

Rank based Least-squares Independent Component Analysis

Jafar Rahmanishamsi and Ali Dolati*

Yazd University

Received: 2/22/2017 Approved: 2/7/2018

Abstract. In this paper, we propose a nonparametric rank-based alternative to the least-squares independent component analysis algorithm developed. The basic idea is to estimate the squared-loss mutual information, which used as the objective function of the algorithm, based on its copula density version. Therefore, no marginal densities have to be estimated. We provide empirical evaluation of the proposed algorithm through simulation and real data analysis. Since the proposed algorithm uses rank values rather than the actual values of the observations, it is extremely robust to the outliers and suffers less from the presence of noise than the other algorithms.

Keywords. Copula; independent component analysis; squared-loss mutual information.

MSC 2010: 62H20; 62H05.

1 Introduction

Independent Component Analysis (ICA) is an important method in signal processing, which consists of recovering unobserved signals from their observed mixtures. Assume that we observe d linear mixtures $\mathbf{x} = (x_1, \dots, x_d)^T$

* Corresponding author

of d independent components $\mathbf{s} = (s_1, \dots, s_d)^T$, that is $\mathbf{x} = A\mathbf{s}$, where A is the $d \times d$ mixing matrix. The independent components s_i 's are latent random variables with zero mean, which cannot be observed directly. To solve this problem the most popular method is to extract the components via an optimization problem, in which the statistical dependency among them is minimized (Hyvarinen et al., 2004). In fact, the goal of ICA is to find linear combinations of the observed data \mathbf{x} into a set of patterns $\mathbf{y} = W\mathbf{x}$, such that whose components y_i 's, are statistically independent from each other as possible. Most of ICA algorithms, minimize an objective function (also called contrast function) which measures the degree of dependency between the components of \mathbf{y} . The performance of the ICA algorithms depends on the distribution of initial sources, the choice of the objective function and the algorithm used for implementation of the optimization problem. Minimization of the mutual information (MI), as a natural dependency measure between sources, is commonly used in this framework and the objective function of the well known ICA algorithms such as JADE (Cardoso and Souloumiac, 1993), Infomax (Bell and Sejnowski, 1995) and RADICAL (Learned-Miller and Fisher, 2003) are based on this index. A problem with MI is the hard task of the estimation of the marginal and joint densities of sources. Suzuki and Sugiyama (2011) employed a squared-loss variant of mutual information as the objective function and introduced the Least-squares Independent Component Analysis (LICA). Their basic idea was to estimate the ratio of probability densities directly for avoiding the hard task of density estimation. Since the LICA algorithm applies the original data, it could be sensitive to the outliers. A solution to this problem is the using of rank values instead of original data. In this paper, we develop a new ICA algorithm called rank-based least squares independent component analysis (RLICA) that resolves the problem mentioned above. Our basic idea is to estimate the squared-loss mutual information based on its copula density version. Therefore, no marginal densities have to be estimated. In some ICA application, original sources may follow nonsymmetric distributions, and a rank-based ICA algorithm could be more effective than the usual ICA algorithms. The remaining of the paper is organized as follows. After recalling some basic concept of ICA in Section 2, Section 3 discusses the copula-based squared-loss mutual information as a rank-based measure of dependence using kernel estimation of the copula density. In Section 4, the detail procedure of RLICA algorithm whose contrast function is the rank-based estimator of the squared-loss mutual information is introduced. In Section 5 we compare the robustness of

LICA and RLICA algorithms to the outliers. In Section 6, we compare the performance of RLICA with LICA and other ICA algorithms in terms of their Amari error and running time through simulated and real data examples. Section 7 concludes the paper.

2 Independent Component Analysis

Independent component analysis of a random vector involves of searching for a linear transformation that minimizes the statistical dependency between its components see (Comon, 1994). ICA can be viewed as a generalization of the principal component analysis (PCA). PCA decorrelates original data so that the sample covariance of obtained components is zero. Whiteness is a stronger constraint that requires both decorrelation and unit variance. The whitening transform can be calculated as $\mathbf{D}^{-1/2}\mathbf{R}^T$, where \mathbf{D} is the diagonal eigenvalues matrix and \mathbf{R} is the orthogonal eigenvectors matrix. ICA has one step further so that convert the whitend data into a set of independent components. In summary, the goal in PCA is to find orthogonal decorrelated components, but ICA separates original data to statistically independent components. Assume we have d independent 1-dimensional sources (random variables) denoted by S_1, \dots, S_d and let $\mathbf{S} = [S_{ij}] \in \mathbb{R}^{N \times d}$ be a matrix of N random samples. It is assumed that these sources are hidden, and a matrix $\mathbf{X} = \mathbf{A}\mathbf{S}$ of mixed samples can be observed where $\mathbf{A} \in \mathbb{R}^{d \times d}$ is the mixing matrix. The task is to recover the sample matrix \mathbf{S} of the hidden sources by finding a demixing matrix $\mathbf{W} \in \mathbb{R}^{d \times d}$ such that the columns of the new matrix $\mathbf{Y} = \mathbf{W}\mathbf{X}$ has the smallest statistical dependency. Jutten and Herault (1991) explained the ICA problem as a general framework to determine the Blind Source Separation (BSS) problem. The well-known example of the ICA is the cocktail party problem, which is described as follows: In a party, there is a discussion of d people. We take d microphones at d different location in the room to record the discussion. When the party finish, if someone needs to recover the speech of each person, he or she can use the d observed signals of d recorders. The problem is how they can recover the speech of each person based on d recording signals. If the speeches of each person are nearly independent, ICA technique can help us to recover latent signals. In practice, ICA can be applied in blind source separation (Comon, 1994), image processing (Hyvarinen and Oja, 2000), natural image patch (Bell and Sejnowski, 1995), video watermarking (Sun et al., 2009), electrocardiogram (Taigang et al., 2006), data mining (Peng and Siming, 2007), financial time series and

