Two pairwise iterative schemes for high dimensional blind source separation

Zaid Albataineh · Fathi Salem

Received: 3 January 2020 / Accepted: 15 June 2020 / Published online: 29 June 2020
© Springer Science+Business Media, LLC, part of Springer Nature 2020

Abstract
In this paper, we address the problem of scalability to higher dimensional space in blind source separation (BSS), where the number of sources is greater than two. Herein, we propose two schemes of pairwise non-parametric independent component analysis (ICA) algorithms based on Convex Cauchy–Schwarz Divergence (CCS–DIV) for high dimensional problem in BSS. We extend the pairwise method to the scenario of more than two sources, two improved ICA algorithms are developed. Moreover, we employ adaptive sampling technique that samples the signal into small time blocks to evaluate the integration of the CCS–DIV and reduce the computational complexity. The two presented methods enable fast and efficient demixing of sources in real-world high dimensional source applications. Finally, we demonstrate with simulations including a wide variety of source distributions, showing that our presented methods outperform many of the presently known methods in terms of performance and computational complexity.

Keywords Blind source separation (BSS) · Cauchy–Schwarz inequality · Non-parametric independent component analysis (ICA) · FastICA · RobustICA

1 Introduction

Blind Signal Separation (BSS) is one of the most challenging areas in signal processing. BSS remains an important area of research and development in many domains, e.g. biomedical engineering, image processing, communication system, speech enhancement, remote sensing, etc. BSS techniques do not assume full apriori knowledge about the mixing environment, source signals, etc. BSS includes two major domains: Independent Component Analysis (ICA), and Multichannel Blind Deconvolution (MBD) (Comon and Jutten 2010; Cichocki and Amari 2002; Zhang 2004).

Independent Component Analysis (ICA) is considered a key approach of BSS and unsupervised learning algorithms (Lin 1991; Chien and Chen 2006). ICA relagates to Principal Component Analysis (PCA) and Factor Analysis (FA) in multivariate analysis and data mining when the components are in the form of Gaussian distributions (Hyvarinen 1999; Zarzoso and Comon 2010; Boscolo et al. 2004). However, ICA is a technique that includes higher order statistics (HOS) where, in the static mixing case, the goal is to represent a set of random variables as a linear transformation of statistically independent components (Chen 2005; Jenssen et al. 2005).

ICA techniques are based on the assumption of non-Gaussianity and independence of the sources. Let an M observation vector \( \mathbf{x} = [x_1, x_2, \ldots, x_M]^T \) be obtained from M statistically independent sources \( \mathbf{s} = [s_1, s_2, \ldots, s_M]^T \) by the relation \( \mathbf{x} = \mathbf{A}\mathbf{s} \), where \( \mathbf{A} \) is an \( M \times M \) unknown invertible mixing matrix. The estimated (original) sources can be estimated by \( \hat{\mathbf{y}} = \mathbf{Wx} \) where \( \mathbf{W} \) is a demixing (filter) matrix. The goal in ICA is to determine a demixing matrix \( \mathbf{W} \) to estimate the source signals. ICA uses the non-Gaussianity of sources and a dependency measure to find a demixing matrix \( \mathbf{W} \). A measure, e.g., could be based on the mutual information (Cichocki and Amari 2002; Lin 1991), Higher Order
Statistic (HOS), such as the Kurtosis (Hyvarinen 1999; Zarzoso and Comon 2010), or Joint Approximate Diagonalization (Cardoso and Souloumiac 1993; Cardoso 1999; Comon 1994). In essence, the demixing matrix is obtained by optimizing such a contrast function (Albataineh 2018b).

Furthermore, the metrics of cumulants, likelihood function, negentropy, kurtosis, and mutual information have been developed to obtain a demixing matrix in different adaptations of ICA-based algorithms. FastICA (Hyvarinen 1999; Zarzoso and Comon 2010) was developed to maximize non-Gaussianity with relative speed and simplicity. Recently, Zarzosoa and Comon (Zarzoso and Comon 2010) proposed the Robust Independent Component Analysis (R-ICA) method for better convergence performance. They used a truncated polynomial expansion, rather than the output marginal probability density functions, to simplify the estimation process. Moreover, in Yokote and Matsuyama (2012), the authors developed the rapid ICA algorithm which takes advantage of multi-step past information with respect to a fixed-point method in order to augment the non-Gaussianity among the estimated signals. In Yoshioka et al. (2011), Albataineh (2018a), Takeda et al. (2010), the authors have presented ICA methods using mutual information. They constructed a formulation by minimizing the difference between the joint entropy and the marginal entropy among the estimated sources. Moreover, the Euclidean distance divergence (ED-DIV) (Xu et al. 1998) and the Kullback divergence (KDIV) (Matsuyama et al. 2000) were used as the measure functions for nonnegative matrix factorization (NMF) problems in Hsieh and Chien (2010), Cichocki et al. (2006), Jen-TzungChien (2012), Jan et al. (2009). Key performance differentiators among these approaches are (i) the quality of the estimated demixed signals, and (ii) the speed of computation. In light of the advancement of computational resources, speed is now a non-issue of most applications. However, the quality of the estimated signals is becoming of utmost importance and differentiation among the proposed methods (Na and Yu 2012; Schobben et al. 1999; Kampa et al. 2011; Albataineh and Salem 2017; Albataineh et al. 2020).

With the advent of growth of high dimensional data, many successful algorithms are proposed to perform ICA in the general scenario of more than two sources (see, e.g., Duda et al. 2001; Takeda et al. 2009; Seth et al. 2011; Zarzoso et al. 2006, and references therein). However, these methods experience many drawbacks in terms of computational complexity and lack of scalability to higher dimensional space. Also, they sometimes show poor convergence, especially for a large number of sources. In Bataineh and Salem (2018), Authors develop an effective and improved measure of dependency among the signals, and then they construct its corresponding (parametric and non-parametric) ICA algorithms. A novel family of dependency divergence is developed which they name Convex Cauchy Schwarz Divergence (CCS–DIV). They develop this new measure by conjugating a convex function into the Cauchy–Schwarz inequality-based divergence measure. This new contrast function has a wide range of effective curvature since it is controlled by a convexity parameter. The corresponding convex Cauchy–Schwarz divergence ICA (CCS–ICA) employs the Parzen window density approximation (Golub and Van Loan 1996) to distinguish the non-Gaussian structure of source densities. However, in this paper, we extended our previous work in Bataineh and Salem (2018) by applying the pairwise iterative schemes to tackle the high dimensional data problem for non-parametric ICA algorithms. We propose two effective pairwise ICA algorithms based on the CCS–DIV: one is based on the gradient descent and the other is based on the Jacobi optimization (Rutishauser 1966). We compare the proposed methods against existing methods of ICA to address their pros and cons.

The paper is organized as follows. In Sect. 2, a brief description of the convex Cauchy–Schwarz divergence is discussed and presented, then in Sect. 3, we propose two schemes of pairwise non-parametric independent component analysis (ICA) algorithms. Comparative simulation results and conclusion are given in Sects. 4 and 5, respectively.

