Independent Component Analysis (ICA) and Method of Estimating Functions

Shun-ichi Amari
RIKEN Brain Science Institute
2-1 Hirosawa, Wako, Saitama 351-0198, Japan

Summary

Independent component analysis (ICA) is a new method of extracting independent components from multivariate data. It can be applied to various fields such as vision and auditory signal analysis, communication systems, and biomedical and brain engineering. There have been proposed a number of algorithms. The present article shows that most of them use estimating functions from the statistical point of view, and give a unified theory, based on information geometry, to elucidate the efficiency and stability of the algorithms. This gives new efficient adaptive algorithms useful for various problems.

keywords: independent components, information geometry, estimating functions, learning, signal processing

1 Introduction

Principal component analysis (PCA) has been widely used for decomposing multivariate statistical signals into independent components. However, PCA implicitly assumes that signals are subject to multivariate Gaussian distributions, and uses orthogonal bases to decompose signals. Under the Gaussian assumption, there are infinitely many independent decompositions, and PCA chooses one by forcing that the basis should be orthonormal.

In many situations, signals are not Gaussian, and can be decomposed into (approximately) independent components by using a non-orthogonal basis, where non-Gaussianity plays a fundamental role. A typical example is the separation of voices from individual speakers in the cocktail-party problem, where voices are linearly mixed and the mixture matrix is not orthogonal so that a non-orthogonal transformation is necessary for unmixing.

Similar problems are formulated in the case where independent source signals are temporally correlated, and in the case where the mixing process is not instantaneous but convolutive. Image data can also be decomposed in a suitable basis.

There are a number of algorithms for ICA [7, 12, 13, 14, 16]. See also monographs [15, 17, 20]. Some are on-line, but some are in the batch mode, although most online algorithms can easily be converted to the batch mode by taking the average. Some algorithms decompose all the components in parallel, while some extract components one by one sequentially. Intermediate algorithms extract several components in parallel.

The present article uses the method of estimating functions ([2, 4, 5]) to provide a unified viewpoint to this problem. Information geometry [11] elucidated the fundamental structure of estimating functions in semiparametric statistical models which includes unknown functions as parameters [10]. We show most existing methods are derived from estimating functions, and their differences are only in the choices of estimating functions.

We then give error analysis and stability analysis in terms of estimating functions. This makes it possible to design various adaptive methods for choosing unknown parameters included in estimating functions, which control accuracy and stability. The article is based on a series of works ([7, 2, 3, 4, 5, 6, 7, 8, 9]) by the author including recent unpublished works.

2 Statements of the Problems and Methods

Let $\mathbf{x}(t) = [x_1(t), \cdots, x_n(t)]^T$, $t = 1, 2, \cdots$, be $n$-dimensional multivariate signals observed at time $t$, where $T$ denotes transposition of a vector or matrix. We assume that they are instantaneous mixtures of $m$ independent signals $\mathbf{s}(t) = (s_1(t), \cdots, s_m(t))^T$ generated at time $t$, as

$$\mathbf{x}(t) = \mathbf{H}\mathbf{s}(t), \tag{1}$$

where $\mathbf{H}$ is an $n \times m$ unknown matrix. Here, s_i and s_j $(i \neq j)$ are independent, but each signal may have temporal correlations.

Let $\mathbf{W}$ be an $m \times n$ matrix, which is a candidate of unmixing $\mathbf{x}(t)$ by

$$\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t) \tag{2}$$

such that $y_i(t)$, $i = 1, \cdots, m$, are original independent random variables, or a rescaled and permuted version of the original source signals $s_i(t)$, that is,

$$y_i(t) = c_{i'}s_{i'}(t) \tag{3}$$

where $(1', \cdots, m')$ is a permutation of $(1, \cdots, m)$ and c_i are constants.

An on-line learning method uses a candidate matrix $\mathbf{W}_t$ at time t, and calculates

$$\mathbf{y}(t) = \mathbf{W}_t\mathbf{x}(t), \tag{4}$$

which is hoped to be the original independent source signals. Then, the candidate $\mathbf{W}_t$ is updated by

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta\mathbf{F}\left(\mathbf{x}(t), \mathbf{W}_t\right), \tag{5}$$

where η is a learning constant (which may depend on t) and $\mathbf{F}$ is a matrix-valued function, such that $\mathbf{W}_t$ converges to the true solution.

There have been proposed various $\mathbf{F}$, which are derived in many cases (but not in all cases) as the gradients of cost functions to be minimized. The cost functions are, for example, higher-order cumulants, entropy, negative log likelihood and others. In many cases, algorithms include free parameters, sometimes free functions, to be chosen adequately or to be determined adaptively. Since the probability density functions of the source signals are unknown, there are no way to avoid such parameters.

There are conditions which the function $\mathbf{F}$ should satisfy. If the algorithm using $\mathbf{F}$ converges to the true solution $\mathbf{W}$, $\mathbf{W}$ should be an equilibrium of dynamics (5). Since this is a stochastic difference equation, we use its continuous time version for mathematical analysis,

$$\frac{d}{dt}\mathbf{W}(t) = -\eta\mathbf{F}\left\{\mathbf{x}(t), \mathbf{W}(t)\right\}. \tag{6}$$

Its expected version is

$$\frac{d}{dt}\mathbf{W}(t) = -\eta E\left[\mathbf{F}\left\{\mathbf{x}(t), \mathbf{W}(t)\right\}\right] \tag{7}$$

where expectation E is taken with respect to $\mathbf{x}(t)$. The condition that the true solution $\mathbf{W}$ is an equilibrium of (7) is given by

$$E\left[\mathbf{F}(\mathbf{x}, \mathbf{W})\right] = 0, \tag{8}$$

where $\mathbf{x} = \mathbf{H}\mathbf{s}$, except for indeterminacy due to permutations and rescaling.

