LOGSPLINE INDEPENDENT COMPONENT ANALYSIS

Kawaguchi, Atsushi
Biostatistics Center, Kurume University

Truong, Young K.
Department of Biostatistics, The University North Carolina at Chapel Hill

https://doi.org/10.5109/1434313

出版情報：Bulletin of informatics and cybernetics. 43, pp.83-94, 2011-12. 統計科学研究会
バージョン：
権利関係：

LOGSPLINE INDEPENDENT COMPONENT ANALYSIS

by

Atsushi Kawaguchi and **Young K. Truong**

Reprinted from the Bulletin of Informatics and Cybernetics
Research Association of Statistical Sciences, Vol.43

FUKUOKA, JAPAN
2011

Bulletin of Informatics and Cybernetics, Vol. 43, 2011

LOGSPLINE INDEPENDENT COMPONENT ANALYSIS

By

Atsushi Kawaguchi[*] and **Young K. Truong**[†]

Abstract

Most recent maximum likelihood approaches to independent component analysis (ICA) are based on nonparametric density estimation. In this paper, we present an algorithm by using the logsplines approach to density estimation. The logarithmic source density functions are modeled by polynomial splines or a linear combination of B-splines with (a) parameters or coefficients of the B-splines estimated by maximizing the log-likelihood function, and (b) knots of the B-splines determined by a stepwise procedure so as to minimize the approximation errors in modeling the log-density functions. We showed in a comparative study that our new algorithm has performed very favorably when compared to several popular density estimation based procedures.

Key Words and Phrases: Blind Source Separation, Independent Component Analysis, Logspline Density Estimation, Maximum Likelihood Estimation

1. Introduction

Independent component analysis (ICA) is a useful tool for extracting important information from data. In this context, the data, or large amounts of data contain relatively little relevant information. That is, the data depend on a relatively small number of driving or causal factors, or simply referred to as sources. The objective of ICA is to identify these sources, and having done so, it would then be possible to estimate the extent to which the observed data depended on each source, so that further or finer classification of the data can be carried out. There are indeed many applications of ICA. For example, ICA has been used extensively in blind source separation for image and acoustic signal processing (Hyvärinen et al. (2001), Cichocki and Amari (2002)); medical research such as functional magnetic resonance imaging (fMRI) analysis (McKeown and Sejnowski (1998), Calhoun et al. (2003) and Stone (2004)). A wide variety of recent applications can also be found in Comon and Jutten (2010) and Hyvärinen (2011).

In general, the solution of ICA is obtained by optimizing an appropriate contrast function. The most common contrast functions are based on maximum likelihood, the infomax principle and mutual information. Amari et al. (1996) proposed the natural gradient based ICA algorithm. The FastICA algorithm (Hyvärinen and Oja (1997)) uses a deflation scheme to compute components sequentially. For each component an

[*] Biostatistics Center, Kurume University, 67 Asahi-machi, Kurume-shi, Fukuoka 830-0011 Japan.

[†] Department of Biostatistics, The University North Carolina at Chapel Hill, Chapel Hill, NC 27599-7420 U.S.A. truong@bios.unc.edu

one-unit contrast function, based on an approximation to the negentropy (negative-entropy) of a component, is maximized. This function can be viewed as a measure of nongaussianity. The JADE algorithm (Cardoso (1999)) is a cumulant-based method that uses joint diagonalization of a set of fourth-order cumulant matrices. It uses algebraic properties of fourth-order cumulants to define a contrast function that is minimized using Jacobi rotations. The extended Infomax algorithm (Lee et al. (1999)) is a variation on the Infomax algorithm (Bell and Sejnowski (1995)) that can deal with either subgaussian or supergaussian components, by adaptively switching between two nonlinearities.

Alternative methods that employ a more flexible model for the probability density functions of the source signals have been introduced (Pham et al. (1992)). Hastie and Tibshirani (2002) proposed maximizing the penalized likelihood estimation. Bach and Jordan (2002) proposed the algorithm based on canonical correlation in a reproducing kernel Hilbert space. Miller and Fisher (2003) proposed the RADICAL algorithm based on the neighborhood density estimator. Vlassis and Motomura (2001), Boscolo et al. (2004) and Chen (2006) used kernel density estimation.

In this paper, we describe an ICA algorithm using polynomial splines (Stone et al. (1997)). The method will overcome two key issues arising from using traditional ICA algorithms. First, misspecification of the source density function may have a serious bias problem and subsequently, not only ICA will perform poorly (Cardoso (1998)), the results will be difficult to interpret. This bias issue can be easily addressed by the flexibility of spline functions. In fact, the polynomial spline approach allows us to estimate any (marginal) distribution of the unknown signal without specifying its functional form a priori. Second, the contrast functions employed usually have multiple local minima (Amari et al. (1996)). This can be effectively addressed by employing multiple initial values at the onset of the algorithm for logspline density estimation. We illustrate its usefulness by comparing several recently developed density estimation based algorithms to ICA. The numerical results indicated that our procedure has an improvement over its peers in estimating the mixing matrix. See Section 3 for a detailed discussion of these results.

