Pattern Recognition 76 (2018) 752–760

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Independent component analysis by lp-norm optimization

Sungheon Park, Nojun Kwak*

Graduate School of Convergence Science and Technology, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul 08826, Republic of Korea

ARTICLE INFO

Article history:
Received 18 October 2016
Revised 19 September 2017
Accepted 7 October 2017
Available online 13 October 2017

Keywords:
ICA
PCA
l_p-Norm
Maximum likelihood estimation
Super-Gaussian
Sub-Gaussian

ABSTRACT

In this paper, a couple of new algorithms for independent component analysis (ICA) are proposed. In the proposed methods, the independent sources are assumed to follow a predefined distribution of the form $f(s) = \alpha \exp(-\beta|s|^p)$ and a maximum likelihood estimation is used to separate the sources. In the first method, a gradient ascent method is used for the maximum likelihood estimation, while in the second, a non-iterative algorithm is proposed based on the relaxation of the problem. The maximization of the log-likelihood of the estimated source $X^T\mathbf{w}$ given the parameter p and the data X is shown to be equivalent to the minimization of l_p-norm of the projected data $X^T\mathbf{w}$. This formulation of ICA has a very close relationship with the Lp-PCA where the maximization of the same objective function is solved. The proposed algorithm solves an approximation of the l_p-norm minimization problem for both super-($p < 2$) and sub-Gaussian ($p > 2$) cases and shows superior performance in separating independent sources than the state of the art algorithms for ICA computation.

1. Introduction

Finding meaningful projections of observed data is an essential task in multivariate data analysis. For the sake of simplicity, it is often represented as a linear transformation of the observations. If we denote the original observation and its transformation as X and Y, respectively, then the problem is to find the projection matrix W so that $Y = WX$ is a meaningful representation of X in some aspects. Several algorithms have been proposed for this purpose, among which, principal component analysis (PCA) [1] and independent component analysis (ICA) [2,3] are the most widely used techniques to find projections in an unsupervised manner.

The objective of ICA is to decompose the observed multivariate signal into a set of independent components (sources) [2,3]. The technique is originally devised for blind source separation and widely used in the field of signal processing and machine learning such as audio source separation [4,5], handling electroencephalography data [6–8], and feature extraction from image data [9–11]. In ICA, it is assumed that the sources are non-Gaussian, and the observed data are constructed by linear combinations of the sources. Various algorithms have been proposed for ICA [4,12–18], and their extensions or generalizations are also proposed in literature [19–21]. In most algorithms, the unmixing (projection) matrix W is chosen to maximize the non-Gaussianity of the projected data Y, among which, FastICA [12,13] have been the most successful algorithm for more than a decade.

On the other hand, classical principal component analysis (PCA), which will be termed as L2-PCA in what follows, tries to maximize the variance of the projected data Y [1]. In other words, it finds the projection vectors which maximize l_2-norm of Y. In L2-PCA, the projection vectors can be easily computed using eigenvalue decomposition. However, L2-PCA is prone to outliers like other l_2-norm based optimization schemes. To resolve this problem, PCA algorithms based on l_1-norm in the projected space, which are robust to outliers, have been proposed [22,23]. Recently, in [24], an l_p-norm extension of [23] has also been proposed for an arbitrary positive number p.

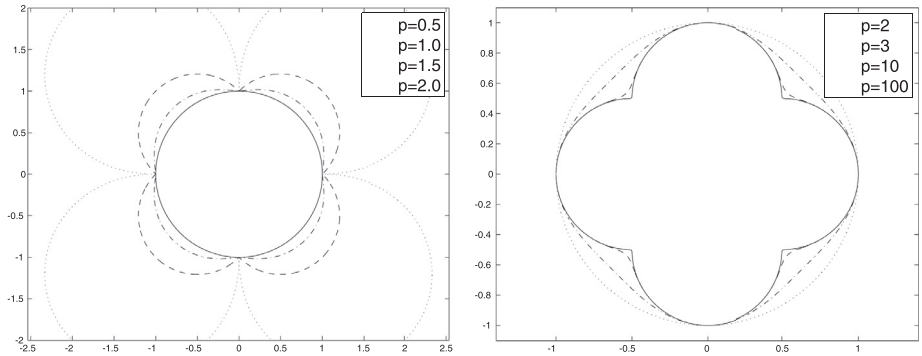
Despite the similarity between the concepts of PCA and ICA, the connection between them has been hardly explored. Interestingly, recent paper by Martín-Clemente and Zarzoso [25] revealed that ICA can be viewed as applying PCA-L1[1] on the whitened data. Inspired by this, a couple of novel methods for ICA using Lp-norm optimization technique is proposed in this paper.

Though FastICA shows satisfactory result, the performance depends highly on the choice of contrast function. By exploiting the connection of ICA with l_p-norm optimization, thus with Lp-PCA [24], the formulation of ICA becomes simple and intuitive. Although the computational complexity of the proposed algorithm

* Corresponding author.
E-mail addresses: sungheonpark@snu.ac.kr (S. Park), nojunk@snu.ac.kr (N. Kwak).

[1] In this paper, L1-PCA refers to the method which minimizes the l_1 reconstruction error in input space, while PCA-L1 denotes the one that maximizes the l_1-norm in feature space.

https://doi.org/10.1016/j.patcog.2017.10.006
0031-3203/



(a) Super-Gaussian sources ($p < 2$) (b) Sub-Gaussian sources ($p > 2$)

Fig. 1. l_p-norm ($||\mathbf{y}||_p$) of points on a unit circle ($||\mathbf{y}||_2 = 1$) for different p values. The distance of a point from the origin is the l_p-norm of the unit vector with the same direction. For $p < 2$, the direction of minimum norm is axis aligned, which corresponds to the sparsest $\mathbf{y}$ direction. For $p > 2$, it corresponds to $\{\pm 1\}^2$, the densest $\mathbf{y}$ direction.

is $\mathcal{O}(n^2)$ where n is the number of samples, it is fast because it is a non-iterative, deterministic algorithm. Simulation results show that the reconstruction performance of the proposed algorithm is also superior to the state of the art algorithms for ICA computation (FastICA and RobustICA [18]) especially for the super-Gaussian sources. One drawback of FastICA is that it requires many iterations to converge or sometimes fails to converge when the number of samples is quite small. On the other hand, the proposed method converges fast even for a small number of samples.

2. A brief review of ICA

Suppose a d-dimensional zero-mean, unit-variance random vector $\mathbf{s} = [s_1, \cdots s_d]^T$ denotes a set of d independent sources and the matrix $S = [\mathbf{s}_1, \cdots, \mathbf{s}_n] \in \mathbb{R}^{d\times n}$ is composed of n samples of $\mathbf{s}$.[2] The observation matrix X_0 is assumed to be a linear combination of source data through a $d \times d$ mixing matrix A_0, i.e. $X_0 = A_0 S$.

