{c=1}^C p(\pi_c)p(\mu_c)p(\beta_c)$. In addition, the following priors are defined over:

ICA mixture indicator variables $p(\mathbf{c}|\mathbf{k})$; ICA mixture coefficients $p(\mathbf{k})$; mixture proportions $p(\pi)$, mean and precision over each MoG $p(\boldsymbol{\mu})$ and $p(\beta)$; bias vector $p(\mathbf{y})$; sensor noise precision $p(\lambda)$; each element of the mixing matrix $p(\mathbf{A})$ with precision α_i for each column; and relevance of each source $p(\alpha)$.

The optimization follows from Bayes' rule $\log p(\mathbf{X}) = \log \frac{p(\mathbf{X}, \boldsymbol{\psi})}{p(\boldsymbol{\psi}|\mathbf{X})}$. The term $\boldsymbol{\psi}$ is the vector of all hidden variables and unknown parameters. This can be written as

$$\begin{aligned} \log p(\mathbf{X}) &= \int p'(\boldsymbol{\psi}) \log \frac{p'(\boldsymbol{\psi})p(\mathbf{X}, \boldsymbol{\psi})}{p'(\boldsymbol{\psi})p(\boldsymbol{\psi}|\mathbf{X})} d\boldsymbol{\psi} \\ &= \int p'(\boldsymbol{\psi}) \log \frac{p'(\mathbf{X}, \boldsymbol{\psi})}{p'(\boldsymbol{\psi})} d\boldsymbol{\psi} + \int p'(\boldsymbol{\psi}) \log \frac{p'(\boldsymbol{\psi})}{p(\boldsymbol{\psi}|\mathbf{X})} d\boldsymbol{\psi} \quad (2.47) \\ &= F[\boldsymbol{\psi}] + KL[p' || p] \end{aligned}$$

where $p'(\boldsymbol{\psi})$ is some approximation to the posterior $p(\boldsymbol{\psi}|\mathbf{X})$; $F[\boldsymbol{\psi}] = \langle \log p(\mathbf{X}, \boldsymbol{\psi}) \rangle_{p'(\boldsymbol{\psi})} + \mathcal{H}[p'(\boldsymbol{\psi})]$; and $KL[p' || p] = \int p'(\boldsymbol{\psi}) \log \frac{p'(\boldsymbol{\psi})}{p(\boldsymbol{\psi}|\mathbf{X})} d\boldsymbol{\psi}$. $\mathcal{H}[p'(\boldsymbol{\psi})]$ is the entropy of $p'(\boldsymbol{\psi})$, and KL is the Kullback–Leibler divergence.

In the mixture model $\pi = \{\mathbf{c}, \mathbf{s}, \mathbf{q}, \mathbf{k}, \Theta\}$. By choosing $p'(\boldsymbol{\psi})$ such that it factorizes, terms in each hidden variable can be maximized individually. In [53], the following factorization was chosen,

$$p'(\boldsymbol{\psi}) = p'(\mathbf{c})p'(\mathbf{s}_c|\mathbf{q}_c, c)p'(\mathbf{q}_c|c)p'(\mathbf{k})p'(\mathbf{y})p'(\lambda)p'(\mathbf{A})p'(\alpha)p'(\boldsymbol{\phi}) \quad (2.48)$$

where $p'(\boldsymbol{\phi}) = p'(\pi)p'(\boldsymbol{\mu})p'(\beta)$ and $p'(a|b)$ is the approximating density of $p(a|b, \mathbf{X})$. Also the posteriors over the sources were factorized such that

$$p'(\mathbf{s}_c, \mathbf{q}_c|c) = \prod_{i=1}^{L_c} p'(q_c|c)p'(s_{c,i}|q_i, c).$$

2.5 Conclusions

In this chapter, an overview of the current techniques in ICA and ICA mixture modelling (ICAMM) has been carried out. These techniques establish a framework for non-linear processing of data with complex non-gaussian distributions.