The remainder of the paper is organized as follows. The proposed method is described in Section 2. In Section 3, an extensive simulation study is presented. Concluding remarks are given in Section 4.

2. Methods

We start by giving a detailed description of the general setting of ICA.

2.1. Modeling Assumptions for ICA

Given an observable p-dimensional random vector $\mathbf{X}$, it is assumed that there exist a p by p mixing matrix $\mathbf{A}$ and a random vector $\mathbf{S} = (S_1, \ldots, S_p)$ of independently distributed components S_j, $j = 1, 2, \ldots, p$ such that

$$\mathbf{X} = \mathbf{A}\mathbf{S}.$$

This is called the ICA model. Note that $\mathbf{S}$ is not observable and is known as the latent source, and $\mathbf{A}$ is called the mixing matrix.

The main statistical problem now is to unmix $\mathbf{X}$ by estimating $\mathbf{A}$ and $\mathbf{S}$. Suppose each S_j has a density function f_j, for $j = 1, 2, \ldots, p$. Then the density function of $\mathbf{X}$

can be expressed as

$$f_{\mathbf{X}}(\mathbf{x}) = \mathsf{det}(\mathbf{W}) \prod_{j=1}^{p} f_j(\mathbf{w}_j^T \mathbf{x}), \tag{1}$$

where $\mathbf{W} = \mathbf{A}^{-1}$ and $\mathbf{w}_j$ is the j-th row of $\mathbf{W}$. The justification of the existence of $\mathbf{W}$ will be given shortly.

The logarithm of each source density is modeled by using polynomial splines

$$\log(f_j(x)) = C(\boldsymbol{\beta}_j) + \beta_{01j}x + \sum_{i=1}^{m_j} \beta_{1ij}(x - r_{ij})_+^3, \quad (j = 1, 2, \ldots, p)$$

where $\boldsymbol{\beta}_j = (\beta_{j01},\ \beta_{j11},\ \ldots,\ \beta_{j1m_j})$ is a vector of coefficients, $C(\boldsymbol{\beta}_j)$ is a normalized constant, r_{ij} are the knots, and $(a)_+ = \max(a, 0)$. See Stone et al. (1997) for a general discussion of modeling the log density function and the methodology called logspline density estimation.

2.2. Estimation

It follows from the last section that the problem of estimating the mixing matrix $\mathbf{A}$ and the source $\mathbf{S}$ is equivalent to estimating $\mathbf{W}$ and $\boldsymbol{\beta}_j, j = 1, \ldots, p$. We now justify the existence of $\mathbf{W}$. In fact, the ICA algorithm can be simplified by first centering and prewhitening the data so that $\mathsf{Cov}(\mathbf{KX}) = \mathbf{I}$, where $\mathbf{K}$ is obtained by applying principal component analysis (PCA) to $\mathbf{X}'\mathbf{X}$. The algorithm then seeks a matrix $\mathbf{W}$ such that $\mathbf{WKX} = \mathbf{S}$. Here $\mathbf{W}$ is called the *unmixing matrix* and it is orthogonal (Hyvärinen et al. (2001)). Since this implies that the determinant in (1) is 1, the computation is significantly simplified. This prewhitening technique can also reduce dimension of $\mathbf{X}$ by invoking the first few columns in $\mathbf{K}$, which are usually used in the fMRI study (Calhoun et al. (2003)).

Let $\mathbf{X}_1,\ \ldots,\ \mathbf{X}_n$ be independent random variables having the same distribution as $\mathbf{X}$. Let denote the vector of parameters in the density function by $\boldsymbol{\theta} = (\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_p)$. We obtain the estimate of $(\mathbf{W},\ \boldsymbol{\theta})$ by maximizing the likelihood function $\ell(\mathbf{W}, \boldsymbol{\theta})$ with respect to $(\mathbf{W},\ \boldsymbol{\theta})$. The likelihood function is given by

$$\ell(\mathbf{W}, \boldsymbol{\theta}) = \sum_{i=1}^{n}\sum_{j=1}^{p} \log(f_j(\mathbf{w}_j^T \mathbf{X}_i)) = \sum_{j=1}^{p} \ell_j(\mathbf{W}, \boldsymbol{\theta}), \tag{2}$$

where $\ell_j(\mathbf{W}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \log(f_j(\mathbf{w}_j^T \mathbf{x}_i))$, which is the likelihood function for each density. Direct implementation of MLE is not feasible in general, one uses the profile likelihood procedure. Thus, we apply the iterative algorithm shown in Table 2. We call this methodology for ICA the logspline ICA, and abbreviate it as LICA.