## 2 The CCS–DIV measure

To improve the performance of the divergence measure and speed up the convergence, A novel divergence method that is based on conjugating the convex function into the Cauchy–Schwarz inequality is proposed in Bataineh and Salem (2018), presents. In this context, this divergence takes advantage of the convexity parameter alpha to control the convexity in the divergence function and to speed up the convergence in the ICA and NMF algorithms. Incorporating the joint distribution \(P_j = p(z_1, z_2)\) and the marginal distributions \(Q_M = p(z_1)p(z_2)\) into the convex function, say, \(f(\cdot)\) and conjugating them to the Cauchy–Schwarz inequality yields

\[
|f(P_j).f(Q_M)|^2 \leq (f(P_j).f(P_j)) \cdot (f(Q_M).f(Q_M))
\]

(1)

\[
|f(p(z_1, z_2)).f(p(z_1)p(z_2))|f(p(z_1, z_2)), f(p(z_1)p(z_2)).f(p(z_1)p(z_2))
\]

(2)

where \(<\cdot, \cdot>\) is an inner product; \(f(\cdot)\) is a convex function, e.g.,

\[
f(t) = \frac{4}{1 - \alpha^2} \left[ \frac{1 - \alpha}{2} + \frac{1 + \alpha}{2} t - \frac{i^2}{2} \right] \forall t \geq 1
\]

(3)

Now, based on the Cauchy–Schwarz inequality a new symmetric divergence measure is proposed, namely:
\[
D_{\text{CCS}}(P_j, Q_M, \alpha)
= \log \left( \frac{\int f^2(p(z_1, z_2)) dz_1 dz_2}{\left[ \int f(p(z_1) \cdot f(Q_M) dz_1 dz_2 \right]^2} \right) \tag{4}
\]

\[
D_{\text{CCS}}(P_j, Q_M, \alpha)
= \log \left( \frac{\int f^2(p(z_1, z_2)) dz_1 dz_2}{\left[ \int f(p(z_1) \cdot f(Q_M) dz_1 dz_2 \right]^2} \right) \tag{5}
\]

where, as usual, \( D_{\text{CCS}}(P_j, Q_M, \alpha) \geq 0 \) and equality holds if and only if \( P_j = Q_M \), which means that they are independent of each other. This divergence function is then used to develop the corresponding ICA and NMF algorithms. It is noted that the joint distribution and the product of the marginal densities in \( D_{\text{CCS}}(P_j, Q_M, \alpha) \) is symmetric. This symmetrical property does not hold for the Kullback–Leibler (KL) divergence (KL–DIV) (Xu et al. 1998), alpha divergence (\( \alpha \)-DIV) (Matsuyama et al. 2000), and f-divergence (f-DIV) (Cichocki and Amari 2002; Cichocki et al. 2006). We anticipate that it would be desirable in the geometric structure of the search space as it would result in similar behavior from all initial conditions. Additionally, the CCS–DIV is tunable by the convexity parameter, \( \alpha \), in the case \( \alpha \) is extendable. However, based on l'Hopital's rule, one can derive the realization of CCS–DIV for the case \( \alpha = 1 \) and \( \alpha = -1 \) by finding the derivatives, with respect to \( \alpha \), of the numerator and denominator for each part of \( D_{\text{CCS}}(P_j, Q_M, \alpha) \). Thus, the CCS–DIV with \( \alpha = 1 \) and \( \alpha = -1 \) are respectively given by (6) and (7).

\[
D_{\text{CCS}}(P_j, Q_M, 1)
= \log \left( \frac{\int \Omega(z_1, z_2)^2 dz_1 dz_2}{\Gamma(z_1, z_2)} \right) \tag{6}
\]

where

\[
\Omega(z_1, z_2) = p(z_1, z_2) \cdot \log(p(z_1, z_2)) - p(z_1, z_2) + 1
\]

\[
\Gamma(z_1, z_2) = \left( \int \Omega(z_1, z_2) \cdot \frac{Y(z_1, z_2) dz_1 dz_2}{\gamma} \right)^2
\]

\[
D_{\text{CCS}}(P_j, Q_M, -1)
= \log \left( \frac{\int \Psi(z_1, z_2)^2 dz_1 dz_2}{\Pi(z_1, z_2)} \right) \tag{7}
\]

where

\[
\Psi(z_1, z_2) = \log(p(z_1, z_2)) - p(z_1, z_2) + 1
\]

\[
\Lambda(z_1, z_2) = \log(p(z_1) \cdot p(z_2)) - p(z_1) \cdot p(z_2) + 1
\]

\[
\Pi(z_1, z_2) = \left( \int \Psi(z_1, z_2) \cdot \Lambda(z_1, z_2) dz_1 dz_2 \right)^2
\]

3 The convex Cauchy–Schwarz divergence independent component analysis (CCS–ICA)

3.1 Non-parametric ICA algorithm

Without loss of generality, we develop the ICA algorithm by using the CCS–DIV as a contrast function. Let us consider a simple system that is described by the vector-matrix form

\[
x = Hs + v
\]

where \( x = [x_1, \ldots, x_M]^T \) is a mixture observation vector, \( s = [s_1, \ldots, s_M]^T \) is a source signal vector, \( v = [v_1, \ldots, v_M]^T \) is an additive (Gaussian) noise vector, and \( H \) is an unknown full rank \( M \times M \) mixing matrix, where \( M \) is the number of source signals. To obtain a good estimate, \( y = Wx \) of the source signal, the contrast function CCS–DIV should be minimized with respect to the demixing filter matrix \( W \).

Thus, the components of \( y \) become least dependent when this demixing matrix \( W \) becomes a scaled permutation of \( H^{-1} \). Following the standard ICA procedure, the estimated source \( y \) can be carried out in two steps: 1) the original data \( x \) should be preprocessed by removing the mean, i.e., one assumes \( E[x] = 0 \) and also by a (pre-)whitening matrix \( V = \Lambda^{-1/2}E^T \), where the matrix \( \Lambda \) represents the eigenvectors of the eigenvalues matrices of the autocorrelation, namely an estimate of \( R_{xx} = E[xx^T] \). Consequently, the whitened data vector \( x_w = vx \) would have zero-mean and its covariance equal to the identity matrix, i.e., \( R_{xw} = I_M \).

The demixing matrix can be iteratively computed by, e.g., the gradient descent algorithm (Cichocki and Amari 2002):

\[
W(k + 1) = W(k) - \gamma \frac{\partial D_{\text{CCS}}(X, W(k))}{\partial W(k)}
\]

where \( k \) represents the iteration index and \( \gamma \) is a step size or a learning rate. Therefore, the updated term in the gradient descent is composed of the differentials of the CCS–DIV with respect to each element \( w_m \) of the \( M \times M \) demixing matrix \( W \). The differentials \( \frac{\partial D_{\text{CCS}}(X, W(k))}{\partial w_m(k)} \), \( 1 \leq m \leq M \) are calculated using a probability model and the CCS–DIV measure as in Jen-TzungChien (2012) and Schobben et al. (1999). However, the update procedure may be stopped, e.g., when the absolute increment of the CCS–DIV measure meets a predefined threshold value. During iterations, one should make the normalization step \( w_m = \frac{w_m}{\|w_m\|} \) for each row of \( W \),
where $\| \cdot \|$ denotes a norm. Please refer to Algorithm 1 for the proposed algorithm based on gradient descent.