Classical statistical signal processing relies on exploiting second-order information. Spectral analysis and linear adaptive filtering are probably the most representative examples. From the perspective of optimality (optimum detection and estimation), second-order statistics are sufficient statistics when Gaussianity holds, but lead to suboptimum solutions when dealing with general probability density models. A natural evolution of statistical signal processing, in connection with the progressive increase in computational power, has been exploiting higher-order information. Thus, high-order spectral analysis and nonlinear adaptive filtering have received the attention of many researchers in this field.

Clearly, within this framework of evolution from second-order to higher-order information, is the transition from PCA to ICA. Briefly, PCA is a technique for linearly transforming a vector of correlated components into a vector of variance-ordered uncorrelated components; meanwhile ICA linearly transforms a vector of statistically dependent components into unordered independent components. ICA can also be considered as a natural evolution of prewhitening linear transformation (like PCA but no variance ordering is being produced). When Gaussianity holds, both ICA and prewhitening get equivalent transformations, and infinite solutions may exist, as any rotation of the prewhitened vector keeps the uncorrelation among the vector components. However, when non-gaussianity appears, ICA produces a different transformation, which can be unique if appropriate constraints are introduced into the design. That is the reason why ICA has become so popular as a technique for blind source separation when at maximum one source is Gaussian.

Even more interesting is to recognize that ICA implicitly assumes a model for multivariate pdf's. The multivariate pdf of the transformed vector will be the product of the (one-dimensional) marginal pdf's of its components. Dealing with one-dimensional pdf's makes different complex problems involving multivariate pdf's tractable. This perspective suggests that ICA can be an interesting tool for use in areas of intensive data analysis. Actually, dealing with estimates of pdf's or defining optimality criteria involving pdf's (like entropy, mutual information, Kullback–Leibler distances, etc.) can be considered the last generation in statistical signal processing approaches: a natural evolution from second-order and higher-order statistics to data distribution information. In this chapter, we have reviewed some of the most representative ICA algorithms derived from entropy, cumulant, and time structure perspectives: InfoMax, JADE, FastIca, and TDSEP. In addition, we have reviewed the principal non-parametric ICA algorithms (Npica, Radical, Kernel-ICA) from a research direction that pursues generalization of the methods; and thus BSS is done with completely unknown information about the sources.

Some authors have termed the approaches above as non-linear information processing [66]. This is relevant since non-linear information processing establishes a bridge between statistical signal processing and computational and artificial intelligence sciences. That is why many people from signal processing are increasingly involved in areas like data mining, machine learning, or clustering, and many researchers from computational sciences are working on new data intensive signal and image processing applications.

Recently, ICAMM was introduced as an extension of ICA. ICAMM is a kind of nonlinear ICA technique that extends the linear ICA method by learning multiple ICA models and weighting them in a probabilistic manner. Thus, ICAMM has emerged as a flexible approach to model arbitrary data densities using mixtures of multiple ICA models with non-gaussian distributions for the independent components (i.e., relaxing the restriction of modelling every component by a multi-variate Gaussian probability density function).

In this chapter, we reviewed three ICAMM methods. The first method is maximum likelihood-based, which uses the learning rule of extended InfoMax algorithm in the ICAMM parameter updating to distinguish between sub-gaussian and super-gaussian sources. The second method extracts the ICA classes sequentially from an initial estimate for each centroid and is based on a distance called Beta divergence, which is an extension of the Kullback–Leibler divergence. This method requires that various parameters be initialized, such as the beta value, initial centroids, a percentage of classification used as stopping criterion. These parameters are estimated rather arbitrarily. As in the first method, the extended InfoMax rule is used for unknown source distributions. Thus, the source model of these methods could only switch between Laplacian and bimodal densities, which is a limitation for source density estimation.

The third reviewed method is a variational Bayesian learning algorithm. This method uses a source model based on mixtures of Gaussians. In order to apply Bayesian inference, a set of prior distributions over all possible parameter values is assumed: source model (MoG), ICA mixture indicator variables; ICA mixture coefficients; mixture proportions, mean and precision over each MoG; bias vector; sensor noise precision; precision for each column of the mixing matrix; and relevance of each source. The algorithm uses variational optimization (to lighten the computationally expensive cost of Bayesian inference) to approximate integrating out the parameters. Taking into account that the source model for every ICA is MoG, the final ICAMM data model for this method can be also considered a kind of MoG.