2.3. Algorithm

The proposed LICA algorithm will now be described. Our algorithm comprises two parts hierarchically, the local maximization for $\mathbf{W}$ (improved upon the initial guess) and the global maximization for $g_j = \log f_j$ using the locally optimized $\mathbf{W}$, coupled with several initial guesses to yield the final density estimates. More detailed descriptions of the local and global maximization algorithms can be found in Tables 1 and 2, respectively. Note that the Amari metric (Amari et al. (1996)) used in the algorithms is defined by

$$d(\mathbf{P},\mathbf{Q}) = \frac{1}{p(p-1)}\left\{\sum_{i=1}^{p}\left(\frac{\sum_{j=1}^{p}|a_{ij}|}{\max_j |a_{ij}|}-1\right)+\sum_{j=1}^{p}\left(\frac{\sum_{i=1}^{p}|a_{ij}|}{\max_i |a_{ij}|}-1\right)\right\},$$

where $a_{ij} = (\mathbf{P}^{-1}\mathbf{Q})_{ij}$, $\mathbf{P}$ and $\mathbf{Q}$ are $p \times p$ matrics. This metric is normalized, is between 0 and 1.

Table 1: Algorithm (Local Maximization)

1. Initialize $\mathbf{W}$.
2. Alternate until convergence of $\mathbf{W}$, using the Amari metric.
   - (a) Given $\mathbf{W}$, estimate the log density g_j for jth element X_j of the prewhitened data $\mathbf{X}$ (separately for each j), using logspline density estimation below (see Section 2.4.).
   - (b) Given g_j $(j = 1, 2, \ldots, p)$,

     $$\mathbf{w}_j \leftarrow E[\mathbf{X}{g_j}'(\mathbf{w}_j^T\mathbf{X})] - E[{g_j}''(\mathbf{w}_j^T\mathbf{X})]\mathbf{w}_j$$

     where $\mathbf{w}_j$ is the jth row of $\mathbf{W}$.
   - (c) Orthogonalize $\mathbf{W}$

Table 2: Algorithm (Global Maximization)

1. Setting 10 initial matrices as follows.
   - (a) Generating 10000 $p \times p$ orthogonal matrices by normal random number
   - (b) Selecting 10 matrices which have 50th, 150th, 250th, ..., 950th largest Amari metrics from an identity matrix.
2. For each initial matrix, estimating the local maximized $\hat{\mathbf{W}}^{(i)}$ using the algorithm described in Table 2.
3. Output $\hat{\mathbf{W}}_{glob}$ such that

   $$\hat{\mathbf{W}}_{glob} = \text{argmax}_{\hat{\mathbf{W}}^{(i)}} C(\hat{\mathbf{W}}^{(i)}).$$

We now describe the logspline density estimation methodology.

2.4. Density Estimation

Let X be a random variable having a continuous and positive density function. In general, the log density of X is modeled by

$$g(x) = \log(f(x)) = C(\boldsymbol{\beta}) + \beta_{01}x + \sum_{j=1}^{m} \beta_{1j}(x - r_j)_+^3,$$

where $\boldsymbol{\beta} = (\beta_{01}, \beta_{11}, \ldots, \beta_{1m})$ is a vector of coefficients, $C(\boldsymbol{\beta})$ is a normalizing constant, r_j are the knots, m is the number of knots, and $(a)_+ = \max(a, 0)$. Let $X_1, \ldots, X_n$ be independent random variables having the same distribution as X. The log-likelihood function corresponding to the logspline family is given by $\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} g(X_i)$, which corresponds to $\ell_j(\mathbf{W}, \boldsymbol{\theta})$ in (2). The maximum likelihood estimate $\hat{\boldsymbol{\beta}}$ is obtained by maximizing the log-likelihood function.

The knot selection methodology involves initial knot placement, involves stepwise knot addition, stepwise knot deletion and final model selection based on the information criterion. We set the initial knot placement to be minimum, median and maximum values of the distribution of data. At each addition step, we first find a good location for a new knot in each of the intervals (L, r_1), (r_1, r_2), $\ldots$, (r_{K-1}, r_K), (r_K, U) determined by the existing knots $r_1, r_2, \ldots, r_K$ and some constants L and U. Let $X_{(1)}, \ldots, X_{(n)}$ be the data written in nondecreasing order. Set $l_1 = 0$ and $u_K = n$. Define l_i and u_i by

$$l_i = d_{\min} + \max\{j : 1 \le j \le n \text{ and } X_{(j)} \le r_i\}, \quad i = 2, \ldots, K$$

and

$$u_i = -d_{\min} + \max\{j : 1 \le j \le n \text{ and } X_{(j)} \ge r_i\}, \quad i = 1, \ldots, K-1,$$

where $d_{\min}$ is the minimum distance between consecutive knots in order statistics.