A function $\mathbf{F}(\mathbf{x}, \mathbf{W})$ satisfying (8) for the true $\mathbf{W}$ is called the estimating function in semiparametric statistical model, as is shown in the next section.

3 Semiparametric Statistical Model and Estimating Function

We formulate the problem in the statistical framework. Let $r_a(s_a)$ be the probability density function of s_a. The joint probability density of s is written as

$$r(\mathbf{s}) = \prod_{a=1}^{n} r_a(s_a) \tag{9}$$

since they are independent. The observation vector $\mathbf{x}$ is a linear function of s, so that its probability density function is given in terms of $\mathbf{W} = \mathbf{H}^{-1}$ by

$$p_X(\mathbf{x}; \mathbf{W}, r) = \det|\mathbf{W}| r(\mathbf{W}\mathbf{x}), \tag{10}$$

where we assumed $n = m$ and $\mathbf{H}^{-1}$ exists for simplicity. We also assume

$$E\left[s_a\right] = 0. \tag{11}$$

Since we do not know r except that it is a product form (9), the probability model of $\mathbf{x}$ includes two parameters, $\mathbf{W}$ called the "parameter of interest" which we want to estimate, and the unknown function $r = r_1 \cdots r_n$ which is called the "nuisance parameter (function)" and which we do not care. Such a statistical model is called a semiparametric model and estimation of the parameter of interest is in general a difficult problem because of the existence of unknown functions.

A method of estimating functions has been developed for semiparametric statistical models in the framework of information geometry (Amari and Kawanabe [10]; Amari and Nagaoka [11] as for information geometry).

An estimating function in the present case is a matrix-valued function $\mathbf{F}(\mathbf{x}, \mathbf{W}) = \{F_{ab}(\mathbf{x}, \mathbf{W})\}$ of $\mathbf{x}$ and $\mathbf{W}$ not including the nuisance parameter r, that satisfies

$$1)\ E_{\mathbf{W},r}[\mathbf{F}(\mathbf{x}, \mathbf{W})] = 0, \tag{12}$$

Here, suffices $a, b, c, \cdots$ represent components of the original source signals or recovered signals, $\mathbf{s}$ or $\mathbf{y}$, $E_{\mathbf{W},r}$ denotes expectation with respect to probability distribution given by (10), and it is required that (12) holds for all r of the form (9). In order to avoid a trivial $\mathbf{F}$ such as $\mathbf{F} = 0$, we require that

$$2)\ \mathbf{L} = E_{\mathbf{W},r}\left[\frac{\partial}{\partial \mathbf{W}}\mathbf{F}(\mathbf{x}, \mathbf{W})\right]. \tag{13}$$

is non-degenerate. This guarantees that

$$E_{\mathbf{W},r}\left[\mathbf{F}\left(\mathbf{x}, \mathbf{W}'\right)\right] \neq 0 \tag{14}$$

for $\mathbf{W}' \neq \mathbf{W}$ at least locally. It should be noted that $\mathbf{L}$ is a matrix-by-matrix linear operator that maps a matrix to a matrix. The components of $\mathbf{L}$ are

$$L_{ab,ci} = E_{\mathbf{W},r}\left[\frac{\partial}{\partial W_{ci}}F_{ab}(\mathbf{x}, \mathbf{W})\right], \tag{15}$$

where W_{ci} denote elements of $\mathbf{W}$, and suffices i, j, k, etc. representing components of observed signals $\mathbf{x}$. It is convenient to use capital indices A, $B, \cdots$ to represent a pair (a, b), (c, i) and so on of indices. Then, for $A = (a, b)$, $B = (c, i)$, $\mathbf{L}$ has a matrix representation $\mathbf{L} = (L_{AB})$ that operates on $(W_B) = (W_{ci})$ as

$$\mathbf{LW} = \sum_B L_{AB}W_B = \sum_{c,i} L_{ab,ci}W_{ci}. \tag{16}$$

The inverse of $\mathbf{L}$ is defined by the inverse matrix of $\mathbf{L} = (L_{AB})$.

When an estimating function $\mathbf{F}(\mathbf{x}, \mathbf{W})$ is found, given observed data $\mathbf{x}(1), \cdots, \mathbf{x}(t)$, we have the estimating equation

$$\sum_{i=1}^{t} \mathbf{F}\{\mathbf{x}(i), \mathbf{W}\} = 0, \tag{17}$$

of which the solution gives an estimator $\hat{\mathbf{W}}$. This is derived by replacing the expectation in (12) by the empirical sum of observations. An on-line learning algorithm is given by (5). These equations work without making use of the unknown r. The problem is how to find a "good" estimating function $\mathbf{F}$ if there are some.

A number of heuristic estimating functions have been proposed including Jutten and Herault [16]; Bell and Sejnowski [12]; Amari, Cichocki and Yang [8]; Cardoso and Laheld [13]; Oja [21]. Estimating function $\mathbf{F}$ is better than $\mathbf{F}'$, when the expected error of estimator $\hat{\mathbf{W}}$ derived by $\mathbf{F}$ is smaller than that by $\mathbf{F}'$. However, it may happen that $\mathbf{F}$ is better than $\mathbf{F}'$ when the true (unknown) distribution is $r(\mathbf{s})$ but $\mathbf{F}'$ is better when it is r'. Hence, they are in general not comparable. A family of estimating functions is said to be admissible, when, given any estimator, an equivalent or better estimating function can be found in the family. Moreover, this class includes the best estimator (that is, the Fisher efficient estimator) in the sense that it satisfies the extended Cramér-Rao bound asymptotically.