In the case of the first two ICAMM methods reviewed, the number of clusters is known a priori. In the third method, the number of clusters can be estimated, although substantial a priori knowledge is required for the model parameters. All three methods consider only unsupervised learning; therefore, semi-supervised and supervised learning are left unspecified.

References

1. C. Jutten, J. Herault, Une solution neuromimétique au problème de séparation de sources. *Traitement du Signal* **5**(6), 389–404 (1989)
2. C. Jutten, J. Herault, Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture. *Signal Process.* **24**, 1–10 (1991)
3. C. Jutten, J. Herault, Blind separation of sources, part II: problems statement. *Signal Process.* **24**, 11–20 (1991)

4. C. Jutten, J. Herault, Blind separation of sources, part III: stability analysis. *Signal Process.* **24**, 21–29 (1991)
5. A. Hyvärinen, J. Karhunen, E. Oja, *Independent Component Analysis* (Wiley, New York, 2001)
6. C.W. Hesse, C.J. James, On semi-blind source separation using spatial constraints with applications in EEG Analysis. *IEEE Trans. Biomed. Eng.* **53**(12-1), 2525–2534 (2006)
7. J. Even, K. Sugimoto, An ICA approach to semi-blind identification of strictly proper systems based on interactor polynomial matrix. *Int. J. Robust Nonlinear Control* **17**, 752–768 (2007)
8. Z. Ding, T. Ratnarajah, C.F.N. Cowan, HOS-based semi-blind spatial equalization for MIMO rayleigh fading channels. *IEEE Trans. Signal Process.* **56**(1), 248–255 (2008)
9. A. Cichocki, S. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications* (Wiley, New York, 2001)
10. T.W. Lee, *Independent Component Analysis—Theory and Applications* (Kluwer Academic Publishers, Boston, 1998)
11. S. Roberts, R. Everson, *Independent Component Analysis—Principles and Practice* (Cambridge University Press, Cambridge, 2001)
12. A. Cichocki, R. Zdunek, A.H. Phan, S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation* (Wiley, Hoboken, 2009)
13. P. Comon, C. Jutten (eds.), *Handbook of Blind Source Separation Independent Component Analysis and Applications* (Academic Press, Oxford, 2010)
14. M.S. Pedersen, J. Larsen, U. Kjems, L.C. Parra, *A Survey of Convolutional Blind Source Separation Methods*, ed. by J. Benesty, A. Huang. *Multichannel Speech Processing Handbook*, Chapter 51 (Springer, Berlin, 2007), pp. 1065–1084
15. H. Buchner, R. Aichner, W. Kellerman, TRINICON: a versatile framework for multichannel blind signal processing. in *Proceedings of 29th IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, pp. III-889–892, Montreal, Canada, 2004
16. W. Kellerman, H. Buchner, R. Aichner, Separating convolutive mixture with TRINICON. in *Proceedings of 31st IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, pp. V-961–964, Toulouse, France, 2006
17. P. Comon, Independent component analysis—a new concept? *Signal Process.* **36**(3), 287–314 (1994)
18. S. Amari, A. Cichocki, H. Yang, *A new learning algorithm for blind signal separation*, *Advances in Neural Information Processing Systems*, vol 8 (MIT Press, Cambridge, 1996), pp. 752–763
19. S. Amari, J.F. Cardoso, Blind source separation-semiparametric statistical approach. *IEEE Trans. Signal Process.* **45**(11), 2692–2700 (1997)
20. A. Hyvärinen, E. Oja, A fast fixed-point algorithm for independent component analysis. *Neural Comput.* **9**(7), 1483–1492 (1998)
21. D.T. Pham, P. Garrat, Blind separation of mixture of independent sources through a quasi-maximum likelihood approach. *IEEE Trans. Signal Process.* **45**(7), 1712–1725 (1997)
22. A. Hyvärinen, Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. Neural Netw.* **10**(3), 626–634 (1999)
23. T.W. Lee, M. Girolami, T.J. Sejnowski, Independent component analysis using an extended InfoMax algorithm for mixed sub-gaussian and super-gaussian sources. *Neural Comput.* **11**(2), 417–441 (1999)
24. S.I. Amari, T.P. Chen, A. Cichocki, Nonholonomic orthogonal learning algorithms for blind source separation. *Neural Comput.* **12**, 1463–1484 (2000)
25. J.F. Cardoso, Dependence, correlation and gaussianity in independent component analysis. *J. Mach. Learn. Res.* **4**, 1177–1203 (2003)
26. A. Chen, P.J. Bickel, Consistent independent component analysis and prewhitening. *IEEE Trans. Signal Process.* **53**(10), 3625–3632 (2005)
27. W. Liu, D.P. Mandic, A. Cichocki, Blind source extraction based on a linear predictor. *IET Signal Process.* **1**(1), 29–34 (2007)