prediction of time series (Hyvarinen et al., 2004). Let δ be a measure of dependency, and \mathbf{x} be a given d -dimensional random vector. Most methods for formulating the ICA problem, use a contrast function which measures the degree of dependency between the components of \mathbf{y} . We define the contrast function C_X as $C_X(\mathbf{W}) = \delta(\mathbf{W}\mathbf{x})$ the dependency of a new random vector $\mathbf{W}\mathbf{x}$. Several algorithms were proposed to estimate the independent components based on different objective functions. A primal approach to ICA is given by the rule of non-Gaussianity. Estimation of the IC's can be executed by maximizing such non-Gaussianity measures. The well known FastICA algorithm is formulated based on maximization of non-Gaussianity via measurements such as kurtosis and negentropy (Hyvarinen, 1999). Other confirmed approach for estimating the IC's is maximum likelihood (ML) estimation. Infomax algorithm uses this method to obtain the components. It takes the information dependency as a contrast function (Bell and Sejnowski, 1995). The mutual information as a natural measure of dependence between random variables, which is non-negative, and is zero only when the variables are independent, is commonly used in this framework. Estimation of IC's by minimization of MI was probably first recommended in (Comon, 1994), who derived an estimation based on cumulants. A gradient ICA algorithm based on minimizing the mutual information is discussed in (Bell and Sejnowski, 1995). The kernel independent component analysis (Kernel ICA) applies contrast functions based on canonical correlation that are related to mutual information and have desirable properties as statistical measures of dependence. This technique, which uses flexible kernels to model the dependence between the variables, was proposed by Bach and Jordan (2002). An optimization method named Robust Accurate Direct ICA algorithm (RADICAL) utilizes an estimate of univariate entropies to find Jacobi rotations that make pairs of components as independent as possible. The method is based on efficient spacing estimates of entropy and avoids explicit estimation of probability densities as an intermediate step. This approach consists of applying higher-order cumulant tensor. A blind identification algorithm by Joint Approximate Diagonalization of Eigen-matrices (JADE) was suggested by Cardoso and Souloumiac (Cardoso and Souloumiac, 1993). Gretton et al. (2005) proposed an algorithm based on Hilbert-Schmidt Independent Criterion (HSIC). FastKICA used an optimization method for HSIC (Shen et al., 2009). This technique uses a proper Newton method on the special orthogonal group. This is a novel optimization approach for one such kernel measure of dependence. The SWICA algorithm introduced based on Schweizer-Wolff

measure of dependence to estimate the independent components (Krishner and Poczos, 2008). Since this dependency measure is nonparametric and is zero only when the sources are independent, it can be used as a contrast function in ICA. Ameri et al. (1996) proposed an algorithm that computes a mixing matrix of ICA, in a parametric subclass of α -stable distributions. In this article, rank-based squared-loss mutual information is applied as a contrast function and introduced a new ICA algorithm called RLICA.

3 Copula-based Squared-loss Mutual Information

In this section, a rank-based squared-loss mutual information will be proposed using kernel estimation of the copula densities. Let X_1 and X_2 be two continuous random variables with the joint density function $f(\cdot, \cdot)$ and the respective univariate marginal density functions $f_1(\cdot)$ and $f_2(\cdot)$. The squared-loss mutual information (SMI) is defined by

$$SMI(X_1, X_2) = \int_{\mathbf{R}^2} \left(\frac{f(x_1, x_2)}{f_1(x_1)f_2(x_2)} - 1 \right)^2 f_1(x_1)f_2(x_2)dx_1dx_2, \quad (1)$$

which is the Pearson's ϕ^2 distance from the joint density function $f(x_1, x_2)$ to the product of marginal density functions $f_1(x_1)f_2(x_2)$. This index is a measure of dependency with the range $[0, \infty]$, and it is zero if and only if X_1 and X_2 are statistically independent (Joe, 1989). Since SMI includes unknown density functions $f(x_1, x_2)$, $f_1(x_1)$ and $f_2(x_2)$, the problem is known to be a hard task. Suzuki and Sugiyama (2011) considered estimating the density ratio function $r(x_1, x_2) = f(x_1, x_2)/f_1(x_1)f_2(x_2)$ directly from data samples without going through estimation of $f(x_1, x_2)$ and $f_1(x_1)$ and $f_2(x_2)$. In what follows we consider a different approach to estimate SMI. Let $F(x_1, x_2) = P(X_1 \leq x_1, X_2 \leq x_2)$ be the joint cumulative distribution function of X_1 and X_2 with the respective univariate marginal distribution functions $F_1(x_1) = P(X_1 \leq x_1)$ and $F_2(x_2) = P(X_2 \leq x_2)$. Let C be the unique copula associated with the pair (X_1, X_2) through the relation $F(x_1, x_2) = C\{F_1(x_1), F_2(x_2)\}$, in view of *Sklar's Theorem* (Sklar, 1959). In fact, C is the cumulative distribution function of the pair $(U, V) = (F_1(X_1), F_2(X_2))$ of uniform (0,1) random variables. See, Nelsen (2006) for a complete treatment on copulas. In view of Sklar's Theorem, the joint density function of X_1 and X_2 could be written as $f(x_1, x_2) = f_1(x_1)f_2(x_2)c\{F_1(x_1), F_2(x_2)\}$, where $c(u, v) = \partial^2 C(u, v)/\partial u \partial v$ is the density of the copula C . Using the

transformations $u = F_1(x_1)$ and $v = F_2(x_2)$ a convenient expression of SMI in terms of the copula density is given by

$$\begin{aligned}
 SMI(X_1, X_2) &= \int_0^1 \int_0^1 (c(u, v) - 1)^2 dudv \\
 &= \int_0^1 \int_0^1 c^2(u, v) dudv - 1 \\
 &= \int_0^1 \int_0^1 c(u, v) dC(u, v) - 1 \\
 &= E_C[c(U, V)] - 1,
 \end{aligned} \tag{2}$$