In most ICA algorithms, to obtain the unknown sources S from the observation X_0, the data X_0 is firstly transformed to its sphered (or whitened) version X such that its sample covariance matrix becomes an identity matrix, i.e., $\frac{1}{n}XX^T = I_d$. This sphered data can be obtained by the scaled version of L2-PCA as $X = \Lambda^{-\frac{1}{2}}U^T X_0$ where Λ and U are the eigenvalue and eigenvector pairs of the sample covariance matrix $\frac{1}{n}X_0X_0^T$.

Now, the goal of ICA becomes finding an unmixing matrix W such that

$$Y = WX = W_0X_0 = PS, \tag{1}$$

where $P \in \mathbb{R}^{d\times d}$ is a permutation matrix and $W_0 \triangleq W\Lambda^{-\frac{1}{2}}U^T$. Note that W is restricted to be an orthogonal matrix since

$$WW^T = WE[\mathbf{x}\mathbf{x}^T]W = E[\mathbf{y}\mathbf{y}^T] = PE[\mathbf{s}\mathbf{s}^T]P^T = I_d. \tag{2}$$

Here, $E(\cdot)$ is the expectation operation which can be replaced by the sample mean. In (2), the first equality is from the whitening process and the last equality is by the assumption that the sources are independent with zero mean and unit variance.

The sources for ICA problem are assumed to be non-Gaussian. Kurtosis and negative entropy (negentropy) are often used to measure the non-Gaussianity of a distribution. Kurtosis is defined as the fourth order cummulant. A distribution with positive kurtosis is called super-Gaussian, while the one with negative kurtosis is called sub-Gaussian. Negentropy measures the difference of entropy compared to the Gaussian distribution. It is known to be statistically more robust but computationally more intensive than kurtosis since estimation of probability density function is needed. Here, we briefly describe two representative ICA algorithms, FastICA [12] and RobustICA [18], which exploit negentropy and kurtosis respectively. These methods will be used in the experiments for performance comparison with the method proposed in this paper.

FastICA Hyvärinen [12] suggested a one-unit contrast function $\mathcal{J}_G$ which approximates negentropy as follows:

$$\mathcal{J}_G(\mathbf{w}) = (E[G(\mathbf{w}^T\mathbf{x})] - E[G(v)])^2 \tag{3}$$

where G is a nonquadratic function, and v represents the Gaussian random variable with zero mean and unit variance. The choice of function G is important for both robustness and convergence of FastICA algorithm [12]. A typical choice of G is either $G(y) = \frac{1}{a}\log\cosh(ay)$ with $1 \le a \le 2$ or $G(y) = -\exp(\frac{y^2}{2})$. A fixed-point algorithm which iterates the following two steps until convergence is used for the maximization of (3):

$$\mathbf{w} \leftarrow E[\mathbf{x}g(\mathbf{w}^T\mathbf{x})] - E[g'(\mathbf{w}^T\mathbf{x})]\mathbf{w}, \qquad \mathbf{w} \leftarrow \frac{\mathbf{w}}{||\mathbf{w}||}. \tag{4}$$

Here, g and g' are the first and the second derivatives of G, respectively.

RobustICA Recently, Zarzoso and Comon [18] suggested RobustICA which use the kurtosis as the contrast function. It maximizes the absolute value of the following kurtosis:

$$\mathcal{K}(\mathbf{w}) = \frac{E[|\mathbf{w}^T\mathbf{x}|]^4 - 2E[|\mathbf{w}^T\mathbf{x}|^2]^2 - |E[(\mathbf{w}^T\mathbf{x})^2]|^2}{E[|\mathbf{w}^T\mathbf{x}|^2]^2} \tag{5}$$

The method performs exact line search on the contrast function. An optimal step-size that maximizes (5) is found by solving the roots of fourth-degree polynomial. The authors argued that RobustICA has less computational cost and faster convergence compared to FastICA. RobustICA can also be applied to complex-valued signals.

Besides RobustICA and FastICA, various ICA algorithms have been proposed in the literature. For instance, Lee et al. [14] extended infomax algorithm for ICA. Theis et al. [26] developed linear geometric ICA based on histograms.

None of the above algorithms exploit the connection between ICA and l_p-norm optimization. A novel algorithm of ICA based on the l_p-norm optimization is proposed in this paper. In Section 3, we will explain how ICA can be converted to the l_p-norm optimization problem. Then, the actual algorithm for slow and fast version is proposed in Sections 4 and 5 respectively.

[2] In this paper, a scalar, a vector, and a matrix will be denoted as a lower-case letter (e.g., x), a bold-faced lower-case letter (e.g., $\mathbf{x}$) and an upper-case letter (e.g., X), respectively. The same notations will be used for both a random variable (or a random vector) and its sample whose meaning will be obvious in the context.

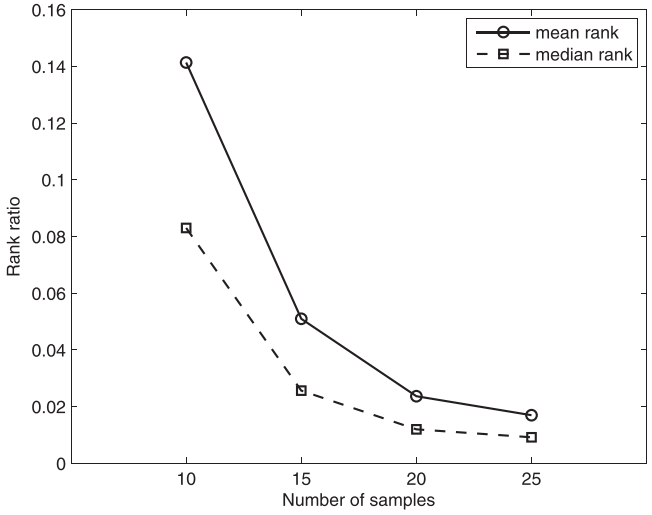


Fig. 2. Mean and median rank ratios of greedy algorithm Lp-ICA-F-. As the number of samples increases, the greedy solution gets closer to the global optima.

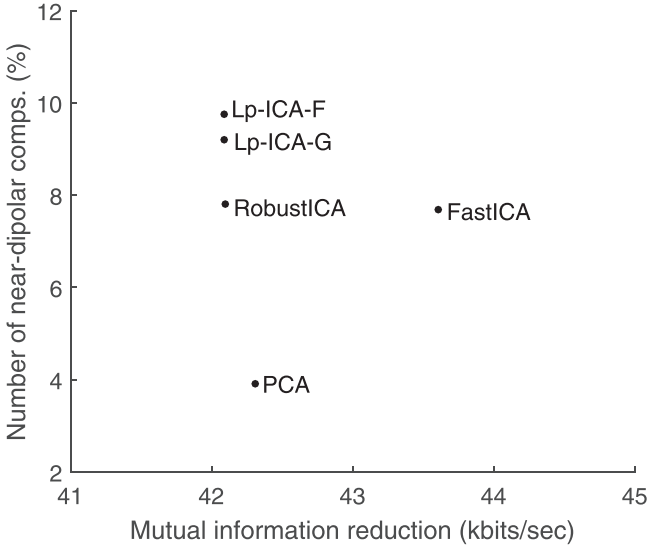


Fig. 3. Performance comparison of ICA algorithms on EEG dataset. Mutual information reduction and the portion of near-dipolar components after decomposition. Higher values implies more independence between decomposed sources.