Amari and Cardoso [5] used the general theory of Amari and Kawanabe [10] to prove that functions of the form

$$\mathbf{F}(\mathbf{x}, \mathbf{W}) = \varphi(\mathbf{y})\mathbf{y}^T - \mathbf{I}, \tag{18}$$

or

$$F_{ab}(\mathbf{x}, \mathbf{W}) = \varphi_a(y_a)y_b - \delta_{ab}$$

in component form, where

$$\varphi(\mathbf{y}) = [\varphi_1(y_1), \cdots, \varphi_n(y_n)]^T$$

consists of arbitrary non-trivial functions φ_a, are estimating functions. They together with their linear concomitants, give a set of admissible estimating functions. This (18) is an estimating function as is easily shown. Indeed, when $\mathbf{W}$ is the true solution, y_a and y_b are independent. Therefore, whatever r is,

$$E_{r,\mathbf{W}}\left[\varphi_a\left(y_a\right)y_b\right] = E\left[\varphi_a\left(y_a\right)\right]E\left[y_b\right] = 0, \quad a \neq b. \tag{19}$$

However, when $\mathbf{W}$ is not the true solution, the above equation does not hold in general. When $a = b$, we have

$$E\left[\varphi_a\left(y_a\right)y_a\right] = 1, \tag{20}$$

which specifies the magnitude of the recovered signal. Since the magnitude may be arbitrary, we may put the diagonal terms F_{aa} arbitrarily.

We give typical examples of estimating functions. Let

$$q(\mathbf{s}) = \prod_{a=1}^{m} q_a\left(s_a\right) \tag{21}$$

be a (possibly misspecified) joint probability density function of $\mathbf{s}$, which might be different from the unknown true one

$$r(\mathbf{s}) = \prod r_a\left(s_a\right). \tag{22}$$

The negative log likelihood of $\mathbf{x}$ derived therefrom is

$$l(\mathbf{x}, \mathbf{W}) = \det|\mathbf{W}| + \sum_{i=1}^{n} \log q_a\left(y_a\right), \tag{23}$$

where y_a is the a-th component of $\mathbf{y} = \mathbf{Wx}$, depending on both $\mathbf{x}$ and $\mathbf{W}$. The criterion of minimizing l is interpreted as maximization of the entropy, or maximization of the likelihood, although it includes unknown functions $q_a\left(y_a\right)$. Let us put

$$\varphi_a\left(y_a\right) = -\frac{d}{dy_a} \log q_a\left(y_a\right). \tag{24}$$

The gradient of l gives an estimating function

$$\tilde{\mathbf{F}}(\mathbf{x}, \mathbf{W}) = \frac{\partial l(\mathbf{x}, \mathbf{W})}{\partial \mathbf{W}} = \mathbf{W}^{-T} - \varphi(\mathbf{y})\mathbf{x}^T, \tag{25}$$

where $\mathbf{W}^{-T}$ is the transpose of the inverse of $\mathbf{W}$ and $\varphi = [\varphi_1, \cdots, \varphi_n]^T$. We can prove that this $\tilde{\mathbf{F}}$ is an estimating function. However, when $\tilde{\mathbf{F}}$ is an estimating function,

$$\mathbf{F}(\mathbf{y}) = \tilde{\mathbf{F}}(\mathbf{x}, \mathbf{W})\mathbf{W}^T\mathbf{W} = \left\{I - \varphi(\mathbf{y})\mathbf{y}^T\right\}\mathbf{W} \tag{26}$$

is also an estimating function and vice versa. It is easy to prove that

$$E\left[\mathbf{F}(\mathbf{y})\right] = 0 \tag{27}$$

and

$$E\left[\tilde{\mathbf{F}}(\mathbf{x}, \mathbf{W})\right] = 0 \tag{28}$$

are equivalent.

When the true distributions are r_a, the best choice of φ_a is

$$\varphi_a(s) = -\frac{d}{ds}\log r_a(s). \tag{29}$$

This gives the maximum likelihood estimator (Pham and Garat [22]). However, even when we use a different φ_a, the estimating equation (17) gives a $\sqrt{t}$-consistent estimator, that is, the estimation error converges to 0 in probability in the order of $1/\sqrt{t}$ as t goes to infinity.

It is easy to show that similar estimating functions are derived from the criterion of maximizing higher-order cumulants and others. The algorithms given by Cardoso, Jutten-Herault, Oja etc use respective estimating functions.

We have shown that $\tilde{\mathbf{F}}(\mathbf{x}, \mathbf{W})$ and $\mathbf{F}(\mathbf{y})$ are equivalent estimating functions, because they are linearly related. More generally, let $\mathbf{R}(\mathbf{W})$ be an arbitrary nonsingular linear operator acting on matrices. When $\mathbf{F}(\mathbf{x}; \mathbf{W})$ is an estimating function matrix, $\mathbf{R}(\mathbf{W})\mathbf{F}(\mathbf{x}; \mathbf{W})$ is also an estimating function matrix, because

$$\begin{aligned} & E_{\mathbf{W},r}[\mathbf{R}(\mathbf{W})\mathbf{F}(\mathbf{x}; \mathbf{W})] \\ = \ & \mathbf{R}(\mathbf{W})E_{\mathbf{W},r}[\mathbf{F}(\mathbf{x}; \mathbf{W})] = 0. \end{aligned} \tag{30}$$

Moreover, $\mathbf{F}$ and $\mathbf{RF}$ are equivalent in the sense that the derived batch estimators are exactly the same, because the two estimating equations

$$\sum \mathbf{F}\{\mathbf{x}(i); \mathbf{W}\} = 0$$
$$\sum \mathbf{R}(\mathbf{W})\mathbf{F}\{\mathbf{x}(i); \mathbf{W}\} = 0$$

give the same solution $\hat{\mathbf{W}}_t$. This defines an equivalent class of estimating functions which are essentially the same in batch estimation.