For $i = 0, \ldots, K$ and for the model with X_{j_i} as a new knot where $j_i = [(l_i + u_i)/2]$ with $[x]$ being the integer part of x, we compute the Rao statistics R_i defined by

$$R_i = \frac{[\boldsymbol{S}(\hat{\boldsymbol{\beta}})]_i}{\sqrt{[\boldsymbol{I}^{-1}(\hat{\boldsymbol{\beta}})]_{ii}}},$$

where $\boldsymbol{S}(\hat{\boldsymbol{\beta}})$ is the score function; that is, the vector with entries $\partial\ell(\hat{\boldsymbol{\beta}})/\partial\beta_j$, and $\boldsymbol{I}(\hat{\boldsymbol{\beta}})$ is the matrix whose entry in row j and column k is given by $-\partial^2\ell(\hat{\boldsymbol{\beta}})/\partial\beta_j\partial\beta_k$. We place the potential new knot in the interval $[X_{l_{i^*}}, X_{u_{i^*}}]$ where $i^* = \operatorname{argmax} R_i$. Within this interval we further optimize the location of the new knot. To do this, we proceed by computing the Rao statistics R_l for the model with $X_{(l)}$ as the knot with $l = [(l_{i^*} + j_{i^*})/2]$ and R_u for the model with $X_{(u)}$ as the knot with $u = [(j_{i^*} + u_{i^*})/2]$. If $R_{i^*} \ge R_l$ and $R_{i^*} \ge R_u$, we place the new knot at $X_{(i^*)}$; If $R_{i^*} < R_l$ and $R_l \ge R_u$, we continue searching for a knot location in the interval $[X_{(l_{i^*})}, X_{(j_{i^*})}]$; If $R_{i^*} < R_u$ and $R_l < R_u$, we continue searching for a knot location in the interval $[X_{(j_{i^*})}, X_{(u_{i^*})}]$.

After a maximum number of knots $K_{\max} = \min(4n^{1/5}, n/4, N, 30)$ where N is the number of distinct X_i's, we continue with stepwise knot deletion. During knot deletion we successively remove the knot which has minimum Wald statistics defined by

$$W_i = \frac{\hat{\beta}_i}{\sqrt{[\boldsymbol{I}^{-1}(\hat{\boldsymbol{\beta}})]_{ii}}}$$

among the existing knots.

Among all models that are fit during the sequence of knot addition and knot deletion we choose the model that minimizes Bayesian information criterion (BIC) defined by

$$\begin{aligned}\mathrm{BIC} &= -2\ell(\hat{\boldsymbol{\beta}}) + m\log(n)\\ &= -2\sum_{i=1}^{n}\left\{C(\hat{\boldsymbol{\beta}}) + \hat{\beta}_{01}X_i + \sum_{j=1}^{m}\hat{\beta}_{1j}(X_i - r_j)_+^3\right\} + m\log(n)\end{aligned}$$

where m is the number of parameters (knots) to be selected and $\hat{\boldsymbol{\beta}} = (\hat{\beta}_{01}, \hat{\beta}_{11}, \ldots, \hat{\beta}_{1m})$ is the maximum likelihood estimator vector of coefficients.

3. Simulation Study

We followed Bach and Jordan (2002) to design our comparative study. Specifically, we used the 18 distributions depicted in Figure 1 and the true mixing matrix chosen at random with bounded condition number between 1 and 2 to simulate our data. The leading ICA algorithms to be compared include FastICA: fICA (Hyvärinen and Oja (1997)), the JADE algorithm (Cardoso (1999)), the extended infomax algorithm: exinfo (Lee et al. (1999)), one of two versions of KernelICA: KGV (Bach and Jordan (2002)), the RADICAL algorithm: RAD (Miller and Fisher (2003)), the NPICA algorithm (Boscolo et al. (2004)), the KDICA (Chen (2006)), the ProDenICA: PDICA (Hastie and Tibshirani (2002)). Since our finding is consistent with Bach and Jordan (2002) that the resulting Amari metric of KGV is smaller than that of KCCA, we picked KGV over KCCA for the comparative study. For KGV, we followed the initialization technique described in Bach and Jordan (2002). For PDICA, we used the default setting in the ProDenICA function of the R package ProDenICA except for the `restarts` argument, which was set to 5, after Hastie and Tibshirani (2002). The other algorithms were used with the default setting of initial guesses. The data was prewhitened before the algorithm was applied. We measured the difference between the true matrix $\mathbf{A}$ and the estimated $\mathbf{W}$ by using the Amari metric after the both metrics were adequately orthogonalized.