where U and V are two uniform (0,1) random variables whose joint distribution function is the copula C (Mohtashami and Amini, 2010). Calsaverini and Vicente (2009) discussed the connection between information and copula theories. A useful property of SMI is that it is invariant under monotone transformations of random variables. If ψ and ϕ are both increasing (resp, decreasing) functions, we note that $c_{\psi(X_1), \phi(X_2)}(u, v) = c_{X_1, X_2}(u, v)$ (resp, $c_{\psi(X_1), \phi(X_2)}(u, v) = c_{X_1, X_2}(1 - u, 1 - v)$). When ψ is decreasing (increasing) and ϕ is decreasing (increasing) then $c_{\psi(X_1), \phi(X_2)}(u, v) = c_{X_1, X_2}(1 - u, v)$ ($c_{\psi(X_1), \phi(X_2)}(u, v) = c_{X_1, X_2}(u, 1 - v)$).

In the following we propose a nonparametric rank-based estimator of SMI index. Given a random sample $\{(X_{1i}, X_{2i})\}$, $i = 1, 2, \dots, n$, from an unknown bivariate distribution F with the associated copula C , and the univariate marginal distribution functions F_1 and F_2 , we derive a non-parametric rank-based estimator for SMI. We assume that C , as well as F_1 and F_2 , are completely unknown. For $j = 1, \dots, n$, let (R_{1j}, R_{2j}) denote the corresponding vectors of ranks, i.e., $R_{ij} = \sum_{l=1}^n \mathbb{I}\{X_{lj} \leq X_{ij}\}$, for $i = 1, 2$ and $1 \leq j \leq n$, where $\mathbb{I}\{A\}$ denotes the indicator function of the set A . Let F_{in} be the re-scaled empirical counterpart of F_i , $i = 1, 2$, i.e.,

$$F_{in}(x) = \frac{1}{n+1} \sum_{j=1}^n \mathbb{I}\{X_{ij} \leq x\},$$

and consider the pseudo-observations

$$\hat{U}_{ij} = F_{in}(X_{ij}) = \frac{R_{ij}}{n+1}, \quad i = 1, 2, \quad j = 1, 2, \dots, n.$$

As the marginal distribution functions are assumed to be continuous, ties

occur with the probability zero. The rescaled empirical distribution function differs from the usual empirical distribution function by the use of the denominator $n+1$ rather than n . This guarantees that the pseudo-observations \widehat{U}_{ij} , are strictly located inside the unit square $[0, 1] \times [0, 1]$. According to the equation (2), we need an estimator for the copula density to obtain a rank-based estimator for *SMI*. The kernel estimator of a copula density (Wand and Jones, 1995) is defined by

$$\widehat{c}(u, v) = n^{-1}h^{-2} \sum_{i=1}^n K \left(\frac{u - U_i}{h}, \frac{v - V_i}{h} \right),$$

where h is a bandwidth parameter and K is a bivariate kernel function. Since copulas are defined on the unit square $[0, 1] \times [0, 1]$, their densities are subject to this restriction as well. For the kernel distribution function see, (Jabari, 2009). Chen (1999) introduced the univariate Beta kernel density estimator to estimate a density function with compact support. Its definition is based on the density function of the univariate Beta distribution

$$\kappa(u; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} u^{\alpha-1} (1-u)^{\beta-1}, \quad u \in [0, 1],$$

where $B(\alpha, \beta) = \int_0^1 x^{\alpha-1} (1-x)^{\beta-1} dx$, is the Beta function. To formulate a bivariate extension, we use the product kernels developed by Charpentier et al. (2006) and consider the estimate of the copula density as

$$\begin{aligned} \widehat{c}(u, v) &= n^{-1}h^{-2} \sum_{i=1}^n \kappa \left(\widehat{U}_{1i}; \frac{u}{h} + 1, \frac{1-u}{h} + 1 \right) \\ &\quad \times \kappa \left(\widehat{U}_{2i}; \frac{v}{h} + 1, \frac{1-v}{h} + 1 \right). \end{aligned} \quad (3)$$

The beta kernels have these advantages: (1) they could match the compact support of the object to be estimated and (2) they have a flexible structure and change the smoothness in a usual way as we move away from the boundaries. As a result, beta kernel estimators are free of boundary bias and could generate estimates with a smaller variance. Monte Carlo simulation results presented in Charpentier et al. (2006) showed that beta kernel estimators have better performance compared to the other estimators such as local linear (Jones, 1993) or boundary kernel (Müller, 1991) estimators, which are

free of boundary bias. In addition, Bouezmarni and Rolin (2003) displays that the density estimator based beta kernel is consistent, even if the density is unbounded at the boundaries. This property may also happen in the copula density estimation problem. For example, the bivariate Gaussian copula density is unbounded at the corners $(0, 0)$ and $(1, 1)$. Thus the beta kernels are applicable candidates to build well-behaved nonparametric estimators for a copula density. Assuming that the copula density c is twice differentiable on $[0, 1] \times [0, 1]$, the bias of $\hat{c}(u, v)$ is given by

$$E(\hat{c}(u, v)) = c(u, v) + Q(u, v)h + o(h),$$

where the bias $Q(u, v)$ is

$$Q(u, v) = (1 - 2u)c_1(u, v) + (1 - 2v)c_2(u, v) + \frac{1}{2} [u(1 - u)c_{1,1}(u, v) + v(1 - v)c_{2,2}(u, v)],$$

and $c_1 = \partial c(u, v)/\partial u$, $c_2 = \partial c(u, v)/\partial v$, $c_{1,1} = \partial^2 c(u, v)/\partial^2 u$, $c_{2,2} = \partial^2 c(u, v)/\partial^2 v$. For Beta kernel the bias is $O(h)$ (everywhere) while it is $O(h^2)$ via standard kernels. Also, when c is twice differentiable, the variance of \hat{c} in the corners, e.g. $(0, 0)$, is given by

$$Var(\hat{c}(0, 0)) = \frac{1}{nh^2} [c(0, 0) + o(n^{-1})].$$

In the interior of borders, e.g. $u = 0$ and $v \in (0, 1)$

$$Var(\hat{c}(0, v)) = \frac{1}{2nh^{3/2}\sqrt{\pi v(1-v)}} [c(0, v) + o(n^{-1})],$$

and in the interior, $(u, v) \in [0, 1] \times [0, 1]$

$$Var(\hat{c}(u, v)) = \frac{1}{4nh\pi\sqrt{u(1-u)v(1-v)}} [c(u, v) + o(n^{-1})].$$