3. ICA By l_p-norm minimization

3.1. Problem formulation

Our key observation is that as in [27–29], maximum likelihood estimation (MLE) can be directly used to separate sources rather than measuring kurtosis or negentropy. In our approach, as in [17], we consider the following generalized Gaussian distribution with zero-mean and unit-variance as the true source distribution:

$$P(s) = \alpha \exp(-\beta |s|^p). \tag{6}$$

Here, $\alpha > 0$ is a constant that makes $P(s)$ a probability distribution, and $\beta > 0$ is a constant that determines the width of the distribution. We chose the generalized Gaussian distribution because it is a parametric distribution family that covers a wide variety of typical real-world unimodal and symmetric distributions. Considering that the sources are assumed to have unit variance, α and β purely depend on p and can be computed as $\alpha(p) = \frac{\beta p}{2\Gamma(1/p)}$ and $\beta(p) = \sqrt{\frac{\Gamma(3/p)}{\Gamma(1/p)}}$, where $\Gamma(\cdot)$ is a gamma function [17].

The interesting thing is that by changing p, the distribution can either be super-Gaussian or sub-Gaussian. More specifically, it becomes super-Gaussian if $p < 2$, sub-Gaussian if $p > 2$, and Gaussian if $p = 2$. Hyvärinen [12] and Koldovsky [17] also mentioned this function can be used as the contrast function, but the FastICA algorithm actually use the approximation of this kind of function as the contrast function. The proposed scheme in this section does not require the measurement of kurtosis or negentropy.

Assume a source s is distributed as (6) and the whitened data $X = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$ are given. In ICA, we are to reconstruct s by $y = \mathbf{w}^T\mathbf{x}$ using an appropriate unmixing vector $\mathbf{w}$ with the constraint $||\mathbf{w}||_2 = 1$.[3] Then, the likelihood of $\mathbf{w}$ and p given X is

$$\mathcal{L}(\mathbf{w}, p|X) = P(\mathbf{y}; p) = \prod_{i=1}^{n} \alpha \exp(-\beta |\mathbf{w}^T\mathbf{x}_i|^p) \tag{7}$$

where $\mathbf{y} = X^T\mathbf{w} = [y_1, \cdots, y_n]^T$ consists of n independent and identically distributed (i.i.d) samples. Taking the log of (7), the log-likelihood becomes

$$\log\mathcal{L}(\mathbf{w}, p|X) = n\log\alpha(p) - \beta(p) \sum_{i=1}^{n} |\mathbf{w}^T\mathbf{x}_i|^p. \tag{8}$$

[3] Here, $\mathbf{x} \in \mathbb{R}^d$ is a random vector whose n realizations are $X = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$.

For a fixed p, the source separation problem boils down to finding the vector $\mathbf{w}^*$ that maximizes the above log-likelihood which can be simplified as

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} |\mathbf{w}^T\mathbf{x}_i|^p \qquad s.t. \quad \mathbf{w}^T\mathbf{w} = 1, \tag{9}$$

because α and β are constants if p is fixed.[4] Note that (9) is the form of l_p-norm minimization for arbitrary $p \neq 2$, whereas the maximization of (9) reduces to the problem of Lp-PCA which has been tackled in [24].

3.2. Discussion

Considering $\mathbf{w}^T\mathbf{w} = 1$ and X is whitened, then $||\mathbf{y}||_2^2 = \mathbf{y}^T\mathbf{y} = \mathbf{w}^T XX^T\mathbf{w} = n$. Now, (9) can be transformed to an optimization problem with respect to $\mathbf{y}$ as follows:

$$\mathbf{y}^* = \arg\min_{\mathbf{y}} ||\mathbf{y}||_p \qquad s.t. \quad ||\mathbf{y}||_2 = \sqrt{n}, \quad \mathbf{y} = X^T\mathbf{w}. \tag{10}$$

If we relax the problem by removing the second constraint in (10), it becomes

$$\mathbf{y}^* = \arg\min_{\mathbf{y}} ||\mathbf{y}||_p \qquad s.t. \quad ||\mathbf{y}||_2 = \sqrt{n}. \tag{11}$$

The above problem is an l_p-norm minimization on an n-dimensional hypersphere. For every $0 < p < 2$ (super-Gaussian sources), the optimal solution $\mathbf{y}^*$ of this relaxed problem is axis aligned, i.e., $\mathbf{y}^* = \pm\sqrt{n}\mathbf{e}_i$ $(\forall i = 1, \cdots, n)$ where $\mathbf{e}_i$ is the unit vector on the ith coordinate axis. On the other hand, for every $p > 2$ (sub-Gaussian), it becomes $\mathbf{y}^* = \{\pm 1\}^n$, i.e., each element of $\mathbf{y}^*$ is either 1 or -1.[5] This is illustrated in Fig. 1, for a simple case of $n = 2$. Because every super-Gaussian sources have exactly the same minimum norm solution of (11), a fixed value of p (e.g., $p = 1$) can be used to find super-Gaussian sources. Likewise, another fixed value (e.g., $p = 3$) can also be used for sub-Gaussian sources.

The solution of (11) has a very close relationship with the kurtosis optimization. Because $\mathbf{y}$ can be interpreted as n samples of an

[4] In this formulation, we cannot guarantee the uniqueness of the optimal solution and the equality in (9) should be replaced with $\in$. However, we use equality sign for simplicity throughout the paper.

[5] This can be proved by a Lagrangian multiplier method.

Table 1
ICA by l_p-norm optimization.

	minimization	maximization
$p<2$	super-G. (see Section 4 and 5)	sub-G. (PCA-L1 [23] or Lp-PCA[24])
	e.g. $\mathbf{w}^* = \arg\min_{\mathbf{w}:\mathbf{w}^T\mathbf{w}=1} \sum_{i=1}^{n} \lvert\mathbf{w}^T\mathbf{x}_i\rvert^1$	e.g. $\mathbf{w}^* = \arg\max_{\mathbf{w}:\mathbf{w}^T\mathbf{w}=1} \sum_{i=1}^{n} \lvert\mathbf{w}^T\mathbf{x}_i\rvert^1$
$p>2$	sub-G. (see Section 4 and 5)	super-G. (Lp-PCA [24])
	e.g. $\mathbf{w}^* = \arg\min_{\mathbf{w}:\mathbf{w}^T\mathbf{w}=1} \sum_{i=1}^{n} \lvert\mathbf{w}^T\mathbf{x}_i\rvert^3$	e.g. $\mathbf{w}^* = \arg\max_{\mathbf{w}:\mathbf{w}^T\mathbf{w}=1} \sum_{i=1}^{n} \lvert\mathbf{w}^T\mathbf{x}_i\rvert^3$

estimated source $y(\simeq s)$, an ICA algorithm by optimizing the kurtosis of y can be thought of a special case of (11) when $p=4$. Formally speaking, the kurtosis optimization problem becomes

$$\mathbf{y}^* = \arg\max_w / \arg\min_w \sum_{i=1}^{n} y_i^4 \quad s.t. \quad ||\mathbf{y}||_2 = \sqrt{n}. \tag{12}$$

Here, the maximization problem is to find the super-Gaussian sources and the minimization is for the sub-Gaussian sources. Note that the solution that maximizes (12) is $\mathbf{y}^* = \pm\sqrt{n}\mathbf{e}_i$, while the one that minimizes (12) is $\mathbf{y}^* = \{\pm 1\}^n$, which are exactly the same as the solutions of (11) for $p<2$ and $p>2$, respectively.