However, two equivalent estimating functions $\mathbf{F}(\mathbf{x}, \mathbf{W})$ and $\mathbf{R}(\mathbf{W})\mathbf{F}(\mathbf{x}, \mathbf{W})$ give different dynamical properties in on-line learning. That is, the dynamical properties of on-line learning algorithms

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta \mathbf{F}\{\mathbf{x}(t), \mathbf{W}_t\} \tag{31}$$
$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta \mathbf{R}(\mathbf{W})\mathbf{F}\{\mathbf{x}(t), \mathbf{W}_t\} \tag{32}$$

are completely different. Therefore, instead of the form (18), we need to consider an enlarged type of estimating functions of the form $\mathbf{R}(\mathbf{W})\mathbf{F}$ to derive a good on-line estimator.

4 Stability of Estimating Functions

In order to show the dynamical properties of on-line learning by using **F** or **RF**, we first study the stability of the algorithm at the true solution **W**, which is an equilibrium of the dynamics. The stability of dynamics (7) at the equilibrium is given by studying the eigenvalues of its Hessian. See Amari, Chen and Cichocki [6]. For the stability analysis, let us put

$$\mathbf{W}(t) = \mathbf{W} + \delta\mathbf{W}(t), \tag{33}$$

where $\delta\mathbf{W}(t)$ is a small deviation from the true **W**. Then, (7) is rewritten as

$$\frac{d}{dt}\delta\mathbf{W}(t) = -\eta E\left[\mathbf{F}\left\{\mathrm{x}, \mathbf{W} + \delta\mathbf{W}(t)\right\}\right]. \tag{34}$$

It is convenient to use the non-holonomic variables

$$\delta\mathbf{X} = \delta\mathbf{W}\mathbf{W}^{-1}, \tag{35}$$

and rewrite the dynamics in the neighborhood of the true solution as

$$\frac{d}{dt}\delta\mathbf{X}(t) = -\eta E\left[\mathbf{F}(\mathrm{x}, \mathbf{W} + \delta\mathbf{X}\mathbf{W})\right]\mathbf{W}^{-1}. \tag{36}$$

By Taylor expansion, we have

$$\frac{d}{dt}\delta\mathbf{X}(t) = -\eta\mathbf{K}(\mathbf{W})\delta\mathbf{X}(t), \tag{37}$$

where

$$\mathbf{K}(\mathbf{W}) = \frac{\partial E\left[\mathbf{F}(\mathrm{x}, \mathbf{W})\mathbf{W}^{-1}\right]}{\partial\mathbf{X}} = \frac{\partial E\left[\mathbf{F}(\mathrm{x}, \mathbf{W})\right]}{\partial\mathbf{W}} \tag{38}$$

is a linear operater which maps a matrix to matrix. Since both $\mathbf{F} = (F_{ab})$ and $\mathbf{X} = (X_{cd})$ are matrices, **K** has four indices $K_{ab,cd}$

$$K_{ab,cd} = \frac{\partial F_{ab}}{\partial X_{cd}} \tag{39}$$

in the component form. At the true value **W** where $y_a = s_a$ is recovered, for **F** given by (18), **K** is calculated as

$$K_{ab,cd} = E[\varphi_a'(s_a)s_b^2]\delta_{bd}\delta_{ac} + \delta_{ad}\delta_{bc}, \tag{40}$$

where φ' denotes the derivative of φ. We derive the above result in the following.

In order to calculate the gradient of $\mathbf{F}$ with respect to $\mathbf{X}$, we put

$$\begin{aligned} d\mathbf{F}(\mathbf{x},\mathbf{W}) &= \mathbf{F}(\mathbf{x},\mathbf{W}+d\mathbf{W})-\mathbf{F}(\mathbf{x},\mathbf{W}) && (41)\\ &= \mathbf{F}(\mathbf{x},\mathbf{W}+d\mathbf{X}\mathbf{W})-\mathbf{F}(\mathbf{x},\mathbf{W}), && (42)\end{aligned}$$

where $d\mathbf{F}$ denotes the increment of $\mathbf{F}$ due to change $d\mathbf{W}$ of $\mathbf{W}$, and expand it in the form

$$dF_{ab}(\mathbf{x},\mathbf{W}) = \sum K_{ab,cd}(\mathbf{x},\mathbf{W})dX_{cd}. \tag{43}$$

We have then

$$K_{ab,cd} = \frac{\partial F_{ab}}{\partial X_{cd}} \tag{44}$$

and its expectation gives $K_{ab,cd}$.

For $\mathbf{F} = (F_{ab})$ given by (18)

$$F_{ab} = \varphi(y_a)y_b - \delta_{ab}, \tag{45}$$

we have

$$\begin{aligned} dF_{ab} &= d\varphi(y_a)y_b + \varphi(y_a)dy_b && (46)\\ &= \varphi'(y_a)dy_a y_b + \varphi(y_a)dy_b. && (47)\end{aligned}$$

From

$$d\mathbf{y} = d\mathbf{W}\mathbf{x} = d\mathbf{W}\mathbf{W}^{-1}\mathbf{W}\mathbf{x} = d\mathbf{X}\mathbf{y}, \tag{48}$$

we have

$$dy_a = \sum_{d=1}^{n} dX_{ad}y_d = \sum_{c,d=1}^{n} y_d\delta_{ac}dX_{cd}. \tag{49}$$

Therefore,

$$K_{ab,cd} = \varphi'(y_a)y_b y_d\delta_{ac} + \varphi(y_a)y_b\delta_{bc}. \tag{50}$$

At the true $\mathbf{W}$, y_a and y_b are independent for $a \neq b$. Hence,

$$\begin{aligned} &E[\varphi'(y_a)y_b y_d] = E[\varphi'(s_a)y_b^2]\delta_{ac}\delta_{bd}, && (51)\\ &E[\varphi(y_a)y_d] = \delta_{ad}. && (52)\end{aligned}$$

The diagonal term F_{aa} may be disregarded, because it can be arbitrary.