In deriving the CCS–ICA algorithm, based on the proposed CCS–DIV measure $D_{CCS}(P_J, P_M, \alpha)$, usually, the vector $P_J$ corresponds to the probability of the observed data $(p(y_i) = p(Wx_i) = \frac{p(x_i)}{|\det(W)|})$ and vector $Q_M$ corresponds to the probability of the estimated or expected data $(\prod_1^M p(y_{ml}) = \prod_1^M p(w_{ml}x_i))$. Here, the CCS–ICA algorithm is detailed as follows. Let the demixed signals $y = Wx$ with its $m$th component denoted as $y_{ml} = w_{ml}x_i$. Then, $P_J = p(y_i) = p(Wx_i)$ and $Q_M = \prod_1^M p(y_{ml}) = \prod_1^M p(w_{ml}x_i)$. Thus, the CCS–DIV as the contrast function with the built–in convexity parameter $\alpha$, is

$$D_{CCS}(P_J, Q_m, \alpha) = \log \frac{\int_{\mathbb{R}^n} f^2(p(Wx_i)) dx_1 \ldots dx_M}{\int_{\mathbb{R}^n} f^2(Q_m) dx_1 \ldots dx_M}$$

$$D_{CCS}(P^T, Q_m, \alpha) = \log \frac{\int_{\mathbb{R}^n} f^2(P_J) dx_1 \ldots dx_M}{\int_{\mathbb{R}^n} f^2(Q_M) dx_1 \ldots dx_M}$$

For any convex function, we use the Lebesgue measure to approximate the integral with respect to the joint distribution of $y_i = \{y_1, y_2, \ldots, y_M\}$. The contrast function thus becomes

$$D_{CCS}(P_J, Q_m, \alpha) = \log \frac{\sum_1^T f^2(p(Wx_i)) \cdot \sum_1^T f^2(\prod_1^M p(w_{ml}x_i))}{\sum_1^T f(p(Wx_i)) \cdot f(\prod_1^M p(w_{ml}x_i))}$$

The adaptive CCS–ICA algorithms are carried out by using the derivatives of the proposed divergence, i.e., $(\delta D_{CCS}(P_J, Q_m, \alpha)/\delta w_{ml})$ as derived in the following section. Note that the determinant demixing matrix $(\det(W))$ with respect to the element $(w_{ml})$ equals the cofactor of entry $(m, l)$ in the calculation of the determinant of $W$, which we denote as $(\frac{\delta \det(W)}{\delta w_{ml}} = W_{ml})$. Also, the joint distribution of the output $y$ is determined by $p(y_i) = \frac{p(x_i)}{|\det(W)|}$.

For simplicity, we can write $D_{CCS}(P_J, Q_m, \alpha)$ as a function of three terms as follows:

$$D_{CCS}(P_J, Q_m, \alpha) = \log \left( \sum_1^T f^2(p(Wx_i)) \right) + \log \left( \sum_1^T f^2(\prod_1^M p(w_{ml}x_i)) \right) - 2\log \left( \sum_1^T f(p(Wx_i)) \cdot f(\prod_1^M p(w_{ml}x_i)) \right)$$

$$\frac{\delta D_{CCS}(P_J, Q_m, \alpha)}{\delta w_{ml}} = \sum_1^T 2(f'(p(Wx_i)) \cdot \prod_1^M p(w_{ml}x_i) \cdot \frac{\delta(p_{ml})}{\delta(w_{ml})} \cdot x_i) + \sum_1^T f^2(p(Wx_i)) \cdot \prod_1^M p(w_{ml}x_i) - 2\sum_1^T f(p(Wx_i)) \cdot f(\prod_1^M p(w_{ml}x_i))$$

$$\frac{\delta D_{CCS}(P_J, Q_m, \alpha)}{\delta w_{ml}} = \sum_1^T 2(f'(p(Wx_i)) \cdot \prod_1^M p(w_{ml}x_i) \cdot \frac{\delta(p_{ml})}{\delta(w_{ml})} \cdot x_i) - 2\sum_1^T f(p(Wx_i)) \cdot f(\prod_1^M p(w_{ml}x_i))$$

$$\frac{\delta D_{CCS}(P_J, Q_m, \alpha)}{\delta w_{ml}} = \sum_1^T 2(f'(p(Wx_i)) \cdot \prod_1^M p(w_{ml}x_i) \cdot \frac{\delta(p_{ml})}{\delta(w_{ml})} \cdot x_i) - 2\sum_1^T f(p(Wx_i)) \cdot f(\prod_1^M p(w_{ml}x_i))$$
where the convex function \( f(t) \) and its derivative \( f'(t) \) are respectively given by:
\[
\begin{align*}
    f(t) &= \frac{4}{1 - \alpha^2} \left[ 1 - \frac{\alpha}{2} t + \frac{1}{2} t^2 \right] \\
    f'(t) &= \frac{2}{1 - \alpha} \left[ 1 - t^2 \right]
\end{align*}
\]

In general, the estimation accuracy of a demixing matrix in the ICA algorithm is limited by the lack of knowledge of the accurate source probability densities. However, non-parametric density estimate is used in Duda et al. (2001), Na and Yu (2012), by applying the effective Parzen window estimation. One of the attributes of the Parzen window is that it must integrate to one. Furthermore, it exhibits a distribution shape that is data-driven and is flexibly formed based on its Kernel functions. Thus, one can estimate the density function \( p(y) \) of the process generating the \( M \)-dimensional sample \( y_1, y_2 \ldots y_M \) due to the Parzen Window estimator (Duda et al. 2001; Golub and Van Loan 1996). For all these reasons, a non-parametric CCS–ICA algorithm is also presented by minimizing the CCS–DIV to generate the demixed signals \( y = [y_1, y_2, \ldots, y_M]^T \). Here, the demixed signals are described by the following univariate and multivariate distribution estimates, respectively (Seth et al. 2011),
\[
\begin{align*}
    p(y_m) &= \frac{1}{Th} \sum_{t=1}^{T} \theta \left( \frac{y_m - y_{mt}}{h} \right) \\
    p(y) &= \frac{1}{Th^M} \sum_{t=1}^{T} \phi \left( \frac{y - y_t}{h} \right)
\end{align*}
\]

where \( h \) represents is a smoothing parameter called the bandwidth, the univariate and multivariate Gaussian Kernel are respectively represented by:
\[
\begin{align*}
    \theta(u) &= (2\pi)^{\frac{1}{2}} e^{-\frac{u^2}{2}} \\
    \phi(u) &= (2\pi)^{\frac{M}{2}} e^{-\frac{1}{2} u^T u}
\end{align*}
\]

The Gaussian kernel(s), used in the non-parametric ICA, are smooth functions. Several studies in the literature (Na and Yu 2012; Schobben et al. 1999), have shown that the performance of a learning algorithm based on the non-parametric ICA is better than the performance of a learning algorithm based on parametric ICA. By substituting (15) and (16) with \( y_t = Wx_t \) and \( y_{mt} = w_m x_t \), into (13).