However, note that because of the constraint $\mathbf{y} = X^T\mathbf{w}$, these optimal points ($\mathbf{y}^* = \pm\sqrt{n}\mathbf{e}_i$ or $\{\pm 1\}^n$) cannot be attained in general. Nevertheless, (11) plays a crucial role in developing a fast ICA algorithm in Section 5.

Another interesting thing to point out in this part is that the minimum norm solution of (11) for $p<2$ (see Fig. 1(a)) is the same as the maximum norm solution for $p>2$ (see Fig. 1(b)) and vice versa. Therefore, for sub-Gaussian sources, the problem can be reformulated as the l_p-norm maximization problem with $p<2$ where PCA-L1 [23] or Lp-PCA [24] algorithms can directly be used. In summary, a fixed value of p can be used for separating both super-Gaussian and sub-Gaussian sources. For example, if we fix $p=1$, ICA problem becomes

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} |\mathbf{w}^T\mathbf{x}_i| \quad \text{s.t.} \|\mathbf{w}\|_2 = 1 \tag{13}$$

for super-Gaussian sources, while it becomes

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \sum_{i=1}^{n} |\mathbf{w}^T\mathbf{x}_i| \quad \text{s.t.} \|\mathbf{w}\|_2 = 1 \tag{14}$$

for sub-Gaussian sources.

Table 1 summarises this section. To separate super-Gaussian (sub-Gaussian) sources, one has to solve an l_p-norm minimization problem with $p<2$ ($p>2$) or an l_p-norm maximization problem with $p>2$ ($p<2$). The maximization problem can be solved by PCA-L1 [23] or Lp-PCA [24]. The solutions for the minimization problems are dealt with in Sections 4 and 5.

4. Lp-ICA-G: a gradient ascent algorithm

In [24] the gradient of the objective function (9) with respect to $\mathbf{w}$ is calculated as

$$\tilde{\nabla}_{\mathbf{w}} = p\sum_{i=1}^{n} \text{sign}(\mathbf{w}^T\mathbf{x}_i)|\mathbf{w}^T\mathbf{x}_i|^{p-1}\mathbf{x}_i, \tag{15}$$

where sign$(\cdot)$ is a function that outputs $+1$ or -1 depending on the sign of the input and outputs 0 when the input is 0. Multiplying $-\beta(p)$ to this yields the gradient $\nabla_{\mathbf{w}}$ of the log likelihood (8) with respect to $\mathbf{w}$. When $p>1$, the gradient (15) is well defined even for the singular points where $\mathbf{w}^T\mathbf{x}_i = 0$ for some $\{\mathbf{x}_i\}_{i=1}^n$. On the other hand, if $0<p<1$, the gradient is not well defined for the singular points. However, in this case, we can consider the subdifferential $\partial f(\mathbf{w})$ of the function $f(\mathbf{w}) = |\mathbf{w}^T\mathbf{x}_i|^p$ and it is obvious that it contains zero. Therefore, the data points that are orthogonal to $\mathbf{w}$ can be ignored in the computation of (15).

In addition to $\nabla_{\mathbf{w}}$, the gradient of the log likelihood (8) with respect to p can be obtained as

$$\nabla_p = n\frac{\alpha'}{\alpha} - \beta'\sum_{i=1}^{n}|\mathbf{w}^T\mathbf{x}_i|^p - \beta\sum_{i=1}^{n}|\mathbf{w}^T\mathbf{x}_i|^p\log|\mathbf{w}^T\mathbf{x}_i|, \tag{16}$$

where α' and β' are the gradients of $\alpha(p)$ and $\beta(p)$ with respect to p, respectively. In practice, the method of numerical differentiation is used to obtain α' and β' since the derivative of a gamma function is not in a closed form.

In finding multiple sources, the kth unmixing vector $\mathbf{w}_k$ should be orthogonal to the previously found vectors $\mathbf{w}_1, \cdots, \mathbf{w}_{k-1}$. For this purpose, the Gram-Schmidt orthogonalization can be used. In our method, to reduce the computational complexity, instead of applying Gram-Schmidt orthogonalization to $\mathbf{w}_k$ in every iteration, the orthogonalization process is applied to the data matrix X only once in finding each source. Once d vectors are found, the final unmixing matrix can be obtained as $W = [\mathbf{w}_1, \cdots, \mathbf{w}_d]^T$. The overall procedure is described in Algorithm 1. Here, 'normal' is a vector

Algorithm 1 Lp-ICA-G: a gradient ascent algorithm for ICA by l_p-norm optimization.

Require: whitened data $X^{(0)} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$
Ensure: unmixing matrix $W = [\mathbf{w}_1, \ldots, \mathbf{w}_d]^T$
  $W \leftarrow [\ ]$
  for $k = 1$ **to** d **do**
    Initialize $\mathbf{w}_k$ as a random vector and $p \leftarrow 2$
    $\mathbf{w}_k \leftarrow \text{normal}(\mathbf{w}_k - WW^T\mathbf{w}_k)$
    $X^{(k)} \leftarrow X^{(k-1)} - WW^TX^{(k-1)}$
    repeat
      $p \leftarrow p + \eta\nabla_p$ by using (16)
      $\mathbf{w}_k \leftarrow \text{normal}(\mathbf{w}_k + \eta\nabla_{\mathbf{w}_k})$by using (15)
    until converge
    $W \leftarrow [W, \mathbf{w}_k]$
  end for
  $W \leftarrow W^T$

normalization operation, i.e., $\text{normal}(\mathbf{a}) = \frac{\mathbf{a}}{||\mathbf{a}||_2}$. The learning rate η is set as a constant in our implementation.

Note that if we know the type of the sources in advance, we can set p as a fixed value and l_p-norm minimization problem in (9) can be tackled. In this case, as shown in Table 1, instead of l_p-norm minimization, an l_p-norm maximization problem can be solved to find the independent components. For l_p-norm maximization, PCA-L1 [23] or Lp-PCA [24], which are basically gradient ascent methods, can be used. More in-depth relationship between ICA and PCA can be found in [25].

5. Lp-ICA-F: a non-iterative fast algorithm

5.1. Separating super-Gaussian sources ($p<2$)

The Lp-ICA-G is a simple and intuitive implementation for ICA. However, the convergence rate of this method is much slower than that of fastICA [12] and there are many chances that the solution is locally optimal, which is an inherent property of gradient descent methods.