Many components of $\mathbf{K}$ vanish. For $a \neq b$,

$$\frac{\partial F_{ab}}{\partial X_{cd}} = 0, \tag{53}$$

unless $(a, b) = (c, d)$ or $(a, b) = (d, c)$. When the pairs (a, b) and (c, d) are equal, (39) gives

$$\begin{aligned} K_{ab,ab} &= k_a \sigma_b^2, \\ K_{ab,ba} &= 1, \end{aligned}$$

where

$$k_a = E[\varphi'(s_a)]. \tag{54}$$

and

$$\sigma_a^2 = E\left[y_a^2\right]. \tag{55}$$

Let us summarize the above results. To this end, we denote $\mathbf{K}$ in the pairwise component form of the enlarged matrix $\mathbf{K} = (K_{AB})$. The results show that, for $A = (a, b)$, $a \neq b$, $K_{AB} = 0$ except for $B = (a, b)$ or $B = (b, a)$. This shows that $\mathbf{K} = (K_{AB})$ is decomposed in the two-by-two minor diagonal matrices of $\partial F_{ab}/\partial X_{ab}, \partial F_{ab}/\partial X_{ba}, \partial F_{ba}/\partial X_{ab}$ and $\partial F_{ba}/\partial X_{ba}$,

$$\begin{bmatrix} K_{AA} & K_{AA'} \\ K_{A'A} & K_{A'A'} \end{bmatrix} = \begin{bmatrix} k_a\sigma_b^2 & 1 \\ 1 & k_b\sigma_a^2 \end{bmatrix}, \tag{56}$$

where $A = (a, b)$ and $A' = (b, a)$ (see [6], [13], [22]).

The inverse of $\mathbf{K}$ has also the same diagonalized form, for (A, A')-part,

$$\begin{bmatrix} k_a\sigma_b^2 & 1 \\ 1 & k_b\sigma_a^2 \end{bmatrix}^{-1} = c_{ab} \begin{bmatrix} k_b\sigma_a^2 & -1 \\ -1 & k_a\sigma_b^2 \end{bmatrix}, \tag{57}$$

where

$$c_{ab} = \frac{1}{k_a k_b \sigma_a^2 \sigma_b^2 - 1}. \tag{58}$$

The on-line dynamics is stable at the true solution $\mathbf{W}$, when $\mathbf{K} = (K_{A,B})$ is positive definite. Since it is decomposed in the two by two submatrices, it is positive definite when all the submatrices $K_{AA'}$ are positive definite. Hence, we have the following stability theorem.

Stability Theorem. Learning dynamics is stable when

$$\begin{aligned} k_a k_b \sigma_a^2 \sigma_b^2 &> 1 && (59)\\ k_a \sigma_b^2 + k_b \sigma_a^2 &> 0. && (60) \end{aligned}$$

The stability depends on the parameters k_a and σ_a^2, which are related to $\varphi$ and r.

5 Standardized Estimating Function and Newton's Method

For the learning dynamics

$$\Delta \mathbf{W}_t = \mathbf{W}_{t+1} - \mathbf{W}_t = -\eta \mathbf{F}(\mathbf{x}, \mathbf{W}_t) \tag{61}$$

Newton's method is given by

$$\Delta \mathbf{W}_t = -\eta \mathbf{K}(\mathbf{W}_t)^{-1} \mathbf{F}(\mathbf{x}, \mathbf{W}_t). \tag{62}$$

Hence, instead of a given estimating function $\mathbf{F}$, Newton's method is derived by using the estimating function

$$\mathbf{F}^*(\mathbf{x}, \mathbf{W}) = \mathbf{K}^{-1}(\mathbf{W})\mathbf{F}(\mathbf{x}, \mathbf{W}). \tag{63}$$

Its convergence is superlinear. Moreover, the true solution $\mathbf{W}$ is always stable, because the Hessian of $\mathbf{F}^*$ is the identity matrix. This is easily shown from

$$\mathbf{K}^* = E\left[\frac{\partial \mathbf{F}^*}{\partial \mathbf{X}}\right] = \frac{\partial \mathbf{K}^{-1}}{\partial \mathbf{X}} E[\mathbf{F}] + \mathbf{K}^{-1} \circ \mathbf{K} = \mathbf{I}. \tag{64}$$

We call $\mathbf{F}^*$ the standardized estimating function, for which $\mathbf{K}^*$ is the identity operator.

By using (56) or (57), the standardized estimating function matrix $\mathbf{F}^*$ is derived as

$$F^*_{ab} = c_{ab}\{k_b \sigma_a^2 \varphi(y_a) y_b - \varphi(y_b) y_a\}, \quad a \neq b. \tag{65}$$

6 Adaptive Approach to Newton's Method

The standardized estimating function $\mathbf{F}^*$ uses the parameters σ_a^2 and k_a, which are usually known, because they depend on the statistical properties of the source signal s_a. Therefore, an adaptive method is necessary to estimate them.

Let $k_{a,t}$ and $\sigma_{a,t}^2$ be their estimates at time $t$. Then, we can use the following adaptive rule to update them:

$$\begin{aligned} k_{a,t+1} &= \left(1-\varepsilon_{1,t}\right) k_{a,t} + \varepsilon_{1,t}\varphi_a'\left(y_a(t)\right), & (66)\\ \sigma_{a,t+1}^2 &= \left(1-\varepsilon_{2,t}\right)\sigma_{a,t}^2 + \varepsilon_{2,t} y_a^2(t), & (67)\end{aligned}$$

where $\varepsilon_{1,t}$ and $\varepsilon_{2,t}$ are learning rates.