In summary, the variance $Var(\hat{c}(u, v))$ depends on the location is of order $O(nh^k)^{-1}$, where $k = 2$ in the corners, $k = 3/2$ in the borders and $k = 1$ in the interior of $[0, 1] \times [0, 1]$. The optimal bandwidth h can be derived, by

maximizing the asymptotic mean square error (Charpentier et al., 2006)

$$b^* \equiv \left(\frac{1}{16\pi n Q(u, v)^2} \cdot \frac{1}{\sqrt{u(1-u)v(1-v)}} \right)^{1/3}.$$

Therefore, the optimal bandwidth is of order $O(n^{-1/3})$ in beta kernel estimation of 2-dimensional copula densities. In this paper we use $h = 1/\sqrt{n}$ as the bandwidth parameter. From the representation (2) the plug-in estimator of SMI is given by

$$\widehat{SMI} = \frac{1}{n} \sum_{j=1}^n [\widehat{c}(\widehat{U}_{1j}, \widehat{U}_{2j})] - 1. \quad (4)$$

In the next section, we will use this plug-in estimator of SMI as a contrast function to develop a rank-based ICA algorithm.

RLICA Algorithm

Input: A $2 \times N$ matrix \mathbf{x} where rows are mixed signals (centered and whitened), k equispaced evaluation angles in the $[0, \pi/2)$ interval for each of K angles θ in the interval $[0, \pi/2)$, $\theta = \frac{\pi k}{2K}$, $k = 0, \dots, K - 1$.

Procedure:

- 1 Compute the rotation matrix

$$\mathbf{W}(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}.$$

- 2 Compute rotated signals $\mathbf{y}(\theta) = \mathbf{W}(\theta)\mathbf{x}$.
- 3 Compute $\widehat{SMI}(\mathbf{y}(\theta))$.
- 4 Find best angle $\theta = \operatorname{argmin}_{\theta} \widehat{SMI}(\mathbf{y}(\theta))$.

Output: Rotation matrix $\widehat{W} := \widehat{W}(\theta)$, demixed signal $\mathbf{y} = \mathbf{y}(\theta)$.

4 The Proposed ICA Algorithm

Let \mathbf{x} be a random vector on the space R^d . Independent component algorithms, search a matrix $W \in R^{d \times d}$ such that the components y_1, \dots, y_d of the new random vector $\mathbf{y} = W\mathbf{x}$ has the smallest statistical dependency. A contrast function of an ICA algorithm based on a measure of dependence κ is a function $\delta_X(\cdot)$ such that $\delta_{\mathbf{x}}(W) = \kappa(W\mathbf{x})$. The ICA problem is finding $\widehat{W} \in R^{d \times d}$ which minimizes the contrast function $\delta_{\mathbf{x}}(W)$. In the following, we present a new algorithm using the estimate of *SMI* given by (4) as a contrast in demixing pairs of variables. We named this algorithm rank based-least squares independent component analysis or RLICA in short. First, we consider a two-dimensional case, i.e., $d = 2$, where the signal \mathbf{s} mixed with a 2×2 matrix A . We assume that A is orthogonal (otherwise achieved by a whitening process). The problem is then reduced to finding a demixing rotation matrix

$$W(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}.$$

We use \widehat{SMI} given by (4) as a contrast function computed on $2 \times N$ matrix $\mathbf{y}(\theta) = W(\theta)\mathbf{x}$ of rotated samples. Given an angle θ , the value of $\widehat{SMI}(\mathbf{y}(\theta))$ can be computed by ranks of the vector $\mathbf{y}(\theta)$. The solution is then found by finding angle θ minimizing $\widehat{SMI}(\mathbf{y}(\theta))$. We find such a solution by searching over K values of θ in the interval $[0, \frac{\pi}{2}]$. This algorithm is outlined in Algorithm 1. The idea for searching a rotation angle such that the corresponding demixed data set has its *SMI* minimized, is similar to the RADICAL algorithm (Learned-Miller and Fisher, 2003). As shown by (Comon, 1994) pairwise independence is a sufficient measure of statistical independence in ICA problem and thus a d -dimensional ICA problem can be solved by solving 2-dimensional ICA problems in succession. In fact, a d -dimensional linear transformation described by a $d \times d$ orthogonal matrix W is equivalent to a composition of 2-dimensional rotations. The transformation matrix itself can be written as a product of corresponding rotation matrices, $W = W_L \times \dots \times W_1$ where each matrix W_l , $l = 1, \dots, L$, is a rotation matrix (by angle θ_l) for some pair of dimensions (i, j) . The program is available upon request as a Matlab code toolbox.