From now on, we will concentrate on finding the solution of (10). The proposed method in this part is closely related to the strategy in [26,30].

Let us consider the super-Gaussian ($p<2$) case first. As we have seen already, (11) has n solutions $\mathbf{y}^* = \sqrt{n}\mathbf{e}_i$, $i = 1, \cdots, n$. However, in our formulation, $\mathbf{y}$ should be equal to $X^T\mathbf{w}$, where the rank of X

is at most d. Therefore, $\mathbf{y}$ spans a subspace in $\mathbb{R}^n$. In other words, the solution for $X^T\mathbf{w} = \sqrt{n}\mathbf{e}_i$ may not exist. In this case, least square solutions of $\mathbf{w}^{(i)}$ are firstly found that make $\mathbf{y}^{(i)} = X^T\mathbf{w}^{(i)}$ be the closest point to $\sqrt{n}\mathbf{e}_i$ for $i = 1, \cdots, n$ as follows:

$$\begin{aligned} \mathbf{w}^{(i)} &= \arg\min_{\mathbf{w}} ||X^T\mathbf{w} - \sqrt{n}\mathbf{e}_i||_2 \\ s.t. \quad & \mathbf{w}^T\mathbf{w} = 1, \quad i = 1, \cdots, n. \end{aligned} \tag{17}$$

Then, the final solution $\mathbf{w}^*$ is set to one of $\mathbf{w}^{(i)}$'s that produces the minimum objective function value as follows:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}^{(i)}} \{||X^T\mathbf{w}^{(1)}||_p, \cdots, ||X^T\mathbf{w}^{(n)}||_p\}. \tag{18}$$

The above procedure (17) and (18) can be used to obtain the first unmixing vector $\mathbf{w}_1$. When finding multiple unmixing vectors, the kth vector $\mathbf{w}_k$ should be orthogonal to the previously found vectors $W_{k-1} = [\mathbf{w}_1, \cdots, \mathbf{w}_{k-1}]$. In the case of $k > 1$, (17) is replaced with

$$\begin{aligned} \mathbf{w}^{(i)} &= \arg\min_{\mathbf{w}} ||X^T\mathbf{w} - \sqrt{n}\mathbf{e}_i||_2 \\ s.t. \quad \mathbf{w}^T\mathbf{w} &= 1, \; W_{k-1}^T\mathbf{w} = 0, \quad i = 1, \cdots, n, \end{aligned} \tag{19}$$

After obtaining n solutions of (19), (18) again is applied to obtain the kth unmixing vector.

Both (17) and (19) can be solved using Lagrangian multipliers. By introducing Lagrangian multipliers λ and $\boldsymbol{\mu} = [\mu_1, \mu_2, \cdots, \mu_{k-1}]$, (19) becomes

$$\mathcal{L}(\mathbf{w}, \lambda, \mu) = \frac{1}{2}\left\|X^T\mathbf{w} - \sqrt{n}\mathbf{e}_i\right\|_2 + \frac{\lambda}{2}(\mathbf{w}^T\mathbf{w} - 1) + \sum_{j=1}^{k-1}\mu_j\mathbf{w}^T\mathbf{w}_j. \tag{20}$$

Setting the derivative with respect to $\mathbf{w}$ to zero as

$$\frac{\partial\mathcal{L}}{\partial\mathbf{w}} = X(X^T\mathbf{w} - \sqrt{n}\mathbf{e}_i) + \lambda\mathbf{w} + \sum_{j=1}^{k-1}\mu_j\mathbf{w}_j = 0, \tag{21}$$

and rearranging terms, it becomes

$$(n + \lambda)\mathbf{w} = \sqrt{n}\mathbf{x}_i - \sum_{j=1}^{k-1}\mu_j\mathbf{w}_j, \tag{22}$$

where the identities $XX^T = nI_d$ and $X\mathbf{e}_i = \mathbf{x}_i$ are used. Left multiplying $\mathbf{w}_j^T$ on both sides yields

$$\mu_j = \sqrt{n}\mathbf{w}_j^T\mathbf{x}_i. \tag{23}$$

Substituting (23) to (22) gives

$$(n + \lambda)\mathbf{w} = \mathbf{x}_i - WW^T\mathbf{x}_i. \tag{24}$$

If we let $\mathbf{x}_i^{(k)} \triangleq (I_d - WW^T)\mathbf{x}_i$, then $\mathbf{w}$ can be calculated as

$$\mathbf{w} = \text{normal}(\mathbf{x}_i^{(k)}) \tag{25}$$

The solution of (17) can be obtained in a similar manner, which is $\mathbf{w} = \text{normal}(\mathbf{x}_i)$. Since we do not know which $\mathbf{y}$ that is closest to $\mathbf{e}_i (i = 1, \cdots, n)$ minimizes the cost function, total n trials are needed. The solution guarantees global minimum for the cost function (17) and (19). Although it does not guarantee the optimality of (9), it is expected to be very close to the global optimal solution because the approximation (17) is close to (9). Furthermore, because it checks all n possible vertices, the proposed algorithm does not fall on the local optimal solution. The proposed algorithm Lp-ICA-F+ is described in Algorithm 2. The first line of the outer loop is the orthogonalization step to make sure that $\mathbf{w}^{(\hat{i})}$ is orthogonal to the previously found set of unmixing vectors.

The computational complexity of Lp-ICA-F is $\mathcal{O}(dn^2)$. Typically, the number of sources is much smaller than the number of samples, i.e. $d \ll n$ and the complexity is quadratic in n. However, The proposed algorithm is not an iterative one which is different from most of other ICA algorithms. Hence, the proposed algorithm can be fast for small datasets.

Algorithm 2 Lp-ICA-F+(super): a fast ICA algorithm for super-Gaussian sources.

Require: whitened data $X^{(0)} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$, $p < 2$
Ensure: unmixing matrix $W = [\mathbf{w}_1, \cdots, \mathbf{w}_d]^T$
  $W \leftarrow [\,]$
  for $k = 1$ **to** d **do**
    $X^{(k)} \leftarrow X^{(k-1)} - WW^TX^{(k-1)}$
    for $i = 1$ **to** n **do**
      $\mathbf{w}^{(i)} \leftarrow \text{normal}(\mathbf{x}_i^{(k)})$, $\quad c(i) \leftarrow ||X^T\mathbf{w}^{(i)}||_p$
    end for
    $\hat{i} \leftarrow \arg\min_i c(i)$, $\quad W \leftarrow [W, \mathbf{w}^{(\hat{i})}]^T$
  end for
  $W \leftarrow W^T$

5.2. Separating sub-Gaussian sources (p > 2)

If the same algorithm as in Section 5.1 is to be applied to sub-Gaussian case, $\sqrt{n}\mathbf{e}_i$ in (17) and (19) must be replaced with $\{\pm 1\}^n$, and the one that minimizes the objective function $||X^T\mathbf{w}||_p$ can be chosen as a solution. However, in this case, the time complexity will be exponential since we have to check 2^n combinations of ± 1 for a possible solution. To deal with the inherent difficulty of combinatorial optimization, a greedy approach that seeks local minimum is applied for fast sub-Gaussian source separation. There are numerous algorithms which are based on greedy approaches, and sufficient evidences are provided that greedy approach practically finds an acceptable solution even if it is not the best [31].