We may use the diagonal term of $\mathbf{F}$ to be equal to

$$F_{aa} = y_a^2 - 1. \tag{68}$$

Then, the recovered signal is normalized to $\sigma_a^2 = 1$, so that $\mathbf{F}^*$ is simplified.

7 Error Analysis

Let us consider the estimation error in the case of batch estimator $\hat{\mathbf{W}}$ which is the solution of the estimating equation

$$\sum_{i=1}^{t} \mathbf{F}\left(\mathbf{x}(i), \mathbf{W}\right) = 0. \tag{69}$$

By using the standard method of statistical analysis, we can calculate the covariance of estimator $\hat{\mathbf{W}} = \mathbf{W} + \Delta\mathbf{W}$, where $\Delta\mathbf{W}$ is the error term. It is easier to calculate $E\left[\Delta\mathbf{X}\Delta\mathbf{X}\right]$ in terms of $\Delta\mathbf{X} = \Delta\mathbf{W}\mathbf{W}^{-1}$.

It should be noted that $\mathbf{F}$ and $\mathbf{RF}$ give the same error since the estimating equations are equivalent. There is a big difference in the case of online learning, where $\mathbf{F}$ and $\mathbf{RF}$ are different in convergence speed and stability. The covariance matrix is now calculated explicitly. To this end, we put

$$\begin{aligned} l_a &= E[\varphi_a(s_a)], & (70)\\ G_{ab,cd}^* &= E[F_{ab}^*(\mathbf{x},\mathbf{W})F_{cd}^*(\mathbf{x},\mathbf{W})] & (71)\end{aligned}$$

by using the standardized estimating function $\mathbf{F}^*$.

Theorem 2. The covariances of ΔX^t_{ab} are given as

$$E[\Delta X^t_{ab}\Delta X^t_{cd}] = \frac{1}{t}G^*_{ab,cd} + O\left(\frac{1}{t^2}\right), \tag{72}$$

where t is the number of observations. In particular,

$$\begin{aligned} &G^*_{ac,bc} = c_{ac}c_{bc}\sigma^2_a\sigma^2_b\sigma^2_c k^2_c l_a l_b, \\ &a \neq b,\ c \neq a,\ c \neq b. \end{aligned} \tag{73}$$

It is possible to evaluate the error by the covariance matrix of the error $\Delta\mathbf{s}$ in the recovered signals

$$\mathbf{y} = (\mathbf{W} + \Delta\mathbf{X}\mathbf{W})\mathbf{x} = \mathbf{y} + \Delta\mathbf{s}, \tag{74}$$

$$\Delta\mathbf{s} = \Delta\mathbf{X}\mathbf{s}. \tag{75}$$

Let us put

$$V_{ab} = E\left[\Delta s_a \Delta s_b\right]. \tag{76}$$

We first calculate, for $a \neq b$,

$$\begin{aligned} &E\left[y_a(t)y_b(t)\right] \\ &= E\left[\left\{s_a(t) + \sum_c \Delta X_{ac}s_c(t)\right\}\right. \\ &\quad \left.\left\{s_b(t) + \sum_d \Delta X_{bd}s_d(t)\right\}\right] \end{aligned} \tag{77}$$

$$= E\left[\Delta s_a \Delta s_b\right] = \sum_{c,d} E\left[\Delta X_{ac}\Delta X_{bd}s_c s_d\right] \tag{78}$$

$$= \sum_{c,d} E\left[\Delta X_{ac}\Delta X_{bd}\right] E[s_c s_d] \tag{79}$$

$$= \sum_c E\left[\Delta X_{ac}\Delta X_{bc}\right]\sigma^2_c. \tag{80}$$

Hence, we have

$$\begin{aligned} V^t_{ab} &= E[\Delta s_a \Delta s_b] = E[y_a(t)y_b(t)] \\ &= \sum_c E[\Delta X^t_{ac}\Delta X^t_{bc}]\sigma^2_c. \end{aligned} \tag{81}$$

Theorem 3. The covariance matrix $\mathbf{V}_t$ of $\Delta\mathbf{s}$ is given by

$$V^t_{ab} = \frac{1}{t}\sum_c G^*_{ac,bc}\sigma^2_c + O\left(\frac{1}{t^2}\right), \qquad (a \neq b). \tag{82}$$

The lemma shows that the the covariances $V^t_{ab} = E[\Delta s_a \Delta s_b] = E[y_a y_b]$ of the recovered signals y_a and y_b $(a \neq b)$ decrease in the order of $1/t$. This fact agrees with ordinary asymptotic statistical analysis, as is expected. However, we can prove superefficiency [2] implying that the covariance of any two recovered signals decreases in the order of $1/t^2$ under a certain condition.

Theorem 4. A batch estimator is superefficient when $E\left[\varphi_a\left(s_a\right)\right] = 0$.

Proof. In this case, we have $l_a = 0$, so that V^t_{ab} $(a \neq b)$ satisfies

$$V^t_{ab} = O\left(\frac{1}{t^2}\right). \tag{83}$$

8 Adaptive Choice of $\varphi$ Function

The estimation error depends on the choice of $\mathbf{F}^*(\mathbf{x}, \mathbf{W})$, or the functions $\boldsymbol{\varphi}$ in the form of (24). Note that $\mathbf{F}$ and its standardized form $\mathbf{F}^*$ have the same asymptotic error for the batch mode estimation. However, their dynamical behaviors are different in on-line learning or iterated batch algorithms. The standardized one coincides with Newton's method and its convergence is superlinear. Hence, the adaptive choice of $\mathbf{F}^*$ in (66), (67) guarantees a good convergence, but its error still depends on $\varphi$.