5 Robustness to the Outliers

Outliers are a basic concern for ICA algorithms. Since RLICA is a rank-based method, we can expect that this method is robust to the outliers. We consider two one-dimensional densities, including uniform on the interval $(-0.5, 0.5)$ and the student-t distribution with 3 degrees of freedom to generate original independent sources. We simulated outliers by randomly choosing up to 25 observations to the corrupt. This was executed by adding the value $+5$ or -5 to selected data point. We will use the Amari Error (Amari et al., 1996) as the performance index (smaller is better):

$$\text{Amari Error} := \frac{1}{2d(d-1)} \sum_{i,j=1}^d \left(\frac{|b_{ij}|}{\max_j |b_{ij}|} + \frac{|b_{ij}|}{\max_i |b_{ij}|} \right) - \frac{1}{d-1}, \quad (5)$$

where $b_{ij} = [\widehat{W}A]_{ij}$ for an estimated demixing matrix \widehat{W} . The Amari error of two algorithms LICA and RLICA and five other algorithms calculated for 50 replications. Figures 1 and 2 show the plot of Amari errors versus the number of outliers for seven algorithms. The results confirm that RLICA, KICA and RADICAL are robust to the outliers. Also Figure 1 and Figure 2 show that when the number of outliers is more than 15, RLICA is more robust to the outliers than the KICA and RADICAL. However, due to low efficiency of RLICA for the t-distribution, Figure 2 show that in the cases without outliers, the Amari error of proposed algorithm is higher than others algorithms but when the number of outliers increases RLICA is robust. The results confirm that the RLICA method is significantly more robust to the outliers than the LICA algorithm.

6 Illustrative Examples

In this section, we evaluate the performance of the proposed ICA algorithm for several artificial and real data sets and compare RLICA algorithm with LICA (Suzanki and Sugiyama, 2011), FastICA (Hyvarinen, 1999), Kernel-ICA (Bach and Jordan, 2002), JADE (Cardoso and Souloumiac, 1993), RADICAL (Learned-Miller and Fisher, 2003) and IMAX (Bell and Sejnowski, 1995) algorithms. We use five real datasets ‘demosig’, ‘10halo’, ‘Sergio7’, ‘Speech4’ and ‘c5signals’ datasets applied by Suzuki and Sugiyama (2011) to compare the performance of the LICA algorithm with the existing

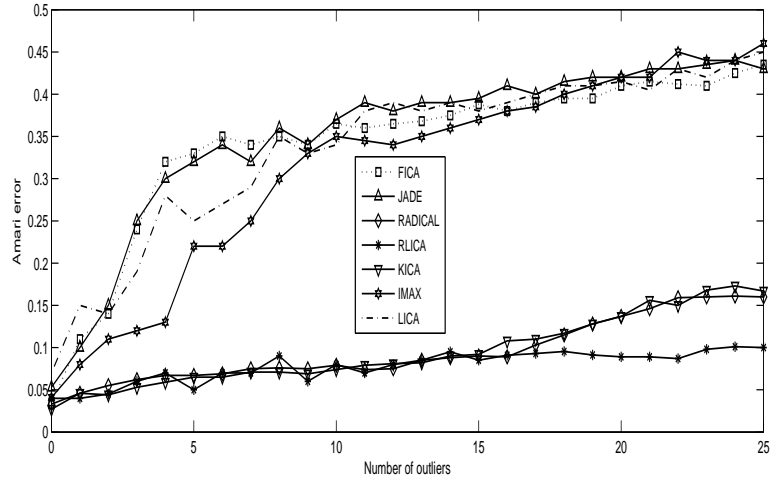


Figure 1. Plot of Amari error versus outliers for simulated data from uniform $(-0.5, 0.5)$ distribution ($d = 2$, $n = 1000$ and 50 replicates).

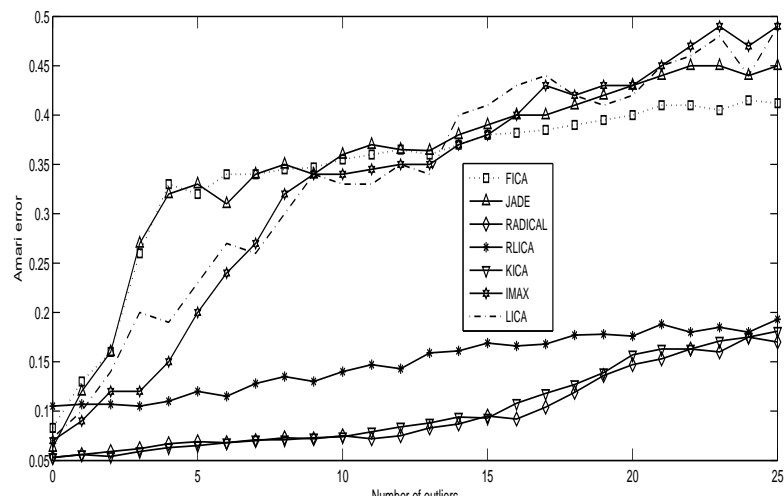


Figure 2. Plot of Amari error versus outliers for simulated data from student-t distribution with 3 degrees of freedom ($d = 2$, $n = 1000$ and 50 replicates).

methods. As training samples, we used the first 200 and 500 samples for Sergio and c5signals, and the 200 and 500 samples between the 1001 th and (1000+200 or +500) th interval for 10halo and Speech4. In the simulation study 18 different one-dimensional densities, including: normal, uniform, student-t, exponential, Laplace, mixtures of Laplace densities, as well as symmetric and non-symmetric mixtures of normal densities are used to generate original independent sources. The shapes of the densities presented in Figure 1. The procedure for generating simulated data is as the following: $N = 1000$ samples generated from each density function for $d = 2$ source signals and placed into a $2 \times N$ matrix \mathbf{x} . A random mixing matrix A was chosen. We used following mixing matrix

$$A = \begin{pmatrix} \cos(\pi/4) & \sin(\pi/4) \\ -\sin(\pi/4) & \cos(\pi/4) \end{pmatrix}.$$