28. J.F. Cardoso, Blind signal separation: statistical principles. *Proceedings of the IEEE. Special Issue on Blind Identification and Estimation*, vol 9, pp. 2009–2025, 1998
29. A. Chen, P.J. Bickel, Efficient independent component analysis. *Annals Stat.* **34**(6), 2825–2855 (2006)
30. F. Meinecke, A. Ziehe, M. Kawanabe, K.R. Müller, Resampling approach to estimate the stability of one-dimensional or multidimensional independent components. *IEEE Trans. Biomed. Eng.* **49**(12), 1514–1525 (2002)
31. J. Himberg, A. Hyvärinen, F. Esposito, Validating the independent components of neuroimaging time-series via clustering and visualization. *Neuroimage* **22**(3), 1214–1222 (2004)
32. A.J. Bell, T.J. Sejnowski, An information-maximization approach to blind separation and blind deconvolution. *Neural Comput.* **7**, 1129–1159 (1995)
33. J.F. Cardoso, A. Souloumiac, Blind beamforming for non gaussian signals. *IEE Proc.-F* **140**(6), 362–370 (1993)
34. A. Ziehe, K.R. Müller, TDSEP- an efficient algorithm for blind separation using time structure. *Proceedings of the 8th International Conference on Artificial Neural Networks, ICANN'98, Perspectives in Neural Computing*, pp. 675–680, 1998
35. J.P. Nadal, N. Parga, Non linear neurons in the noise limit: a factorial code maximizes information transfer. *Netw. Comput. Neural Syst.* **5**(3), 565–585 (1994)
36. J.F. Cardoso, InfoMax and maximum likelihood for blind source separation. *IEEE Signal Process. Lett.* **4**(4), 112–114 (1997)
37. S.I. Amari, Natural gradient works efficiently in learning. *Neural Comput.* **10**, 251–276 (1998)
38. J.F. Cardoso, B. Laheld, Equivariant adaptive source separation. *IEEE Trans. Signal Process.* **45**(2), 434–444 (1996)
39. C. Nikias, A. Petropulu, *Higher-order Spectral Analysis—A Nonlinear Signal Processing Framework* (Prentice Hall, Englewood Cliffs, 1993)
40. J.F. Cardoso, P. Comon, Tensor-based independent component analysis. *Proceedings of the Fifth European Signal Processing Conference, EUSIPCO 1990*, pp. 673–676, 1990
41. J.F. Cardoso, A. Souloumiac, Jacobi angles for simultaneous diagonalization. *SIAM J. Matrix Anal. Appl.* **17**(1), 161–164 (1996)
42. J.F. Cardoso, High-order contrasts for independent component analysis. *Neural Comput.* **11**(1), 157–192 (1999)
43. A. Ziehe, K.R. Müller, G. Nolte, B.M. Mackert, G. Curio, Artifact reduction in magnetoneurography based on time-delayed second order correlations. *IEEE Trans. Biomed. Eng.* **41**, 75–87 (2000)
44. A. Belouchrani, K. Abed-Meraim, J.F. Cardoso, E. Moulines, A blind source separation technique using second-order statistics. *IEEE Trans. Signal Process.* **45**, 434–444 (1997)
45. R. Boscolo, H. Pan, Independent component analysis based on nonparametric density estimation. *IEEE Trans. Neural Netw.* **15**(1), 55–65 (2004)
46. R. Boustany, J. Antoni, Blind extraction of a cyclostationary signal using reduced-rank cyclic regression—a unifying approach. *Mech. Syst. Signal Process.* **22**, 520–541 (2008)
47. J. Even, K. Sugimoto, An ICA approach to semi-blind identification of strictly proper systems based on interactor polynomial matrix. *Int. J. Robust Nonlinear Control* **17**, 752–768 (2007)
48. F.R. Bach, M.I. Jordan, Kernel independent component analysis. *J. Mach. Learn. Res.* **3**, 1–48 (2002)
49. T. Hastie, R. Tibshirani, *Independent Component Analysis Through Product Density Estimation*, Technical Report, Stanford University, 2002
50. E.G. Learned-Miller, J.W. Fisher, ICA using spacings estimates of entropy. *J. Mach. Learn. Res.* **4**, 1271–1295 (2003)
51. A. Samarov, A. Tsybakov, Nonparametric independent component analysis. *Bernoulli* **10**(4), 565–582 (2004)
52. B.W. Silverman, *Density Estimation for Statistics and Data Analysis* (Chapman and Hall, London, 1985)