In Algorithm 3, the proposed greedy algorithm is shown. In our greedy algorithm, firstly a random vector $\mathbf{r} \in \{\pm 1\}^n$ whose elements are either 1 or -1 are generated to form an initial sum $\mathbf{t} = X\mathbf{r}$. Then $\mathbf{t}$ is normalized to unit vector $\mathbf{w}$, and l_p-norm of the projection $X^T\mathbf{w}$ is calculated and stored as v. After this, a sample $\mathbf{x}_{c_i}$ is randomly selected to form a new sum $\mathbf{t}'$ whose polarity of $\mathbf{x}_{c_i}$ is negated from the original sum vector $\mathbf{t}$. The normalized vector $\mathbf{w}'$ of $\mathbf{t}'$ is checked whether to decrease the l_p-norm of the projection $X^T\mathbf{w}'$. If so, this is a more appropriate candidate for the optimal unmixing vector and $\mathbf{w}$ and $\mathbf{t}$ are replaced with $\mathbf{w}'$ and $\mathbf{t}'$, respectively. This procedure continues for all the samples. The time complexity of the proposed algorithm is approximately three times that of the super-Gaussian case.

Algorithm 3 Lp-ICA-F-(sub): a fast ICA algorithm for sub-Gaussian sources.

Require: whitened data $X = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$, $p > 2$
Ensure: unmixing matrix $W = [\mathbf{w}_1, \cdots, \mathbf{w}_d]^T$
  $W \leftarrow [\,]$
  for $k = 1$ **to** d **do**
    $X^{(k)} \leftarrow X^{(k-1)} - WW^TX^{(k-1)}$
    $\mathbf{r} \leftarrow \text{rand}(\{\pm 1\}^n)$, $\quad \mathbf{c} \leftarrow \text{randperm}([1, \cdots, n]^T)$.
    $\mathbf{t} \leftarrow X^{(k)}\mathbf{r}$, $\quad \mathbf{w} \leftarrow \text{normal}(\mathbf{t})$, $\quad v \leftarrow ||X^T\mathbf{w}||_p$
    for $i = 2$ **to** n **do**
      $\mathbf{t}' \leftarrow \mathbf{t} - 2r_{c_i}\mathbf{x}_{c_i}$, $\quad \mathbf{w}' \leftarrow \text{normal}(\mathbf{t}')$
      if $||X^T\mathbf{w}'||_p < v$ **then**
        $\mathbf{t} \leftarrow \mathbf{t}'$, $\quad \mathbf{w} \leftarrow \mathbf{w}'$
      end if
    end for
    $W \leftarrow [W, \mathbf{w}]$
  end for
  $W \leftarrow W^T$

To measure the optimality of the proposed greedy algorithm, we compared the optimality of the greedy solution with all possible combinations of $\{\pm 1\}^n$ for a small number of samples n as follows: For the mixed sub-Gaussian sources of uniform distribu-

tion, we find $\mathbf{w}_1$ according to Algorithm 3 with $p = 3$. Also, $\mathbf{w}$ that is closest to each point of $\{\pm 1\}^n$ is calculated for every possible points. Then, the rank of Lp-ICA-F- is determined as the number of combinations that has smaller value of $||Xw||_3$ than the greedy solution. The result of the experiments for the cases $n = 10, 15, 20,$ and 25 is illustrated in Fig. 2. The rank ratio is calculated as the rank divided by 2^n and the mean and the median rank ratio of 100 trials for each case are reported in the figure. It is shown that the rank ratio gets smaller as the number of samples increases. Therefore, it can be expected that the proposed algorithm Lp-ICA-F- will find the solution that is very close to the optimal one for a large data set. Since the ICA problem quite often deals with the case of $d \ll n$, the greedy solution will give a promising result. Experimental result on the next section enforces this statement.

The algorithms Lp-ICA-F+ and Lp-ICA-F- can be combined to separate arbitrary sources. In this case, the k-th unmixing vector $\mathbf{w}_k$ can be found by comparing the log-likelihood (8) of solutions by Lp-ICA-F+ and Lp-ICA-F- and setting it to the one that outputs maximum log-likelihood. This approach which is denoted as Lp-ICA-F is used in all the experiments in the next section. The computational complexity of Lp-ICA-F is approximately four times that of the basic Lp-ICA-F+ shown in Algorithm 2.

6. Simulation

In this section, the performance of proposed algorithms, Lp-ICA-G in Section 4 and Lp-ICA-F in Section 5, are compared with FastICA [12] and RobustICA [18]. We considered two types of widely used non-Gaussian distribution for the simulation. Laplacian distribution is considered as a representative of the super-Gaussian sources. Since speech signal approximately follows Laplacian distribution, ICA for Laplacian distribution is often used for blind source separation. On the other hand, uniform distribution represents the sub-Gaussian sources, where ICA can be applied to noise removal. In the following experiments, Laplacian and uniform distribution are generated with zero mean and unit variance. After the generated sources are mixed through a random mixing matrix, the mixed signals are whitened by the whitening process described in Section 2. Then, Lp-ICA-G, Lp-ICA-F, FastICA and RobustICA algorithms are performed on the whitened data. The separation performance is measured in terms of a signal to noise ratio (SNR) which is computed by

$$\mathrm{SNR(dB)} = \frac{1}{d}\sum_{k=1}^{d} 10\log\left(\frac{s_k^2}{(s_k - y_k)^2}\right) \tag{26}$$

where s_k and y_k are the kth original source and the kth estimated source, respectively. All the experiments are conducted 100 times for each case and the average values of SNR are reported in this paper. For FastICA and RobustICA algorithms, the default settings of the original authors' MATLAB implementation were used. In FastICA, the contrast function 'pow3 ($g(u) = u^3$)' was used ,which is the default setting of the MATLAB implementation. We chose the contrast function since it can be used to separate both sub-Gaussian and super-Gaussian sources. All the experiments were performed using MATLAB on Intel i7 core at 2.20 GHz.

6.1. Noiseless case

First, to measure the performance of various algorithms for the basic case, the size of the data is fixed to $n = 500$ and $d = 8$. We varied the number of sub- and super-Gaussian sources from 0 to 8 and reported the average SNRs in Table 2. In all the tables afterwords, the numbers in the parentheses are the standard deviations. For sub-Gaussian only case (8:0), the PCA-L1 algorithm which uses Lagrangian method of l_1-norm maximization [23] was also tested and the resulted SNR was 9.71 dB with standard deviation 3.66 dB, which is far below the performances of other algorithms. For the pure sub-Gaussian sources (8:0), RobustICA performed best but other algorithms also performed well. In this case, since the greedy algorithm Lp-ICA-F- does not check all the 2^n possible solutions, it is slightly inferior to the other algorithms. For super-Gaussian only case (0:8), we measured the performance of Lp-PCA [24] with $p = 3$. The mean SNR was 14.21dB with standard deviation of 1.47, which outperforms Lp-ICA and all the other algorithms. However, Lp-PCA algorithm is not robust to noise, which will be discussed in Section 6.2. For other cases, the proposed algorithms, especially Lp-ICA-G, performed better than FastICA and RobustICA. However, Lp-ICA-G runs much slower than other algorithms since its convergence rate is linear. Running time of Lp-ICA-F is three times to that of FastICA while Lp-ICA-F gives better reconstruction result than FastICA.