Notably, in a high dimensional problem, this nonparametric CCS–DIV deteriorates from high computational complexity when recovering the joint distribution. However, we proposed to mitigate this problem due to adopt the pairwise iterative scheme (Rutishauser 1966; Golub and Van Loan 1996; Zarzoso et al. 2006) to minimize CCS-DIV divergence function in (13). Algorithm 1 outlines the steps of the non-parametric gradient descent ICA algorithm.

**Algorithm 1: The ICA Based algorithm using the gradient descent**

*Input:* \((M \times T)\) matrix of realizations \(X\), Initial demixing matrix \(W = I_M\), Max. number of iterations \(itr\), Step Size \(\gamma\) i.e., \(\gamma = 0.3\), alpha \(\alpha \) \(\alpha = -0.99999\)

*Perform Pre-Whitening*

\[X = V \cdot X = A^{(-1/2)} E^T X.\]

*For loop: for each I iteration do*

*For loop: for each \(t = 1, \ldots, T\) Evaluate the proposed contrast function and its derivative\(
(\partial D_{CCS}(P, Q_M, \alpha)) / \partial W\)*

*End For*

Update demixing matrix \(W\)

\[W = W - \gamma \frac{\partial D_{CCS}(P, Q_M, \alpha)}{\partial W}\]

*Check Convergence*

\[||\Delta W|| \leq \varepsilon \text{ i.e., } \varepsilon = 10^{-4}\]

*End For*

*Output: Demixing Matrix \(W\), estimated signals \(y\)*

### 3.2 Scenario of three source signals and more

Generally Speaking, the non-parametric ICA algorithm suffers from insufficient data and high computation in a high dimensional problem, especially when estimating the joint distribution. However, in several previous reports in the literature, e.g., Jen-Tzung Chien (2012), Takeda et al. (2009), the authors suggest applying the pairwise iterative schemes to tackle the high dimensional data problem for non-parametric ICA algorithm(s). However, there are no results indicating how the performance would hold up with the pairwise scheme, especially in terms of computational complexity and in terms of the accuracy of the non-parametric ICA algorithm. In this work, we present two effective pairwise ICA algorithms: one is based on the gradient descent and the other is based on the Jacobi optimization (Rutishauser 1966). Without loss of generality, one can represent the demixing matrix \(W\) as a series of rotational matrices in terms of an unknown angle \(\theta_{ij} \in [-\pi/4, \pi/4]\) between each pair \((i, j)\) of the observed signals. Specifically, one can define the pairwise rotation matrix as follows:

\[
W(\theta_{ij}) = \begin{bmatrix}
    \cos \theta_{ij} & -\sin \theta_{ij} \\
    \sin \theta_{ij} & \cos \theta_{ij}
\end{bmatrix}
\]

(17)

The idea is to make each pair of the estimated (marginal) output as “independent” as possible (i.e., minimize...
dependency). It was proven and pointed out by Comon in Comon (1994) that the mutual independence between the $M$ whitened observed signals can be attained by maximizing the independence between each pair of them. To that end, we present two algorithms to solve the high dimensional problem in the non-parametric scheme. First, we adopt the non-parametric algorithm based on the gradient descent into the pairwise iterative scheme of Algorithm 2.

Algorithm 2: The ICA method based on pairwise gradient decent scheme

**Input:** ($M \times T$) matrix of realizations $X$, Initial demixing matrix $W = I_M$, number of iterations $itr$, Step Size $\gamma$, e.g., $\gamma = 0.3$, $\alpha$, e.g., $\alpha = -0.99999$  

**Perform Pre-Whitening**  

$[X = V \ast X = A^{(c−1/2)}E^T \ast X]$,

**For loop:** for each $i = 1 \ldots M − 1$  

**For loop:** for each $j = i + 1 \ldots M$  

**Initial demixing matrix** $W_2 = I_2$  

**While:** while (true)  

Find $W_2$, e.g., using Algorithm 1 for each pairs of $X$;  

**End While**  

**Initial rotational matrix** $R = I_M$,  

**Update rotational matrix** $R([i, j], [i, j]) = W_2$  

**Update Demixing matrix** $W = R \ast W$  

**End For $i$**  

**End For $j$**  

**Output:** Demixing matrix $W = W \ast V$ and demixed sources $Y=W \ast X$.

Second, we propose a CCS–ICA algorithm based on Jacobi pairwise scheme in Algorithm 3. This algorithm is based on finding the rotation matrix in (15) that attains the minima of CCS–DIV. Thus, we set up the resolution of thetas such that $\theta_g \in [-\pi/4 : \theta_g : \pi/4]$, where $\theta_g$ is the grid search, for instance $\theta_g = \pi/8$. Then for each pair $(i, j)$ of the observation data in the range, we find the demixing matrix $W_2$, which attains the minimum of the CCS-DIV. Please refer to Algorithm 3 for more details.

Algorithm 3: The ICA Based on pairwise Jacobi scheme

**Input:** ($M \times T$) matrix of realization $X$, Initial demixing matrix $W = I_M$, number of iterations $itr$, Step Size $\gamma$, e.g., $\gamma = 0.3$, $\alpha$, e.g., $\alpha = -0.99999$  

**Perform Pre-Whitening**  

$[X = V \ast X = A^{(c−1/2)}E^T \ast X]$,

**While** (True)  

**For loop:** for each $i = 1 \ldots M − 1$  

**For loop:** for each $j = i + 1 \ldots M$  

If $CM([i, j], [i, j]) == 0$  

**Continue;**  

**End While**  

**For loop:** For each $\theta_1 = \pi/4$, $\pi/8$, $\pi/4$  

$$W_2 = \begin{bmatrix} \cos \theta_1 & -\sin \theta_1 \\ \sin \theta_1 & \cos \theta_1 \end{bmatrix}$$  

Evaluate $D_c(X([i, j], :), W_2 \ast X([i, j], :), \alpha)$ For all $i = 1, \ldots, T$.  

**End For**

Find $W_2 = \min W_2 \ast X$, $\theta_1$  

**Initial rotational matrix** $R = I_M$,  

**Update rotational matrix** $R([i, j], [i, j]) = W_2$  

**Update Demixing matrix** $W = R \ast W$  

**Update Convergence matrix** $CM([i, j], [i, j]) = \theta_1 \ast 180/\pi$  

**End For**  

**End For**  

**End while loop** If $sum(CM) <= 1$  

**Output:** Demixing matrix $W$.

3.3 Computational complexity

Given T realizations of M observation signals, the computational complexity of the proposed algorithms rely on T and the number of observation signals M, and is approximately given by $O\left(\frac{M(M-1)}{2}T^2\right)$. The computational complexity has been a measure of merit for ICA algorithms. With the advent
of Graphics Processing Units (GPUs) (see www.Nvidia.com, e.g.), and more powerful computing platforms, however performance accuracy holds more merit. In our comparison among the ICA algorithms, we employ several metrics including computational time and accuracy. We also employ adaptive sampling techniques that improve the performance in terms of both metrics (accuracy and computational load).