In order to improve the error, an adaptive choice of $\boldsymbol{\varphi}$ is useful. An adaptive choice of $\varphi$ is also useful for guaranteeing stability, when we do not use Newton's method. When $\boldsymbol{\varphi}$ is derived from the true probability distributions of the sources, the estimated $\hat{\mathbf{W}}$ is mle, and is efficient in the sense that the asymptotic error is minimal and is equal to the inverse of the Fisher information matrix. However, it is highly costful to estimate the probability density functions of the sources. Instead, we use a parametric family of $\boldsymbol{\varphi}$,

$$\varphi_a = \varphi_a\left(y; \boldsymbol{\xi}_a\right), \tag{84}$$

for each source s_a and update the parameter $\boldsymbol{\xi}_a$ by

$$\Delta \boldsymbol{\xi}_a = -\eta' \frac{\partial l}{\partial \boldsymbol{\xi}_a}. \tag{85}$$

The Gaussian mixture was proposed for approximating the source probability density, which is the parametric family

$$q(y; \boldsymbol{\xi}) = \sum v_i \exp\left\{-\frac{(x-\mu_i)^2}{2\sigma_i^2}\right\}, \tag{86}$$

$\boldsymbol{\xi}$ consisting of a number of v_i, μ_i and σ_i^2. The corresponding parametric $\varphi(y;\boldsymbol{\xi})$ is derived therefrom. This covers both sub-Gaussian and super-Gaussian distributions. However, this family is computationally costful. Zhang et al [23] proposed an exponential family connecting three typical distributions; Gaussian, super-Gaussian and sub-Gaussian.

Let us use the following exponential family of distributions

$$q_a\left(s,\boldsymbol{\theta}_a\right) = \exp\left\{\boldsymbol{\theta}_a \cdot \mathbf{g}(s) - \psi\left(\boldsymbol{\theta}_a\right)\right\}, \tag{87}$$

where $\boldsymbol{\theta}_a$ is the canonical parameters, $\mathbf{g}(s)$ is an adequate vector function and $\psi$ is the normalization factor. The function φ_a is derived as

$$\varphi_a(y) = -\frac{d}{dy}\log q_a\left(y,\boldsymbol{\theta}_a\right) = \boldsymbol{\theta}_a \cdot \mathbf{g}'(y). \tag{88}$$

Zhang et al [23] proposed the three-dimensional model,

$$\mathbf{g}(y) = \left(\log \operatorname{sesh}(y), -y^4, -y^2\right)^T \tag{89}$$

or

$$\mathbf{g}'(y) = \left(\tanh(y), y^3, y\right)^T, \tag{90}$$

of which components correspond to the typical φ' s proposed so far. They are responsible for the super-Gaussian, sub-Gaussian and linear cases, respectively. The $\varphi_a(y)$ is their linear combination, covering all the cases. The parameter $\boldsymbol{\theta}_a$ is adaptively determined as

$$\boldsymbol{\theta}_{a,t+1} = \boldsymbol{\theta}_{a,t} - \varepsilon_t\left\{\mathbf{g}\left(y_a(t)\right) - E\left[\mathbf{g}\left(y_a\right)\right]\right\}, \tag{91}$$

where $E\left[\mathbf{g}\left(y_a\right)\right]$ may be adaptively estimated.

9 Estimating Functions in Noisy Case

Let us analyze the noisy case

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \boldsymbol{\nu}, \tag{92}$$

where $\boldsymbol{\nu}$ is a noise vector in the measurement. We assume that $\boldsymbol{\nu}$ is Gaussian and that their components are uncorrelated. Let

$$\Sigma = \operatorname{diag}\left(\sigma_1^2, \cdots, \sigma_n^2\right) \tag{93}$$

be its covariance matrix. In order to fix the scale, we also assume

$$E\left[s_i^2\right] = 1. \tag{94}$$

Let $\mathbf{W} = \mathbf{H}^{-1}$ be the true unmixing matrix, and put

$$\mathbf{y} = \mathbf{W}\mathbf{x}. \tag{95}$$

Then, we have

$$\mathbf{y} = \mathbf{s} + \mathbf{W}\boldsymbol{\nu} = \mathbf{s} + \boldsymbol{\mu}, \tag{96}$$

where $\boldsymbol{\mu} = \mathbf{W}\boldsymbol{\nu}$ is a noise vector whose components are correlated.

In the noisy case, functions of the type

$$\mathbf{F} = I - \varphi(\mathbf{y})\mathbf{y}^T \tag{97}$$

are not in general estimating functions. Indeed,

$$E\left[I - \varphi(\mathbf{y})\mathbf{y}^T\right] \neq 0 \tag{98}$$

even when $\mathbf{y}$ is derived from the true $\mathbf{W}$, because y_i and y_j are no more independent even when $\mathbf{W} = \mathbf{H}^{-1}$. However, estimating functions exist in the noisy case. Kawanabe and Murata [19] studied a family of estimating functions in the noisy case.