Table 2
Average SNR for $n = 500$ and $d = 8$ (in dB). The left-most column indicates the number of sub- and super-Gaussian sources. The last row is the average time in ms for all the experiments.

Sub:Super	Lp-ICA-G	Lp-ICA-F	FastICA	RobustICA
0:8	**12.12** (3.27)	9.87 (1.51)	8.51 (2.02)	8.76 (2.06)
2:6	**14.55** (1.95)	13.65 (2.12)	10.97 (1.76)	10.56 (1.62)
4:4	16.76 (1.92)	**17.06** (1.36)	13.46 (1.87)	12.49 (1.56)
6:2	**18.67** (2.48)	18.50 (1.18)	16.75 (1.57)	15.74 (1.53)
8:0	19.63 (4.08)	19.54 (1.07)	20.57 (1.25)	**20.83** (1.10)
Time (ms)	1850	19.5	**6.5**	13.5

Table 3
The t-values of one-tailed t-test for various hypotheses. Positive t-values indicate Lp-ICA is superior to the compared method while negative t-values indicate the inferiority of Lp-ICA.

Sub:Super	Lp-ICA-G = FastICA	Lp-ICA-F = FastICA	Lp-ICA-G = RobustICA	Lp-ICA-F = RobustICA
0:8	9.392	5.393	8.694	4.346
2:6	13.63	9.727	15.74	11.58
4:4	12.31	15.57	17.26	22.08
6:2	6.541	8.910	10.05	14.28
8:0	−2.203	−6.260	−2.840	−8.406

We conducted t-test to the result in Table 2 for statistical significance testing. One-tailed t-test is used to statistically verify whether Lp-ICA-G and Lp-ICA-F are superior to FastICA and RobustICA or not. The t-values for various hypotheses are provided in Table 3. We set up the hypotheses to examine the superiority of Lp-ICA algorithms over FastICA and RobustICA is statistically meaningful. The thresholds for one-tailed t-test are around 1.65 for 95% confidence interval and around 2.34 for 99% confidence interval respectively. Most t-values presented in Table 3 exceed the threshold values with large margins. Hence, the result statistically justifies Lp-ICA-G and Lp-ICA-F are superior to FastICA and RobustICA in the case that only super-Gaussian sources exist and the case that super- and sub-Gaussian sources are mixed. On the other hand, when only sub-Gaussian sources exist, FastICA and RobustICA performs better than Lp-ICA algorithms.

Next, the algorithms were tested with various numbers of samples. The number of samples are varied as 100, 300, 500, 700, and 900 with a fixed number of sources (3 sub-Gaussian, 3 super-Gaussian). The results are shown in Table 4. The SNR increases as the sample size gets bigger. Lp-ICA-F performs best for all the cases. Moreover, the gap between Lp-ICA-F and the other algorithms tends to increase as the number of samples gets larger. In addition to the poor performance in the mixed source case, FastICA has another drawback that it sometimes fails to converge when the sample size is small. In our experiment, 7 out of 100 trials failed to converge for $n = 100$ and these cases were excluded in the com-

Table 4
Average SNRs and Time with varying number of samples.

# Samples	Lp-ICA-G	Lp-ICA-F	FastICA	RobustICA
(a) SNR (in dB)				
100	8.25 (3.27)	**9.37** (2.80)	7.53 (2.91)	7.48 (2.46)
300	16.09 (2.82)	**16.20** (2.25)	13.46 (2.30)	12.73 (1.91)
500	18.72 (2.23)	**19.00** (1.46)	15.82 (2.15)	14.82 (1.87)
700	20.63 (1.89)	**20.94** (1.72)	16.95 (2.58)	15.74 (1.90)
900	21.62 (2.37)	**22.09** (1.60)	18.08 (2.23)	17.13 (2.11)
(b) Time (in ms)				
100	247 (11)	**2.08** (0.97)	47.70 (85.21)	7.81 (9.20)
300	606 (11)	6.94 (0.20)	**3.93** (10.65)	7.44 (2.23)
500	927 (14)	13.46 (0.55)	**2.98** (1.08)	7.67 (2.91)
700	775 (14)	21.52 (1.65)	**3.38** (1.35)	7.26 (1.43)
900	948 (22)	31.53 (2.89)	**3.72** (0.92)	7.70 (1.26)

putation of mean and standard deviation. In all the cases, FastICA performed better than RobustICA.

Looking at the computation time in Table 4(b), Lp-ICA-G performed very slow compared to other algorithms because it is a simple gradient based method. For small number of samples (100), Lp-ICA-F was fastest but as the number of samples increased it slowed down. To find the empirical power law relationship between the number of samples n and the time, we used the five points and could get $time(ms) \propto n^{1.23}$. Note that the exponent 1.23 is much smaller than 2, which is from the theoretical analysis. The speed of FastICA and RobustICA were almost constant regardless of the number of samples. The relatively large number for FastICA when $n = 100$ is due to the 7 failed trials.

Table 5 reports the performances with different numbers of sources. In the experiments, the number of samples were fixed to 500. We increased the number of sources from 2 to 18 where the number of sub- and super-Gaussian sources were set to be equal. The effect of increasing the number of sources is similar to decreasing the number of samples. Lp-ICA-F gave the best result and one trial of FastICA failed to converge for $d = 18$. The results are shown in Table 5(a).

To check the performances of the proposed methods for the sources that does not follow the generalized Gaussian distribution, we generated two types of non-symmetric distribution as follows: (1) (Lap+Gau) Half of the samples were generated by the Laplacian and the Gaussian distributions respectively and the samples from the Laplacian distribution were enforced to have negative values and those from the Gaussian to have positive values by applying the absolute value operation on the samples. (2) (Uni+Gau) In the procedure of generating the first type of sources (Lap+Gau), the Laplacian distribution was replaced by the uniform distribution. In Table 5(b), we reported the performances with various number of sources. In all the experiments, the numbers of (Uni+Gau) and (Lap+Gau) sources were set to be equal. In the table, we can see that the performance of ICA algorithms degrade drastically as the number of non-symmetrically distributed sources increases. Except the case of $d = 8$, Lp-ICA-F performed best.

6.2. Noisy case

In this experiment, Gaussian noise and uniform noise are added to the original source. The noise is modelled as a zero-mean random variable that is independent from the source. We varied the variance of the Gaussian noise and the support of uniform noise. For each experiment, it was set $n = 500$ and $d = 6$. Equal numbers of sub- and super-Gaussian sources were used. Performances with additive Gaussian noise and uniform noise are illustrated in Table 6. The results of Lp-PCA [24] are also included, where $p = 1$ and $p = 3$ are used to separate sub-Gaussian and super-Gaussian respectively. The SNR decreases as the variance of noise increases.