The presented technique samples the signal into small time blocks in order to evaluate the integration of the proposed divergence and reduce the computational complexity. Thus, we have introduced a sampling factor $T_s$ to evaluate the proposed divergence at each $T_s$ instance. Therefore, the computational complexity of the proposed algorithm is reduced by the square of the sample factor $T_s$ to be less than $O\left(\frac{M(M-1)}{2} \left(\frac{T_s}{T_i}\right)^2\right)$. Namely, we quantize the specific area of integration of the proposed divergence into equal $\left(\frac{T_s}{T_i}\right)$ segments to evaluate the proposed divergence.

### 4 Performance evaluation of the proposed CCS-ICA algorithms versus the existing ICA-based algorithms

In this section, Monte Carlo Simulations are carried out. It is assumed that the number of sources is equal to the number of observations “i.e., sensors”. All algorithms have used the same whitening method. The simulations have been carried out using the MATLAB software on an Intel Core i5 CPU 2.4-GHz processor and 4G MB RAM. Each entry in the forthcoming tables corresponds to the average of corresponding trial “independent Monte Carlo” runs in which the mixing matrix is randomly chosen.

Firstly, we start with the $2 \times 2$ mixture matrix case as a baseline for verifying the performance of the presented algorithms and thoroughly studying the impact of various classes of source signals, namely, uniform distributions, Laplacian distributions, Rayleigh distributions and log-normal distributions, on the performance of the proposed algorithm.

We compare the performance of the ICA algorithms based on the CCS–DIV, CS-DIV, E-DIV, KL–DIV, and C-DIV with $\alpha = 1$ and $\alpha = -1$ for the Algorithm 3 scheme. We also compare it with other benchmark algorithms such as FastICA$^1$ (Hyvarinen 1999), RobustICA$^2$ (Zarzoso and Comon 2010), JADE$^3$ (Cardoso 1999) and RapidICA$^4$ (Yokote and Matsuyama 2012). For these methods, the default setting parameters are used according to their toolboxes and their publications.

During the comparison, we use the bandwidth as a function of sample size, namely, $h = 1.067\frac{T}{M}$ (Duda et al. 2001). The demixing matrix has been initialized as an identity i.e., $W = I_M$ for all algorithms. Note that CCS2 and CCS3 represent Algorithms 2 and 3, respectively. In addition, the minus and plus signs represent $\alpha = 1$ and $\alpha = -1$ cases, respectively. Tables $1$, $2$ and $3$ summarize the performance of the proposed non-parametric ICA algorithms “CCS2 and CCS3” against other aforementioned algorithms. In this task, our goal is to separate mixtures of two sub-Gaussians, two sup-Gaussians, and both sub and sup-Gaussian signals.

### Table 1 The performance of the ICA algorithm based on the proposed divergence and other widely used ICA algorithms in terms of the Amari error (multiplied by 100)

| Source | Samples | Trials | FastICA | JADE | RobustICA | Rapid ICA | IK-DIV | CS-DIV |
|--------|---------|--------|---------|------|-----------|-----------|--------|--------|
| $s_1, s_1$ | 1000 | 100 | 6.16 | 4.77 | 5.27 | 5.07 | 3.32 | 2.66 |
| $s_2, s_2$ | 1000 | 100 | 22.34 | 18.51 | 28.29 | 20.26 | 6.78 | 7.39 |
| $s_3, s_3$ | 1000 | 100 | 2.45 | 2.10 | 2.24 | 2.14 | 2.31 | 2.21 |
| $s_4, s_4$ | 1000 | 100 | 3.34 | 3.03 | 3.13 | 3.29 | 1.93 | 2.02 |
| $s_5, s_5$ | 1000 | 100 | 5.11 | 4.53 | 5.39 | 5.17 | 2.44 | 2.07 |

Each entry averages over the corresponding number of trials. Observation mixtures consist of two source signals that follow the same distribution as denoted in the corresponding example.

---

1. [http://www.cis.hut.fi/projects/ica/fastica/code/dlcod e.html](http://www.cis.hut.fi/projects/ica/fastica/code/dlcod e.html).
2. [http://www.i3s.unice.fr/~zarzos ro/robustica.html](http://www.i3s.unice.fr/~zarzos ro/robustica.html).
3. [http://www.tsi.enst.fr/icacentral/algos.html](http://www.tsi.enst.fr/icacentral/algos.html).
4. [http://dx.doi.org/10.4236/jsip.2012.33037](http://dx.doi.org/10.4236/jsip.2012.33037).
Specifically, we use the following distributions: For the sub-Gaussian distribution, we use (i) the uniform distribution as
\[ p(s_1) = \begin{cases} \frac{1}{2r_1} & \text{if } -r_1 < s_1 < r_1 \\ 0 & \text{otherwise} \end{cases} \]
and (ii) the Rayleigh distribution, i.e.,
\[ p(s_2) = s_2 \exp \left( -\frac{s_2^2}{2} \right) \]
For the super-Gaussian distribution, we use (i) the Laplacian distribution
\[ p(s_3) = \frac{1}{2\tau_2} \exp \left( -\frac{|s_3|}{\tau_2} \right) \]
and (ii) the log-normal distribution, i.e.,
\[ p(s_4) = \exp \left( -\frac{(\log(s_4))^2}{2} \right) \]
Also, data samples, \( T = 1000 \), are selected and randomly generated by using \( \tau_1 = 3 \) and \( \tau_2 = 1 \) in the above. Kurtoses for all aforementioned signals are \(-1.2, -2.99, -0.7224, \) and \( 8.4559 \) respectively, and they are evaluated using \( Kurt(s) = \frac{\text{Var}[s]}{\text{Var}[e]^2} - 3 \). Furthermore, a different, randomly generate dmixing matrices are used to generate the mixtures.

One can observe several patterns from Tables 1, 2 and 3. The presented algorithms based on the proposed measure show the best performance in terms of accuracy (in most cases) and repeatability (in terms of variance). The proposed algorithm CCS3 exhibits comparable behavior in terms of speed and stability with KL and ED. Clearly, the proposed divergence improves on the CS-DIV in terms

### Table 2
The computational load, in seconds, of the ICA algorithm based on the proposed divergence and other widely used ICA algorithms, each entry averages over the corresponding number of trials.

| Source | Samples | Trials | FastICA | JADE | RobustICA | Rapid ICA | IK-DIV | CS-DIV |
|--------|---------|--------|---------|------|-----------|-----------|-------|-------|
| \( s_1, s_1 \) | 1000 | 100 | 0.0 | 0.0 | 0.0 | 0.0 | 20.1 | 22.1 |
| \( s_2, s_2 \) | 1000 | 100 | 0.0 | 0.1 | 0.0 | 0.0 | 20.1 | 21.3 |
| \( s_3, s_3 \) | 1000 | 100 | 0.0 | 0.0 | 0.0 | 0.0 | 19.1 | 20.7 |
| \( s_4, s_4 \) | 1000 | 100 | 0.0 | 0.1 | 0.0 | 0.0 | 20.4 | 24.3 |
| \( s_1, s_3 \) | 1000 | 100 | 0.0 | 0.0 | 0.0 | 0.0 | 20.2 | 20.1 |