A matrix \mathbf{y} of dimension $2 \times N$ was formed as the mixture $\mathbf{y} = A\mathbf{x}$, the data was whitened by multiplying \mathbf{y} by the inverse of the square root of the sample covariance matrix, yielding an $2 \times N$ matrix of whitened data \mathbf{y} . This matrix is the input of the ICA algorithms. Each of the ICA algorithms, outputs a demixing matrix \widehat{W} which can be applied to the matrix \mathbf{y} to recover the independent components. We computed the Amari index between the true A and \widehat{W}^{-1} for \widehat{W} estimated by each algorithm. Table 1 and Table 2 summarize the mean and standard deviation of the Amari error of each algorithm. For simulated data Amari error of each algorithm calculated for 50 replicates. As we note from Table 1 for 9 cases (b), (e), (f), (j), (k), (l), (p), (q) and (r) the RLICA algorithm exhibits a better performance than the other ICA methods. As we know, the major restriction in ICA is that the initial components must have non-Gaussian distribution. The Gaussian components cannot be separated from each other, so the most ICA methods perform poorly when the initial sources come from near Gaussian distribution. Results show for as (a), (d), (i), (m), and (o), the RLICA and some other ICA algorithms also perform poorly, because the initial sources come from near Gaussian distribution. RLICA requires more samples to separate near Gaussian sources. For the cases (c) and (g) the proposed algorithm tends to work reasonably well. From Table 2, for the real data sets, the RLICA algorithm exhibits a better performance than the other ICA methods for 'demosig', '10halo', 'Sergio7', 'Speech4' datasets but it performs poorly for 'c5signals' data sets. Overall, the proposed RLICA algorithm

is shown to be a promising ICA method, when the sources come from non-symmetric distributions like the exponential distribution, mixture of Laplace distributions and non-symmetric mixture of normal distributions.

In the following we consider the computational cost of the RLICA algorithm compare with the other ICA algorithms. The last row of the Table 1 displays average running time of 18 different densities (a to r) for the all algorithms we studied (in seconds). Results show that RLICA is slower than KICA, FICA, JADE, RADICAL and IMAX algorithms but is much faster than LICA. Also the last row of Table2 displays average running time of 5 different data sets for seven algorithms (in seconds). Results show that for $n = 200$ the running time of the RLICA is similar to the KICA and RADICAL algorithms and it is much faster than the LICA algorithm. For $n = 500$, the RLICA algorithm is much faster than LICA, but is slower than the six other ICA algorithms.

7 Conclusion

In this article, a new ICA method is proposed. The proposed algorithm uses a rank-based estimator of the squared-loss mutual information as its contrast function. By using simulated and real data examples, we showed that the proposed algorithm properly estimate the true demixing matrix and attains higher performance than usual ICA algorithms when original sources come from nonsymmetric distributions. We also showed that the proposed algorithm is extremely robust to the outliers than its competitor least-squares independent Component analysis, proposed by Suzuki and Sugiyama (2011). The proposed algorithm is slower than some usual ICA methods, but it is much faster than LICA, due to the using of rank values rather than the actual values of the observations.

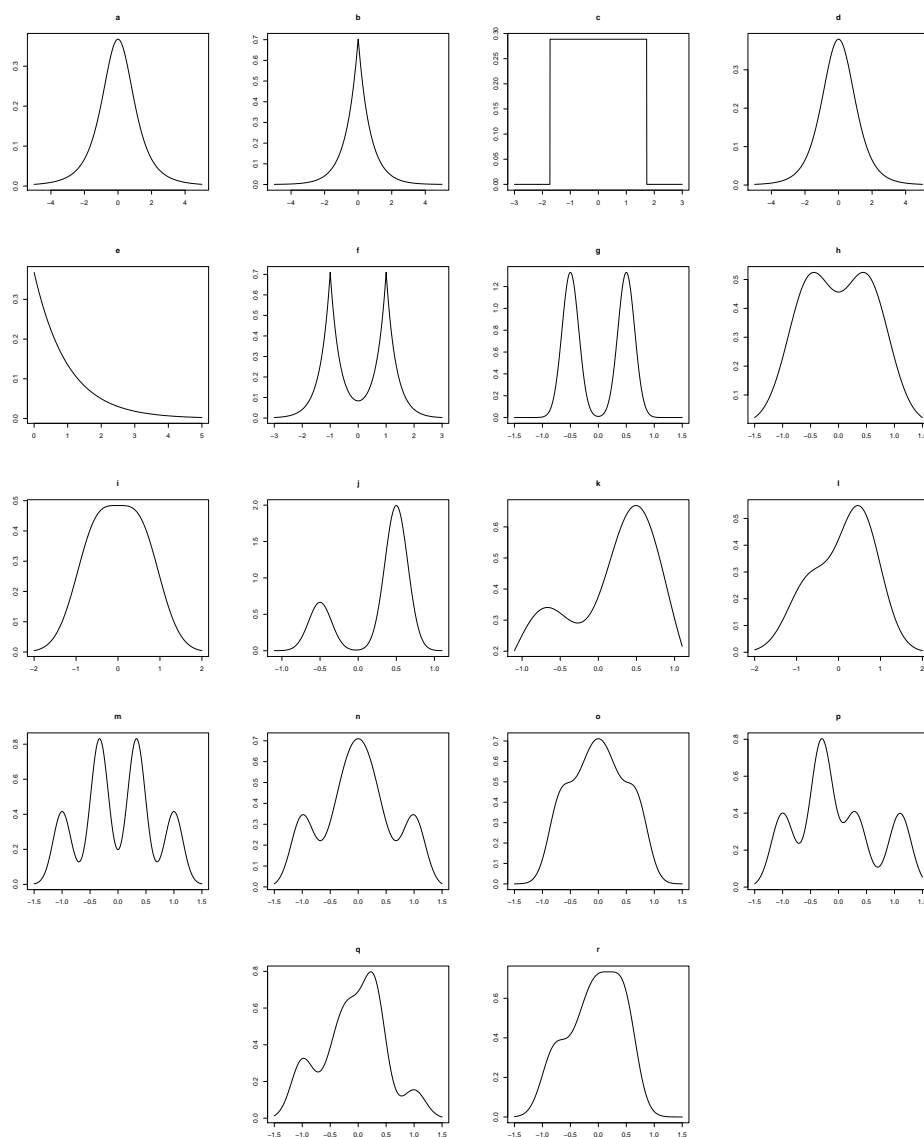


Figure 3. Density plots of 18 different distribution of sources: (a) Student-t with 3 degrees of freedom; (b) double exponential; (c) uniform; (d) Student-t with 5 degrees of freedom; (e) exponential; (f) mixture of two Laplace densities; (g)-(h)-(i) symmetric mixtures (multimodal, transitional and unimodal) of two normal densities; (j)-(k)-(l) nonsymmetric mixtures (multimodal, transitional and unimodal) of two normal densities; (m)-(n)-(o) symmetric mixtures (multimodal, transitional and unimodal) of four normal densities; (p)-(q)-(r) nonsymmetric mixtures (multimodal, transitional and unimodal) of four normal densities.