The first study was carried out for 2 components. Here the sample size or the number of input points was $N = 250$, and there were 100 replications of each experiment. The results are given in Table 3. The first column indicates the pair of source density functions taken from Figure 1. The remaining columns are the mean (over replications) Amari metric values of the methods being compared. The average of these mean values are given in the row labeled as MEAN. The bottom row RAND of the table shows the average over replications for which two sources were chosen at random among the 18 densities. KGV, KDICA, and PDICA had the smallest average in the MEAN row. KGV also had the smallest average in the RAND row. LICA performed better than the other algorithms except for KGV, KDICA, and PDICA, which were (especially for KGV) only marginally better than our algorithm. NPICA seemed to have the largest average, but when the initialization employed by KGV was used, the value became smaller. This means that the accuracy of this algorithm depends crucially on the setting of initial values. For random pairings, the top two were KGV and LICA.

The numerical results for $N = 1000$ are shown in Table 4. The smaller mean Amari metric values are clearly the effect of the larger sample size N. Our LICA procedure

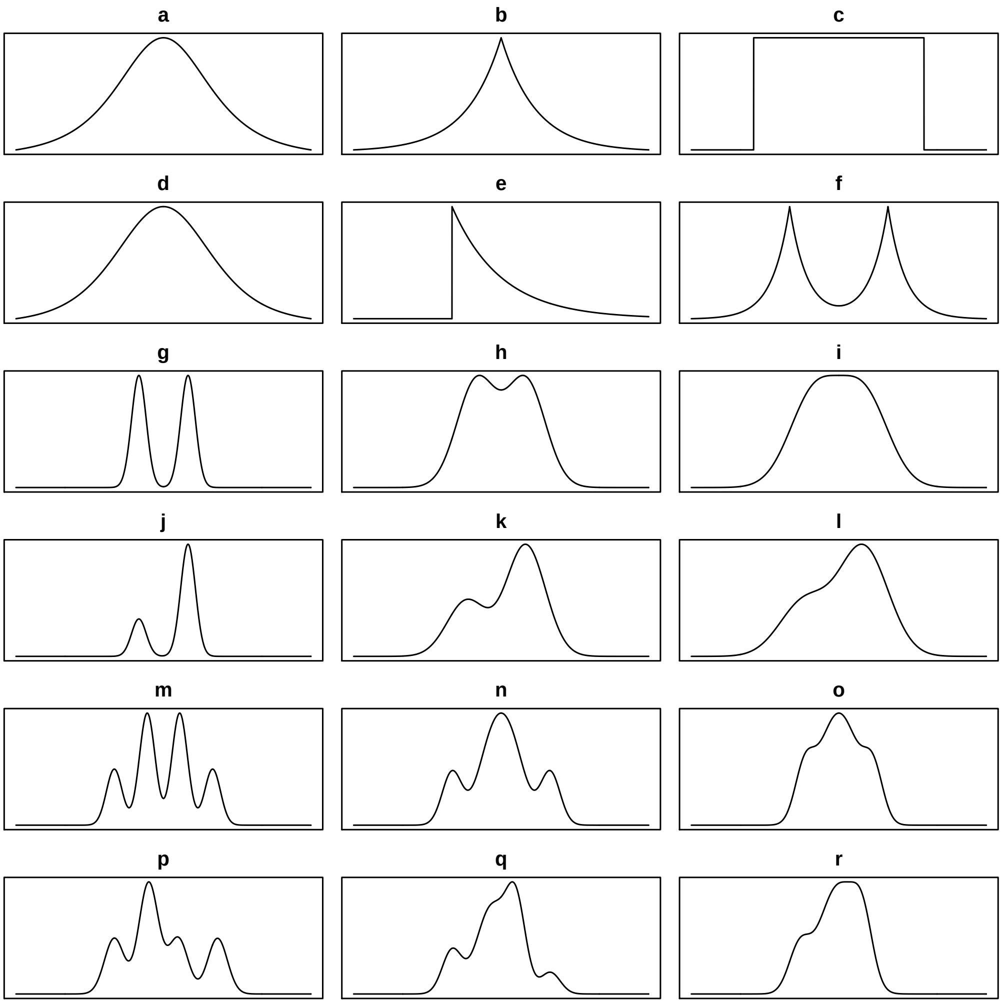

Figure 1: Probability density functions of sources with their kurtoses: (a) Student t with three degrees of freedom; (b) double exponential; (c) uniform; (d) Student with five degrees of freedom; (e) exponential; (f) mixture of two double exponentials; (g)-(h)-(i) symmetric mixtures of two Gaussians: multimodal, transitional and unimodal; (m)-(n)-(o) symmetric mixtures of four Gaussians: multimodal, transitional and unimodal; (p)-(q)-(r) nonsymmetric mixtures of four Gaussians: multimodal, transitional and unimodal. The Matlab code to sample from these distributions was provided from the website of Francis Bach, UC Berkeley. We translated it into R.