### Table 3
The corresponding variance of the performance

| Source | Samples | Trials | FastICA | JADE | RobustICA | Rapid ICA | IK-DIV |
|--------|---------|--------|---------|------|-----------|-----------|-------|
| \( s_1, s_1 \) | 1000 | 100 | 11.02 | 12.07 | 38.05 | 11.74 | 0.72 |
| \( s_2, s_2 \) | 1000 | 100 | 102.53 | 211.75 | 332.76 | 95.06 | 0.78 |
| \( s_3, s_3 \) | 1000 | 100 | 1.11 | 1.80 | 1.71 | 1.27 | 0.86 |
| \( s_4, s_4 \) | 1000 | 100 | 18.47 | 15.34 | 17.44 | 14.64 | 0.83 |
| \( s_1, s_3 \) | 1000 | 100 | 13.91 | 12.88 | 13.90 | 14.16 | 1.37 |

Also, data samples, \( T = 1000 \), are selected and randomly generated by using \( \tau_1 = 3 \) and \( \tau_2 = 1 \) in the above. Kurtoses for all aforementioned signals are \(-1.2, -2.99, -0.7224, \) and \( 8.4559 \) respectively, and they are evaluated using \( Kurt(s) = \frac{\text{Var}[s]}{\text{Var}[e]^2} - 3 \). Furthermore, a different, randomly generate dmixing matrices are used to generate the mixtures.

One can observe several patterns from Tables 1, 2 and 3. The presented algorithms based on the proposed measure show the best performance in terms of accuracy (in most cases) and repeatability (in terms of variance). The proposed algorithm CCS3 exhibits comparable behavior in terms of speed and stability with KL and ED. Clearly, the proposed divergence improves on the CS-DIV in terms
of repeatability and performance. Notably, most the presented divergences struggle to separate the Rayleigh distributions \(\mathcal{R}(\mu_1^2, \mu_2^2)\) (including the KL-DIV) except the proposed divergence and C-DIVs. Moreover, Table 3 shows the variance of the performance of the proposed algorithm outperforms the CS-DIV and renders the divergence more robust against variation in parameters. It is also worthwhile to represent the average performance of each method in Fig. 1.

Also, it is noted that the non-parametric methods result in better performance and repeatability than methods such as JADE, Fast ICA and other algorithms. Nevertheless JADE performs better than each of Fast ICA, Robust ICA and Rapid ICA in terms of accuracy in some cases. However, in terms of speed, we find that these later algorithms outperform the JADE algorithm, especially rapid ICA and Robust ICA.

Secondly, an extensive analytical study is carried out to evaluate and show the performance of the proposed algorithm and other algorithms for the high dimensional case (i.e., mixture of more than 2 sources). To that end, we form five groups based on the number of source signals in the mixtures, namely, 2, 4, 8, 16 and 20. Then, we implement our proposed algorithms and other hosted algorithms on each corresponding group using various sample size to show the impact of the sample size on their performance.

A new set of randomly generated source signals (refer to Table 4) and mixing matrices are generated for the next comparative case study. Table 5 summarizes the performance of the aforementioned algorithms in a more complex separation process for mixtures of multiple sources. We use the label dimension to identify the number of signal sources in Table 5. In a nutshell, Table 5 summarizes the performance of each algorithm in terms of the standard Ammari error metric (multiplied \(\times 100\)). All results have been averaged over a number of independent Monte Carlo runs.

Several patterns can be observed from Table 5. First, the non-parametric ICA algorithms attain the best performance in terms of accuracy. However, in terms of speed, Rapid ICA, Fast ICA, Robust ICA and JADE perform better. Second, the non-parametric ICA based on the proposed divergence (CCS3) provides the best performance in terms of accuracy (in most cases). Third, the performance of the

| Signals’ Notation | Kurtosis |
|-------------------|----------|
| \(s_1\)           | -1.2116  |
| \(s_2\)           | 2.9324   |
| \(s_3\)           | -1.3995  |
| \(s_4\)           | 136.0108 |
| \(s_5\)           | 11.6452  |
| \(s_6\)           | 4.219    |
| \(s_7\)           | -1.2065  |
| \(s_8\)           | 3.1965   |
| \(s_9\)           | 3.4302   |
| \(s_{10}\)        | -1.3049  |
| \(s_{11}\)        | -1.6805  |
| \(s_{12}\)        | 0.65419  |
| \(s_{13}\)        | 0.33421  |
| \(s_{14}\)        | 1.6935   |
| \(s_{15}\)        | 0.86239  |
| \(s_{16}\)        | 0.60566  |
| \(s_{17}\)        | 0.75488  |
| \(s_{18}\)        | 0.65645  |
| \(s_{19}\)        | 0.81022  |
| \(s_{20}\)        | 0.7692   |
| \(s_{21}\)        | 0.27737  |
| \(s_{22}\)        | -0.56816 |
Table 5  The performance of the ICA algorithm based on the proposed divergence and other widely used ICA algorithms in terms of Amari error (Cichocki and Amari 2002) (multiplied by 100)