Table 1. Mean and standard deviation of the Amari error (for 50 replicates of $n = 1000$ samples) for the simulated data sets of 18 different densities. The best algorithm in terms of the mean Amari error and comparable ones based on the paired sample t-test at the significance level 1% are indicated by boldface.

Data set	RLICA	LICA	KICA	FICA	JADE	RADICAL	IMAX
a	0.112(0.058)	0.077(0.044)	0.073(0.049)	0.112(0.064)	0.110(0.060)	0.069(0.038)	0.053(0.025)
b	0.083(0.049)	0.094(0.078)	0.072(0.066)	0.104(0.070)	0.105(0.040)	0.084(0.041)	0.081(0.048)
c	0.047(0.028)	0.038(0.023)	0.033(0.025)	0.067(0.029)	0.032(0.014)	0.035(0.022)	0.048(0.041)
d	0.168(0.094)	0.134(0.178)	0.142(0.151)	0.089(0.079)	0.099(0.068)	0.138(0.097)	0.061(0.045)
e	0.019(0.007)	0.036(0.026)	0.038(0.045)	0.091(0.046)	0.103(0.034)	0.044(0.030)	0.024(0.011)
f	0.022(0.008)	0.030(0.020)	0.039(0.034)	0.074(0.088)	0.044(0.040)	0.046(0.028)	0.095(0.047)
g	0.024(0.009)	0.029(0.020)	0.040(0.034)	0.035(0.019)	0.018(0.009)	0.037(0.030)	0.013(0.008)
h	0.102(0.072)	0.102(0.079)	0.121(0.101)	0.113(0.067)	0.077(0.034)	0.119(0.101)	0.049(0.026)
i	0.255(0.201)	0.200(0.217)	0.387(0.284)	0.186(0.116)	0.112(0.159)	0.277(0.139)	0.124(0.078)
j	0.011(0.008)	0.128(0.020)	0.032(0.021)	0.085(0.079)	0.017(0.009)	0.038(0.024)	0.210(0.087)
k	0.052(0.043)	0.073(0.043)	0.089(0.073)	0.107(0.081)	0.062(0.048)	0.072(0.046)	0.071(0.039)
l	0.116(0.180)	0.203(0.283)	0.168(0.132)	0.198(0.152)	0.128(0.153)	0.098(0.077)	0.152(0.095)
m	0.099(0.066)	0.034(0.019)	0.040(0.058)	0.090(0.049)	0.055(0.024)	0.039(0.028)	0.070(0.042)
n	0.075(0.115)	0.046(0.022)	0.050(0.035)	0.111(0.103)	0.079(0.110)	0.052(0.033)	0.158(0.110)
o	0.131(0.132)	0.070(0.041)	0.139(0.166)	0.099(0.091)	0.066(0.068)	0.088(0.065)	0.080(0.046)
p	0.031(0.023)	0.051(0.024)	0.049(0.085)	0.080(0.073)	0.048(0.044)	0.049(0.035)	0.089(0.053)
q	0.073(0.036)	0.107(0.057)	0.102(0.234)	0.231(0.166)	0.322(0.270)	0.091(0.039)	0.302(0.182)
r	0.104(0.042)	0.132(0.069)	0.136(0.085)	0.137(0.090)	0.131(0.118)	0.113(0.050)	0.140(0.095)
Average							
of time	31.02	707.12	2.40	0.06	0.07	3.31	0.23
(in seconds)							

Table 2. Mean and standard deviation of the Amari error (for 50 replicates of $n = 1000$ samples) for the benchmark data sets. The best algorithm in terms of the mean Amari error and comparable ones based on the paired sample t-test at the significance level 1% are indicated by boldface.

Data set	n	RLICA	LICA	KICA	FICA	JADE	RADICAL	IMAX
demosisg	200	0.026(0.014)	0.043(0.017)	0.068(0.021)	0.083(0.039)	0.077(0.060)	0.080(0.099)	0.099(0.050)
	500	0.021(0.008)	0.025(0.011)	0.044(0.012)	0.042(0.028)	0.042(0.022)	0.069(0.029)	0.068(0.032)
10halo	200	0.137(0.048)	0.291(0.025)	0.341(0.029)	0.330(0.059)	0.366(0.011)	0.305(0.172)	0.335(0.201)
	500	0.121(0.039)	0.233(0.021)	0.360(0.031)	0.217(0.028)	0.293(0.009)	0.218(0.103)	0.220(0.148)
Sergio7	200	0.045(0.031)	0.046(0.012)	0.381(0.042)	0.058(0.021)	0.068(0.013)	0.060(0.021)	0.108(0.073)
	500	0.040(0.027)	0.054(0.023)	0.370(0.034)	0.044(0.014)	0.045(0.009)	0.041(0.018)	0.043(0.015)
Speech4	200	0.092(0.044)	0.176(0.026)	0.290(0.052)	0.196(0.033)	0.201(0.014)	0.119(0.059)	0.217(0.109)
	500	0.060(0.029)	0.090(0.007)	0.135(0.043)	0.097(0.038)	0.071(0.008)	0.071(0.030)	0.067(0.035)
c5signals	200	0.123(0.058)	0.122(0.014)	0.241(0.140)	0.096(0.025)	0.124(0.013)	0.113(0.071)	0.180(0.122)
	500	0.069(0.033)	0.060(0.035)	0.072(0.056)	0.049(0.020)	0.069(0.007)	0.077(0.052)	0.091(0.036)
Average of time	200	19	234	20	0.32	0.43	17	0.12
(in seconds)	500	68.29	1033.01	22.11	0.48	0.52	32.98	0.18

Acknowledgement

The authors are very grateful to the Editor and two referees whose comments helped in improving the quality of this work.