Table 3: Amari metric (multiplied by 100) for two-component with 250 samples. For each probability density function (from a to r), averages over 100 replicates are presented. The overall mean is calculated in the row labeled as MEAN. The RAND row presents the average over 1000 replications when two (generally different) densities were chosen randomly among the 18 possible densities. The optimal values are highlighted with bold face font.

Density	fICA	JADE	exinfo	KGV	RAD	NPICA	KDICA	PDICA	LICA
a	7.69	6.43	13.22	7.04	5.99	10.39	4.84	**4.82**	5.04
b	7.08	10.87	10.12	5.91	6.45	15.67	5.87	**5.51**	5.69
c	3.75	**2.34**	10.83	2.93	5.00	10.49	2.47	3.76	2.52
d	**10.13**	12.27	10.54	15.36	13.96	28.13	11.53	11.8	12.33
e	9.78	9.65	11.03	1.84	2.15	1.66	**1.58**	3.26	1.95
f	4.52	4.89	9.18	1.57	2.05	76.04	**1.49**	3.08	1.70
g	2.17	1.45	11.30	1.20	1.93	91.69	**1.19**	4.43	1.29
h	11.71	**8.58**	11.40	10.18	21.21	23.91	11.17	9.18	14.53
i	20.51	15.25	**11.41**	25.48	38.98	41.99	18.49	23.91	20.94
j	50.65	12.60	11.67	**1.29**	2.24	77.50	2.22	3.13	1.34
k	13.93	10.10	11.71	**4.96**	8.00	9.11	5.64	5.36	5.78
l	23.37	19.71	11.71	9.74	20.08	21.16	15.3	**9.52**	14.7
m	10.30	6.47	11.60	2.64	2.21	17.95	**1.50**	2.73	1.72
n	26.21	10.64	11.64	4.89	7.61	25.92	4.78	**3.99**	4.43
o	10.43	**7.26**	11.57	9.11	21.08	19.24	8.03	9.75	10.86
p	11.70	5.69	11.70	3.16	3.04	4.03	**2.18**	3.63	2.65
q	43.09	25.96	11.67	**5.64**	14.25	26.97	11.81	5.91	7.70
r	10.57	9.80	11.55	7.53	13.70	14.47	8.87	**7.36**	10.07
MEAN	15.42	10.00	11.33	6.69	10.55	28.68	**6.61**	6.73	6.95
RAND	11.85	9.11	19.43	**4.98**	7.76	11.48	5.30	5.89	5.11

had the smallest average on both rows MEAN and RAND. Moreover, it had the most reduction as N increased from 250 to 1,000. For instance, in identical pairing, the percent or relative reductions of KGV, KDICA and LICA were 57%, 60% and 64%, respectively. In random pairing, the relative reductions of KGV, KDICA and LICA were 58%, 62% and 75%, respectively. Plots of the percent reduction are shown Figure 2. To achieve the best effect for plotting, only the best of five or six methods are selected in these figures. These results are useful for an informal assessment of the efficiency among the ICA procedures being considered for comparison.

Table 4: Amari metric (multiplied by 100) for two-component with 1000 samples. For each probability density function (from a to r), averages over 100 replicates are presented. The overall mean is calculated in the row labeled as MEAN. The RAND row presents the average over 1000 replications when two (generally different) densities were chosen randomly among the 18 possible densities. The optimal values are highlighted with bold face font.