| Dimensions \ Samples | Trials | JADE | FastICA | RapidICA | RobustICA | CS | CDIV | KLDIV | CCS2 | CCS3 |
|-----------------------|--------|------|---------|----------|-----------|----|------|-------|------|------|
| 2                     | 1000   | 512  | 5.6     | 7.3      | 6.1       | 7.2| 2.5  | 2.2   | 2.3  | 2.1  | 2.0  |
|                       | 2000   | 512  | 5.1     | 5.9      | 5.5       | 6  | 1.9  | 1.7   | 1.7  | 1.8  | 1.8  |
|                       | 4000   | 512  | 3.1     | 4.1      | 3.5       | 4.3| 1.7  | 1.6   | 1.5  | 1.6  | 1.4  |
|                       | 8000   | 512  | 2.4     | 2.6      | 2.5       | 2.6| 1.3  | 1.2   | 1.1  | 1.4  | 1.1  |
| 4                     | 1000   | 200  | 8       | 9.7      | 9.1       | 9.8| 3.0  | 2.4   | 3.1  | 3.1  | 2.5  |
|                       | 2000   | 200  | 5.4     | 7.3      | 6.5       | 7.2| 2.4  | 2.2   | 2.1  | 2.5  | 1.8  |
|                       | 4000   | 200  | 4.2     | 4.2      | 4.1       | 4.3| 1.7  | 1.4   | 1.4  | 1.4  | 1.6  |
|                       | 8000   | 200  | 2.1     | 2.7      | 2.5       | 2.7| 1.4  | 1.2   | 1.3  | 1.4  | 1.2  |
| 8                     | 1000   | 75   | 10.5    | 10.3     | 9.6       | 11.2| 4.6  | 3.6   | 4.2  | 4.4  | 3.2  |
|                       | 2000   | 75   | 8.1     | 8.0      | 7.6       | 8.2| 3.5  | 3.1   | 3.3  | 3.2  | 3.2  |
|                       | 4000   | 75   | 5.7     | 4.1      | 4.4       | 4.9| 2.5  | 2.3   | 2.7  | 2.6  | 2.8  |
|                       | 8000   | 75   | 2.7     | 3.1      | 3.0       | 3.2| 2.3  | 2.1   | 2.1  | 2.1  | 1.9  |
| 16                    | 1000   | 15   | 8       | 9.7      | 9.1       | 9.8| 6.7  | 6     | 6.7  | 7.3  | 5.5  |
|                       | 2000   | 15   | 5.4     | 7.3      | 6.5       | 7.2| 6.1  | 5.2   | 6    | 6.9  | 5.1  |
|                       | 4000   | 15   | 4.2     | 4.2      | 4.1       | 4.3| 5.4  | 4.4   | 5.1  | 5.6  | 4.2  |
|                       | 8000   | 15   | 2.1     | 2.7      | 2.5       | 2.7| 3.6  | 2.6   | 3.1  | 3.8  | 2.9  |
| 20                    | 1000   | 5    | 22.3    | 21.1     | 20.1      | 26.2| 14.1 | 9.1   | 10.1 | 13.1 | 8.9  |
|                       | 2000   | 5    | 15.7    | 15.6     | 15.2      | 16.2| 7.7  | 6.7   | 7.3  | 8.3  | 7.2  |
|                       | 4000   | 5    | 7.8     | 7.2      | 7.1       | 7.2| 6.2  | 7.6   | 6.4  | 6.7  | 5.3  |
|                       | 8000   | 5    | 4.5     | 4.1      | 3.9       | 4.0| 2.7  | 2.2   | 2.6  | 4.4  | 2.3  |

Each entry averages over the corresponding number of trials.

Table 6  The performance of the ICA algorithm based on the proposed divergence in terms of the Amari error (multiplied by 100)

| Dimensions $M$ \ Samples $T$ | Trials | $CCS3$ at $0.1T$ | $CCS3$ at $0.01T$ | $CCS3$ at $0.001T$ | $CCS3$ at $1T$ |
|-----------------------------|--------|------------------|------------------|-------------------|----------------|
| 2                           | 1000   | 1024             | 4.6              | 2.9               | 2.1            | 2.0            |
|                             | 2000   | 1024             | 3.6              | 2.3               | 1.9            | 1.8            |
|                             | 4000   | 1024             | 2.8              | 1.9               | 1.6            | 1.4            |
|                             | 8000   | 1024             | 2.2              | 1.6               | 1.1            | 1.2            |
| 4                           | 1000   | 250              | 5.8              | 3.8               | 2.4            | 2.5            |
|                             | 2000   | 250              | 5                | 2.9               | 2              | 1.8            |
|                             | 4000   | 250              | 3.5              | 2.5               | 1.6            | 1.6            |
|                             | 8000   | 250              | 2.7              | 2.2               | 1.3            | 1.3            |
| 8                           | 1000   | 100              | 5.6              | 3.8               | 2.5            | 3.2            |
|                             | 2000   | 100              | 3.7              | 3.1               | 2.2            | 3              |
|                             | 4000   | 100              | 3.1              | 2.6               | 2.2            | 2.8            |
|                             | 8000   | 100              | 3.0              | 2.2               | 1.9            | 1.9            |
| 16                          | 1000   | 25               | 20.5             | 15.8              | 8.6            | 5.5            |
|                             | 2000   | 25               | 12.6             | 10.1              | 7              | 5.1            |
|                             | 4000   | 25               | 8.6              | 8                 | 4.5            | 4.2            |
|                             | 8000   | 25               | 5.8              | 3.9               | 1.9            | 2.9            |
| 20                          | 1000   | 10               | 27.7             | 15.1              | 13.7           | 8.9            |
|                             | 2000   | 10               | 22.8             | 11.3              | 12             | 7.2            |
|                             | 4000   | 10               | 15.6             | 9                 | 7.2            | 5.3            |
|                             | 8000   | 10               | 9.8              | 6.3               | 3              | 2.3            |

Each entry averages over the corresponding number of trials.
new algorithms perform more consistently and exhibits performance improvement as the sample size increases. Lastly, in terms of speed, Rapid ICA, Fast ICA, Robust ICA and JADE perform better. Thus, these latter algorithms could be chosen to initialize the process of the proposed algorithms in order to reduce the overall computational load. Since the comparison between the ICA algorithms has relied on two criteria, namely, accuracy and computational load, a trade-off between these two criteria has always been assessed for each targeted application. We also note that with the advent of powerful computing platforms including Graphics Processing Units (GPUs), computational load/speed becomes less of a factor, and the true metric becomes accuracy or quality of estimates.

Furthermore, Table 6 summarizes the performance of CCS-ICA (see Algorithm 3) based on the proposed divergence and other widely used ICA algorithms, each entry averages over the corresponding number of trials.

| Dimensions $M$ | Samples $T_s$ | Trials | CCS3 at 0.1 | CCS3 at 0.01 | CCS3 at 0.001 |
|---------------|---------------|--------|-------------|--------------|---------------|
| 2             | 1000          | 1024   | 0.4         | 2.8          | 29.8          |
|               | 2000          | 1024   | 0.5         | 4.8          | 44.8          |
|               | 4000          | 1024   | 0.8         | 8            | 77.9          |
|               | 8000          | 1024   | 1.5         | 10.6         | 137           |
| 4             | 1000          | 250    | 1.8         | 24           | 218.1         |
|               | 2000          | 250    | 4.3         | 39           | 344.8         |
|               | 4000          | 250    | 5.9         | 47.9         | 593.4         |
|               | 8000          | 250    | 10.2        | 83.6         | 1105          |
| 8             | 1000          | 100    | 19.3        | 128.7        | 1053          |
|               | 2000          | 100    | 31.5        | 201.7        | 1743          |
|               | 4000          | 100    | 46.5        | 266.4        | 3109          |
|               | 8000          | 100    | 74.2        | 241.8        | 5534          |
| 16            | 1000          | 25     | 170.6       | 909.5        | 6282          |
|               | 2000          | 25     | 242.3       | 1171         | 9320          |
|               | 4000          | 25     | 305.5       | 1403         | 14,717        |
|               | 8000          | 25     | 329.9       | 2297         | 25,658        |
| 20            | 1000          | 10     | 339         | 1195.7       | 9605          |
|               | 2000          | 10     | 427.4       | 1724.2       | 14,708        |
|               | 4000          | 10     | 607.6       | 2398.3       | 23,634        |
|               | 8000          | 10     | 900         | 3754.5       | 42,538        |

5 Conclusion

Two schemes of pairwise non-parametric ICA algorithms are derived and presented based on the CCS–DIV to solve the problem of high dimensional data in BSS. Extensive comparative case-studies are carried out to show that the presented methods outperform many of the presently known methods. We provide a comparative Monte–Carlo metric performance with a host of leading ICA algorithms. We employ adaptive sampling technique that samples the signal into small time blocks to evaluate the integration of the CCS–DIV and reduce the computational complexity.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Albataineh, Z. (2018aa). Blind decoding of massive MIMO uplink systems based on the higher order cumulants. Wireless Personal Communications, 103, 1835–1847.
Albataineh, Z. (2018bb). Robust blind channel estimation algorithm for linear STBC systems using fourth order cumulant matrices. Telecommunications and System, 68, 573–582.
Albataineh, Z., Hayajneh, K., Salameh, H., Dang, C., & Dagmseh, A. (2020). Robust massive MIMO channel estimation for 5G
networks using compressive sensing technique. AEU—International Journal of Electronics and Communications, 120, 153197.