References

- Amari, S., Cichocki, A. and Yang, H. (1996). A New Learning Algorithm for Blind Signal Separation. *Advances in Neural Information Processing Systems*. 757-763, MIT Press.
- Ameri, M.R., Shokripour, M., Mohammadpour, A., and Nassiri, V. (2013). Parametric Independent Component Analysis for Stable Distributions. *Artificial Intelligence Research*, **2**, 27-34.
- Bach, F.R. and Jordan, M.I. (2002). Kernel Independent Component Analysis. *JMLR*, **3**, 1-48.
- Bell, A.J. and Sejnowski, T.J. (1995). An Information Maximization Approach to Blind source Separation and Blind Deconvolution. *Neural Comput.* **7**, 1129-1159.
- Bouezmarni, T. and Rolin, J.M. (2003). Consistency of the Beta Kernel Density Function Estimator. *The Canadian Journal of Statistics/La Revue Canadienne de Statistique*, **31**, 89-98.
- Calsaverini, R.S. and Vicente, R. (2009). An Information-theoretic Approach to Statistical dependence: Copula Information. *EPL (Europhysics Letters)*, **88**, 68003.
- Cardoso, J.F. and Souloumiac, A. (1993). Blind Beamforming for Non Gaussian Signals. *IEE Proceedings-F*, **140**, 362-370.
- Charpentier, A., Fermanian, J.D. and Scaillet, O. (2006). The Estimation of Copulas: Theory and Practice, in *Copulas: from Theory to Application in Finance*, J. Rank, ed., Risk Book, London, pp. 35-60.
- Chen, S.X. (1999). Beta Kernel Estimators for Density Functions, *Comput. Statist. Data Anal.*, **31**, 131-145.
- Comon, P. (1994). Independent Component Analysis, a New Concept? *Signal Proc.*, **36**, 287-314.
- Gretton, A., Bousquet, O., Smola, A. and Schölkopf, B. (2005). Measuring Statistical Dependence with Hilbert-Schmidt Norms. *In Algorithmic learning theory* (pp. 63-77). Springer Berlin Heidelberg.

- Hyvärinen, A. (1999). Fast and Robust Fixed-point Algorithms for Independent Component analysis. *Neural Networks, IEEE Transactions on*, **10**, 626-634.
- Hyvärinen, A. and Oja, E. (2000). Independent Component Analysis: Algorithms and Application. *Neural Networks*, **13**, 411-430.
- Hyvärinen, A., Karhunen, J. and Oja, E. (2004). *Independent Component Analysis*. John Wiley & Sons, New York.
- Jabari, N.H. (2009). Almost Sure Convergence of Kernel Bivariate Distribution Function Estimator under Negative Association. *J. Statist. Res. Iran*, **6**, 243-255.
- Joe, H. (1989). Relative Entropy Measures of Multivariate Dependence. *JASA*, **84**, 157-164.
- Jones, M.C. (1993). Simple Boundary Correction for Kernel Density Estimation. *Statistics and Computing*, **3**, 135-146.
- Jutten, C. and Herault, J. (1991). Blind Separation of Sources, Part I : An Adaptive Algorithm based on Neuromimetic Architecture. *Signal Processing*, **24**, 1-10.
- Krishner, S. and Poczos, B. (2008). ICA and ISA using Schweizer-Wolff Measure of Dependence. In *Proceedings of the 25th International Conference on Machine Learning* (pp. 464-471). ACM.
- Learned-Miller, E.G. and Fisher, J.W. (2003). ICA using Spacings Estimates of Entropy. *JMLR*, **4**, 1271-1295.
- Mohtashami Borzadaran, R. and Amini, M. (2010). Information Measures via Copula Functions. *J. Statist. Res. Iran*, **7**, 47-60
- Müller, H.G. (1991). Smooth Optimum Kernel Estimators Near Endpoints. *Biometrika*, **78**, 521-530.
- Nelsen, R.B. (2006). *An Introduction to Copulas*. Springer, New York.
- Peng, H. and Siming Z. (2007). Handling of Incomplete Data Sets using ICA and SOM in Data Mining. *Neural Computing and Applications*, **16**, 167-172.
- Shen, H., Jegelka, S. and Gretton, A. (2009). Fast Kernel-based Independent Component Analysis. *Signal Processing, IEEE Transactions on*, **57**, 3498-3511.
- Sklar, A. (1959). Functions de Répartition à n Dimensions et Leurs Marges. *Publ. Inst. Statist. Univ. Paris*, **8**, 229-231.
- Sun, Z., Liu, J., Sun, J., Sun, X. and Ling, J. (2009). A Motion Location based Video Watermarking Scheme using ICA to Extract Dynamic Frames. *Neural Computing and Applications*, **18**, 507-514.

Suzuki, T. and Sugiyama, M. (2011). Least-squares Independent Component Analysis. *Neural Computation*, **23**, 284-301.

Wand, M.P. and Jones, M.C. (1995). *Kernel Smoothing*. Chapman and Hall, London.

Taigang, H., Clifford, G. and Tarassenko, L. (2006). Application of Independent Component Analysis in Removing Artefacts from the Electrocardiogram. *Neural Computing and Applications*, **15**, 105-116.

Jafar Rahmanishamsi

Department of Statistics,
Yazd University,
Yazd, Iran.
email: *jrahmanishamsi@yahoo.com*

Ali Dolati

Department of Statistics,
Yazd University,
Yazd, Iran.
email: *adolati@yazd.ac.ir*