Density	fICA	JADE	exinfo	KGV	RAD	NPICA	KDICA	PDICA	LICA
a	2.96	3.89	4.12	2.70	3.92	9.22	2.59	2.43	**2.36**
b	5.23	4.63	11.09	2.76	3.16	6.42	**2.23**	2.35	2.40
c	4.49	1.34	1.84	0.86	1.63	6.74	0.85	1.84	**0.65**
d	6.35	4.86	11.00	5.05	6.33	17.55	5.55	**4.07**	4.59
e	3.07	4.18	7.47	0.56	0.74	0.70	**0.53**	1.52	0.97
f	2.25	2.32	1.07	0.73	1.01	89.98	**0.70**	1.36	0.74
g	1.55	0.62	0.54	0.56	0.90	93.01	**0.52**	4.13	0.57
h	3.56	3.91	**3.47**	4.99	6.84	9.10	4.34	3.79	4.28
i	8.54	6.92	**6.22**	11.27	15.59	29.12	9.16	7.14	9.00
j	59.45	4.55	7.02	0.61	1.01	87.41	**0.58**	2.32	0.61
k	5.32	3.97	3.44	2.67	3.46	2.83	2.41	2.42	**2.07**
l	7.93	6.80	5.86	5.85	6.71	6.22	5.51	**3.65**	4.34
m	4.21	2.17	3.47	0.57	0.81	50.16	**0.56**	1.28	0.59
n	14.44	3.39	15.98	1.42	2.26	28.40	**1.36**	1.69	1.49
o	3.44	**2.81**	3.88	3.78	6.20	5.96	3.20	3.05	3.42
p	6.97	3.01	7.46	**0.85**	1.43	1.72	0.86	1.58	0.92
q	44.00	12.34	23.8	**2.20**	3.16	9.86	2.94	2.48	2.31
r	6.63	3.73	4.49	4.32	5.44	**2.92**	3.54	3.13	3.49
MEAN	10.58	4.19	6.79	2.87	3.92	25.41	2.64	2.79	**2.49**
RAND	5.47	4.19	5.91	2.11	2.97	4.58	1.99	2.43	**1.27**

The second study was for 2, 4, 8 and 16 components. The source densities were selected at random from the 18 densities. Results are presented in Table 5. The bottom row shows the column averages. It is seen that LICA has the smallest average among its peers.

In summary, LICA performed similarly with the best for smaller data sets while it was the best for moderate data sets. This may be explained by the fact that LICA is based on the highly adaptive logspline density estimation methodology having the knots selected adaptively. For smaller data sets, the variability of the logspline estimates seemed to dominate the bias. The variability apparently subsided for moderate data sets.

Table 5: Results for experiments in higher dimensions (again, mean Amari error multiplied by 100). The table shows experiments for dimensions 2 through 16. The number of points used for each experiment is shown in the second column and the number of experiment replications.

dims	N	#repl	fICA	JADE	exinfo	KGV	RAD
2	250	1000	11.85	9.11	19.43	**4.98**	7.76
2	1000	1000	5.47	4.19	5.91	2.11	2.97
4	1000	100	4.27	4.56	11.06	3.43	3.12
4	4000	100	1.91	2.21	4.21	1.29	1.49
8	2000	50	2.97	3.21	11.32	3.53	1.90
8	4000	50	1.97	2.22	6.04	2.79	1.19
16	4000	25	1.95	2.50	13.52	17.83	1.44
		MEAN	4.34	4.00	10.21	5.13	3.28
			NPICA	KDICA	PDICA	LICA	
2	250	1000	11.48	5.3	5.89	5.11	
2	1000	1000	4.58	1.99	2.43	**1.27**	
4	1000	100	2.30	2.36	2.56	**1.96**	
4	4000	100	1.36	1.22	**0.53**	0.93	
8	2000	50	1.36	2.07	1.72	**1.34**	
8	4000	50	0.99	1.48	1.10	**0.94**	
16	4000	25	**0.91**	3.05	1.24	1.06	
		MEAN	2.84	2.50	2.21	**1.80**	

4. Conclusion

In this paper, we described a new algorithm for ICA using polynomial splines to model the logarithm of the source density functions, and studied issues related to the global maximization. We also closely examined issues related to the initialization of the existing ICA algorithms. An important finding in our study was that the initialization played an important role in the performance of these procedures, and the results could be very sensitive to the initialization. Therefore, in using these procedures, a proper choice for initial guesses is essential. Another issue is that the global maximization can be very time consuming, especially for high dimensional problems involving a large number of components or sources. Nevertheless, we found that the logspline based approach to ICA was extremely rewarding, its superior performance over existing methods is very encouraging and we are now modifying our procedure, which is specifically designed for blind source separation, to handle fMRI data. Another important feature that was currently set aside in our current approach is the temporal or spatial correlation structure in the data, which should be fully incorporated in the implementation of ICA in order to achieve better performance or desirable results such as highly interpretable components (or factors or sources) in practical applications.