For the true $\mathbf{W} = \mathbf{H}^{-1}$, the noise term is

$$\boldsymbol{\mu} = \mathbf{W}\boldsymbol{\nu}, \tag{99}$$

which is Gaussian. Let its covariance matrix be

$$\mathbf{V} = E\left[\boldsymbol{\mu}\boldsymbol{\mu}^T\right] = E\left[\mathbf{W}\boldsymbol{\nu}\boldsymbol{\nu}^T\mathbf{W}^T\right] = \mathbf{W}\Sigma\mathbf{W}^T. \tag{100}$$

Kawanabe and Murata [19] calculated all possible estimating functions. The following is a simplest estimating function $\mathbf{F}(\mathbf{y}, \mathbf{W})$ whose components are

$$F_{ab}(\mathbf{y}, \mathbf{W}) = y_a^3 y_b - 3v_{aa}y_a y_b - 3v_{ab}y_a^2 + 3v_{aa}v_{ab}, \tag{101}$$

where v_{ab} are elements of $\mathbf{V}$. We can easily prove that

$$E\left[\mathbf{F}(\mathbf{y}, \mathbf{W})\right] = 0 \tag{102}$$

when $\mathbf{W} = \mathbf{H}^{-1}$. Hence,

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \eta_t \mathbf{F}\left(\mathbf{y}(t), \mathbf{W}_t\right) \mathbf{W}_t \tag{103}$$

is a learning algorithm effective even under large noise.

When the covariance matrix $\sum$ of the measurement noise is unknown, we need to estimate it. Factor analysis provides a method of estimating it (Ikeda and Toyama [18]). The off-diagonal term can be adaptively estimated from

$$v_{ab}^{t+1} = (1 - \varepsilon_t)\, v_{ab}^t + \varepsilon_t y_a(t) y_b(t), \tag{104}$$

where ε_t is a learning rate.

The learning algorithm (103) is not necessarily stable. A stable algorithm is given by the standardized estimating function $\mathbf{F}^*$, which is Newton's method. We can obtain $\mathbf{F}^*$ explicitly in a method similar to the noiseless case.

10 Conclusions

There have been proposed a number of algorithms for extracting independent components from mixtured signals. The method is in general called Independent Component Analysis, and applied to blind source separation or extraction. Most of the algorithms use estimating functions implicitly or explicitly.

The present paper gives a unified approach to ICA based on estimating functions. The stability analysis and Newton's method is derived from estimating functions. Estimation error is also analyzed in terms of estimating functions. Based on these analyses, a new method of ICA is proposed which uses an adaptive choice of estimating functions.

We have analyzed only the case of instantaneous mixtures, but the same method is applicable to the case where independent sources have (unknown) temporal correlations [4] and to the case of convoluted mixture signals [9].

References

[1] S. Amari,“Natural gradient works efficiently in learning”, Neural Computation, vol. 10, pp. 251–276, 1998.

[2] S. Amari, "Supereffiency in blind source separation", IEEE Trans. Signal Processing, vol. 47, no. 4, pp. 936–944, 1999.

[3] S. Amari, "Natural gradient for over- and under-complete ICA", Neural Computation, vol. 11, pp. 1875–1883, 1999.

[4] S. Amari, "Estimating functions of independent component analysis for temporally correlated signals", Neural Computation, vol. 12, pp. 2083–2107, 2000.

[5] S. Amari, and J.F. Cardoso, "Blind source separation — Semi-parametric statistical approach", IEEE Trans. on Signal Processing, vol. 45, no. 11, pp. 2692–2700, 1997.

[6] S. Amari, T.-P. Chen, and A. Cichocki, "Stability analysis of adaptive blind source separation", Neural Networks, vol. 10, no. 8, pp. 1345–1351, 1997.

[7] S. Amari, and A. Cichocki, "Adaptive blind signal processing — neural network approach", Proceedings of IEEE, vol. 86, pp. 2026–2048, 1998.

[8] S. Amari, A. Cichocki, and H.H. Yang, "A new learning algorithm for blind signal separation", Advances in Neural Information Processing Systems 8, (NIPS-1995) MIT Press, Boston, pp. 752–763, 1996.

[9] S. Amari, S.C. Douglas, and A. Cichocki, "Multichannel blind deconvolution and source separation using the natural gradient", unpublished.

[10] S. Amari, and M. Kawanabe, "Information geometry of estimating functions in semi-parametric statistical models", Bernoulli, vol. 3, no. 1, pp. 29–54, 1997.

[11] S. Amari, and H. Nagaoka, "Introduction to Information Geometry", AMS and Oxford University Press, 1999.

[12] A.J. Bell, and T.J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution", Neural Computation, vol. 7, pp. 1129–1159, 1995.

[13] J.-F. Cardoso, and B. Laheld, "Equivariant adaptive source separation", IEEE Trans. Signal Processing, SP-43, pp. 3017–3029, 1996.

[14] P. Comon, "Independent component analysis: a new concept?", Signal Processing, vol. 36, pp. 287–314, 1994.

[15] M. Girolami, Self-organizing neural networks - Independent component analysis and blind source separation, Springer-Verlag, 1999.

[16] C. Jutten, and J. Herault, "Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture", Signal Processing, vol. 24, pp. 1–20, 1991.

[17] A. Hivärinen, J. Karhunen, and E. Oja, Independent component analysis, John Willy and Sons, New York, 2001.

[18] S. Ikeda and K. Toyama, "Independent component analysis for noisy data–MEG data analysis", Neural Networks, vol. 13, no. 10, pp. 1063–1074, 2001.

[19] M. Kawanabe and N. Murata, "Independent component analysis in the presence of Gaussian noise based on estimating functions", submitted.

[20] T.-W. Lee, Independent component analysis - Theory and applications, Kluwer, 1998.

[21] E. Oja, "The nonlinear PCA learning rule in independent component analysis", Neurocomputing, vol. 17, pp. 25–45, 1997.

[22] D.T. Pham, and P. Garat, "Blind separation of mixture of independent sources through a quasi-maximum likelihood approach", IEEE Trans. Signal Processing, vol. 45, pp. 1457–1482, 1997.

[23] L.-Q. Zhang, S. amari, and A. Cichocki, "Equi-convergence algorithm for blind separation of sources with arbitrary distributions", IWANN 2001, LNCS 2085, eds. J. Mira and A. Prieto, pp. 826–833, 2001.