Albataineh, Z., & Salem, F. M. (2017). Adaptive blind CDMA receivers based on ICA filtered structures. Circuits Systems and Signal Processing, 36, 3320–3348.

Bataineh, Z., & Salem, F. (2018). A convex Cauchy-Schwarz divergence measure for blind source separation. International Journal of Circuits, Systems and Singal Processing, 12, 94–104.

Boscolo, R., Pan, H., & Roychowdhury, V. P. (2004). Independent component analysis based on nonparametric density estimation. IEEE Transactions on Neural Networks, 15(1), 55–65.

Cardoso, Jean-François. (1999). High-order contrasts for independent component analysis. Neural Computation, 11(1), 157–192.

Cardoso, J., & Souloumiac, A. (1993). Blind beamforming for non-Gaussian signals. IEEE Proceedings F Radar and Signal Processing, 140(6), 362–370.

Chen, Y. (2005). Blind separation using convex function. IEEE Transactions on Signal Processing, 53(6), 2027–2035.

Chien, J.-T., & Chen, B.-C. (2006). A new independent component analysis for speech recognition and separation. IEEE Transactions on Audio, Speech, and Language Processing, 14(4), 1245–1254.

Cichocki, A., & Amari, S. (2002). Adaptive Blind Signal and Image Processing. Chichester, UK: Wiley.

Cichocki, A., Zdunek, R., & Amari, S. (2006). Csiszár’s divergences for non-negative matrix factorization: Family of new algorithms. In 6th International Conference on Independent Component Analysis and Blind Signal Separation, Charleston SC, USA, March 5–8, 2006. Springer LCNS 3889, pp. 32–39.

Comon, P. (1994). Independent component analysis, a new concept? Signal Processing (Special Issue on Higher-Order Statistics), 36(3), 287–314.

Comon, P., & Jutten, C. (Eds.). (2010). Handbook of blind source separation independent component analysis and applications. Oxford: Academic Press.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern classification, chapter 4 (pp. 2–18). New York: Wiley.

Golub, G. H., & Van Loan, C. F. (1996). Matrix computations (3rd ed.). Baltimore, MD: Johns Hopkins Univ. Press.

Hsieh, H.-L., & Chien, J.-T. (2010). A new nonnegative matrix factorization for independent component analysis. In Proceedings of the International Conference Acoustics, Speech, and Signal Processing (pp. 2026–2029).

Hyvarinen, A. (1999). Fast and robust fixed-point algorithm for independent component analysis. IEEE Transactions on Neural Network, 10(3), 626–634.

Jan, T., Wenwu, W., & DeLiang, W. (2009). A multistage approach for blind separation of convolutive speech mixtures. Acoustics, Speech and Signal Processing (ICASSP), 1713–1716.

Jenssen, R., Erdogmus, D., Hild, K., Principe, J., & Eltoft, T. (2005). Optimizing the Cauchy-Schwarz PDF distance for information theoretic, nonparametric clustering. In Energy Minimization Methods in Computer Vision and Pattern Recognition (pp. 34–45). Springer.

Jen-Tzung Chien, H.-L. H. (2012). Convex divergence ICA for blind source separation. IEEE Transactions on Audio, Speech, and Language Processing, 20(1), 302–313.

Kampa, K., Hasanbelliu, E., & Principe, J. C. (2011). Closed-form Cauchy-Schwarz pdf divergence for mixture of Gaussians. In International Joint Conference on Neural Networks (IJCNN) (pp. 2578–2585).

Lin, J. (1991). Divergence measures based on the Shannon entropy. IEEE Transactions on Information Theory, 37(1), 145–151.

Matsuyama, Y., Katsumata, N., Suzuki, Y., & Imahara, S. (2000). The $a$-ICA algorithm. In Proceedings of the International Workshop on Independent Component Analysis and Blind Signal Separation (pp. 297–302).

Na, Y., & Yu, J. (2012). Kernel and spectral methods for solving the permutation problem in frequency domain BSS. Neural Networks (IJCNN), 1–8.

Rutishauser, H. (1966). The Jacobi method for real symmetric matrices. Numerical Mathematics, 9, 1–10.

Schobben, D., Torkkola, K., & Smaragdis, P. (1999). Evaluation of blind signal separation methods. In Proceedings International Workshop on Independent Component Analysis and Blind Signal Separation (pp. 261–266).

Seth, S., Rao, M., Park, I., & Principe, J. C. (2011). A unified framework for quadratic measures of independence. IEEE Transactions on Signal Processing, 59(8), 3624–3635.

Takeda, R., Nakadai, K., Takahashi, T., Komatani, K., Ogata, T., & Okuno, H. G. (2009). Step-size parameter adaptation of multichannel semi-blind ICA with piecewise linear model for barge-in able robot audition. Intelligent Robots and Systems (pp. 2277–2282).

Takeda, R., Nakadai, K., Takahashi, T., Komatani, K., Ogata, T., & Okuno, H. G. (2010). Upper-limit evaluation of robot audition based on ICA-BSS in multi-source, barge-in and highly reverberant conditions. In IEEE International Conference on Robotics and Automation (ICRA) (pp. 4366–4371).

Xu, D., Principe, J. C., Fisher, J. I. I. I., & Wu, H.-C. (1998). A novel measure for independent component analysis (ICA). Proceedings of the International Conference on Acoustics, Speech and Signal Processing, 2, 1161–1164.

Yokote, R., & Matsuyama, Y. (2012). Rapid algorithm for independent component analysis. Journal of Signal and Information Processing, 3, 275–285.

Yoshioka, T. N., Tomohiro, M., Masato, O., & Hiroshi, G. (2011). Blind separation and dereverberation of speech mixtures by joint optimization. IEEE Transactions on Audio Speech and Language Processing, 19(1), 69.

Zarzoso, V., & Comon, P. (2010). Robust independent component analysis by iterative maximization of the Kurtosis contrast with algebraic optimal step size. IEEE Transactions on Neural Networks, 21(2), 248–261.

Zarzoso, V., Murillo-Fuentes, J. J., Boloix-Tortosa, R., & Nandi, A. K. (2006). Optimal pairwise fourth-order independent component analysis. IEEE Transactions on Signal Processing, 54(8), 3049–3063.

Zhang, J. (2004). Divergence function, duality, and convex analysis. Neural Computing, 16, 159–195.