Acknowledgement

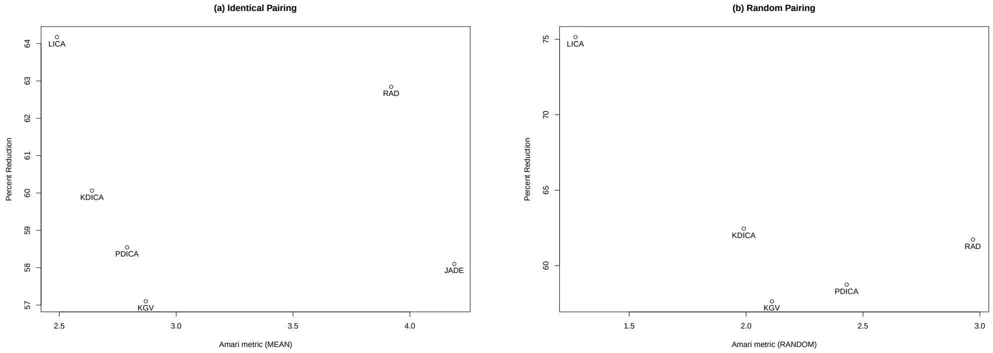

Figure 2: Scatterplots of percent reduction in Amari metric values (when sample size was increased from 250 to 1000) vs Amari metric values with 1000 samples (Table 4). LICA has the most relative reduction and the smallest Amari metric values at 1000 samples. (a) Identical pairings: Six smallest Amari metric values; LICA achieved 64% relative reduction with a value of 2.49 in Amari metric. (b) random pairing: Five smallest Amari metric values; LICA achieved 75% relative reduction with a value of 1.27 in Amari metric.

This research was supported in part by Grants-in-Aid from the Ministry of Education, Culture, Sport, Science and Technology of Japan (21700312) to AK, and by NSF DMS-0707090 to YT.

References

Amari, S., Cichocki, A. and Yang, H. H. (1996). A new learning algorithm for blind signal separation. *Advances in Neural Information Processing Systems*, 8, 757–763.

Bach, F. R. and Jordan, M. I. (2002). Kernel independent component analysis. *Journal of Machine Learning Research*, 3, 1–48.

Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation* **7**, 1129–1159.

Boscolo, R., Pan, H. and Roychowdhury V. P. (2004). Independent component analysis based on nonparametric density estimation. *IEEE Trans. Neural Networks* **15**, 55–65.

Calhoun, V. D., Adali, T., Hansen, L. K., Larsen, J. and Pekar, J. J. (2003). ICA of functional MRI data: an overview. In Proceedings of 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), 281–288.

Cardoso, J. F. (1998). Blind signal separation: Statistical principles. *Proc. IEEE. Special Issue on Blind Identification and Estimation* **9**, 2009–2025.

Cardoso, J. F. (1999). High-order contrasts for independent component analysis. *Neural Computation* **11**, 157–192.

Chen, A. (2006). Fast Kernel Density Independent Component Analysis. *In Proceedings of 6th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA 2006)*, 24–31.

Cichocki, A. and Amari, S. (2002). *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. New York: John Wiley & Sons.

Comon, P. and Jutten, C. (2010). *Handbook of Blind Source Separation: Independent component analysis and applications*. New York: Academic Press.

Hastie, T. and Tibshirani, R. (2002). Independent component analysis through product density estimation. *Proceedings of Neural Information Processing Systems* 649–656.

Hyvärinen, A. (2011). Testing the ICA mixing matrix based on inter-subject or inter-session consistency. *Neuroimage* **58** 122–136.

Hyvärinen, A. and Oja, E. (1997). A fast fixed point algorithm for independent component analysis. *Neural Computation* **9**, 1483–1492.

Hyvärinen, A., Karhunen, J. and Oja, E. (2001). *Independent Component Analysis*. New York: John Wiley & Sons.

Lee, T. W., Girolami, M. and Sejnowski, T. J. (1999). Independent component analysis using an extended Infomax algorithm for mixed sub-gaussian and super-gaussian sources. *Neural Computation* **11**, 417–441.

McKeown, M. J. and Sejnowski, T. J. (1998). Independent Component Analysis of fMRI Data: Examining the Assumptions. *Human Brain Mapping* **6**, 368–372.

Miller, E. and Fisher, J. (2003). ICA using spacings estimates of entropy. *Journal of Machine Learning Research* **4**, 1271–1295.

Pham, D. T., Garrat, P. and Jutten C. (1992). Separation of a mixture of independent sources through a maximum likelihood approach. *In Proceedings of EUSIPCO*, 771–774.

Stone, C. J., Hansen, M., Kooperberg, C. and Truong, Y. K. (1997). Polynomial splines and their tensor products in extended linear modeling, with discussions. *The Annals of Statistics* **25**, 1371–1470.

Stone, J. V. (2004). *Independent Component Analysis: A Tutorial Introduction.* MIT Press: Cambridge.

Vlassis, N. and Motomura, Y. (2001). Efficient source adaptivity in independent component analysis. *IEEE Trans. Neural Networks* **12**, 559–565.

Received March 17, 2011
Revised December 7, 2011