53. R. Choudrey, S. Roberts, Variational mixture of bayesian independent component analysers. *Neural Comput.* **15**(1), 213–252 (2002)
54. M.E. Tipping, C.M. Bishop, Mixtures of probabilistic principal component analysers. *Neural Comput.* **11**(2), 443–482 (1999)
55. Z. Ghahramani, M. Beal, Variational inference for Bayesian mixtures of factor analysers. *Adv. Neural Inf. Process. Syst.* **12**, 449–445 (2000)
56. C. Archambeau, N. Delannay, M. Verleysen, Mixtures of robust probabilistic principal component analyzers. *Neurocomputing* **71**(7–9), 1274–1282 (2008)
57. M. Svensén, C.M. Bishop, Robust Bayesian mixture modelling. *Neurocomputing* **64**, 235–252 (2005)
58. T.W. Lee, M.S. Lewicki, T.J. Sejnowski, ICA mixture models for unsupervised classification of non-gaussian classes and automatic context switching in blind signal separation. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(10), 1078–1089 (2000)
59. S. Roberts, W.D. Penny, Mixtures of independent component analysers. in *Proceedings of ICANN2001*, Vienna, August 2001, pp. 527–534
60. J.A. Palmer, K. Kreutz-Delgado, S. Makeig, An Independent Component Analysis Mixture Model with Adaptive Source Densities, Technical Report, UCSD, 2006
61. K. Chan, T.W. Lee, T.J. Sejnowski, Variational learning of clusters of undercomplete nonsymmetric independent components. *J. Mach. Learn. Res.* **3**, 99–114 (2002)
62. C.T. Lin, W.C. Cheng, S.F. Liang, An on-line ICA-mixture-model-based self-constructing fuzzy neural network. *IEEE Trans. Circuits Syst.* **52**(1), 207–221 (2005)
63. T. Yoshida, M. Sakagami, K. Yamazaki, T. Katura, M. Iwamoto, N. Tanaka, Extraction of neural activity from in vivo optical recordings using multiple independent component analysis. *IEEJ Trans. Electron. Inf. Syst.* **127**(10), 1642–1650 (2007)
64. J.A. Palmer, S. Makeig, K. Kreutz-Delgado, B.D. Rao, Newton method for the ICA mixture model. *Proceedings of the 33rd IEEE International Conference on Acoustics, Speech, and Signal*, pp. 1805–1808, Las Vegas, USA, 2008
65. N.H. Mollah, M. Minami, S. Eguchi, Exploring latent structure of mixture ICA models by the minimum β -Divergence method. *Neural Comput.* **18**, 166–190 (2005)
66. D. Erdogmus, J.C. Principe, From linear adaptive filtering to nonlinear information processing—the design and analysis of information processing systems. *IEEE Signal Process. Mag.* **23**(6), 14–33 (2006)

On Statistical Pattern Recognition in Independent
Component Analysis Mixture Modelling

Salazar, A.

2013, XXII, 186 p., Hardcover

ISBN: 978-3-642-30751-5