Table 5
Average SNR with varying number of sources (in dB). (a) Laplacian & the Uniform (b) Lap+Gau & Uni+Gau distributions.

# Sources	Lp-ICA-G	Lp-ICA-F	FastICA	RobustICA
(a) Same number of sources from the Laplacian & uniform distributions				
2	32.99 (7.88)	**33.11** (6.68)	27.50 (8.78)	24.81 (7.46)
6	18.77 (2.07)	**19.43** (1.77)	15.82 (2.43)	14.83 (2.11)
10	14.75 (2.00)	**15.32** (1.23)	11.89 (1.46)	11.17 (1.35)
14	11.61 (2.37)	**12.17** (1.71)	9.44 (1.32)	8.50 (1.31)
18	**8.15** (2.34)	7.84 (2.45)	6.56 (1.61)	5.73 (1.42)
(b) Same number of sources from the Lap+Gau & Uni+Gau distributions				
2	26.91 (8.30)	**29.11** (7.50)	25.90 (9.50)	23.31 (8.05)
4	8.23 (1.92)	**8.25** (0.78)	7.96 (0.80)	7.83 (0.75)
6	4.77 (1.40)	**5.02** (0.36)	4.67 (1.15)	4.81 (0.59)
8	3.27 (1.02)	3.21 (1.08)	3.09 (1.07)	**3.35** (0.58)

Table 6
Average SNRs with additive noise (in dB).

(a) Additive Gaussian noise					
Std.	Lp-ICA-G	Lp-ICA-F	FastICA	RobustICA	Lp-PCA
10^{-3}	18.78 (2.09)	**18.93** (1.66)	15.87 (2.25)	14.39 (1.80)	14.53 (2.96)
10^{-2}	16.06 (3.45)	**16.12** (3.30)	13.50 (2.81)	12.69 (2.49)	13.34 (3.72)
10^{-1}	7.00 (2.35)	**7.04** (2.31)	6.92 (1.86)	6.73 (1.79)	6.11 (2.32)
(b) Additive uniform noise					
Interval	Lp-ICA-G	Lp-ICA-F	FastICA	RobustICA	Lp-PCA
$\pm 5 \times 10^{-4}$	18.87 (2.16)	**19.15** (1.59)	16.02 (2.54)	14.59 (1.93)	15.17 (2.84)
$\pm 5 \times 10^{-3}$	17.89 (2.73)	**18.23** (2.66)	14.93 (2.75)	13.87 (2.31)	14.76 (2.83)
$\pm 5 \times 10^{-2}$	12.60 (3.45)	**12.88** (3.47)	11.36 (2.73)	10.84 (2.52)	10.84 (3.27)

Table 7
Average SNRs of Gaussian-mixed sources for different ratios of Gaussian samples (in dB).

Ga. Ratio	Lp-ICA-G	Lp-ICA-F	FastICA	RobustICA
0.1	16.76 (3.03)	**17.96** (1.69)	14.60 (2.06)	13.66 (2.06)
0.2	16.13 (2.10)	**16.47** (2.14)	13.71 (1.95)	13.23 (1.99)
0.3	13.78 (2.84)	**14.79** (2.38)	12.35 (2.02)	12.16 (2.12)
0.4	11.92 (3.63)	**13.24** (2.55)	11.42 (2.26)	11.14 (2.32)
0.5	9.72 (3.10)	**10.58** (2.72)	9.72 (2.57)	9.54 (2.23)

Relative performances between the methods are analogous to the noiseless case. In this experiment, it can be verified that both Lp-ICA-G and Lp-ICA-F work well in the presence of noise but Lp-ICA-F is slightly better than Lp-ICA-G. Lp-PCA performs worse than Lp-ICA in all the cases.

Lastly, to make the source distribution more challenging, the source distribution is changed from Laplacian (uniform) to the mixture of Gaussian and Laplacian (uniform) for super-Gaussian (sub-Gaussian) source. In other words, each super-Gaussian (sub-Gaussian) source contains the samples from Laplace (uniform) distribution together with the samples from the zero-mean, unit-variance Gaussian distribution. The distribution of samples are still sub- or super-Gaussian depending on the original unmixed distribution, but the distribution gets closer to the Gaussian distribution as the ratio of the Gaussian samples increases, i.e., the absolute value of the normalized kurtosis of the distribution gets smaller. SNRs are measured for different ratios of Gaussian samples from 0.1 to 0.5. The results are shown in Table 7. Lp-ICA-F showed the best performance. FastICA failed to converge for 5 cases when the ratio of Gaussian samples is 0.5.

7. Experiments on real data

To show the applicability of the proposed method on real world data, we applied Lp-ICA in electroencephalographic (EEG) data. We used the dataset from [8] which contains 71-channel EEG signals of

13 subjects.[6] Two evaluation schemes are proposed to measure the performance of ICA algorithms in [8]. First, mutual information reduction (MIR) between the recorded data channels and the recovered components measures independence in terms of mutual information. Second, since independent EEG sources are dipolar [8], the amount of the decomposed signals that are near-dipolar can be used to assess the quality of decomposed signals. Each decomposed signal is fitted into a single equivalent dipole using the DIPFIT plug-in of the EEGLAB toolbox [32]. Then, residual variance is calculated, which is the error of dipole fitting. Near-dipolar components are defined as the signals whose residual variance is less than 5%. We refer the readers to [7,8] for the details of the evaluation methods. Good ICA decomposition has high MIR value and large amount of near-dipole decomposed sources. We used the first 10,000 samples of the original signals for the experiment.

We measured MIR (kbits/sec) and the portion of near-dipole sources (%) for the results of various ICA algorithms and PCA, and we plot the results on the 2D plane (Fig. 3). Both Lp-ICA-F and Lp-ICA-G generates more near-dipolar sources than FastICA and RobustICA. Meanwhile, MIR values of Lp-ICA algorithms are smaller than FastICA and are similar to RobustICA. Therefore, Lp-ICA outperforms RobustICA and is competitive with FastICA for EEG data decomposition. This result demonstrates the applicability of Lp-ICA algorithms on real world data.

8. Conclusion

In this paper, ICA is analysed in different perspective and a couple of new algorithms of ICA based on l_p-norm minimization is proposed. The proposed methods assume that the sources are distributed according to a generalized Gaussian distribution, i.e., $p(s) = \alpha \exp(\beta |s|^p)$. By applying the maximum likelihood estimation, ICA reduces to an l_p-norm minimization problem, which has a very close relationship with the PCA based on l_p-norm maximization [24]. The relationship between the proposed l_p-norm minimization and kurtosis optimization is also discussed. In addition, we also show that an ICA algorithm can be formulated as an l_1-norm minimization or maximization problem depending on the type of sources, which is related to the work in [25].

In addition to the gradient-based method, fast non-iterative algorithms for the super- and the sub-Gaussian sources were also provided. The simulation results show that the performance of the proposed Lp-ICA-F algorithm is compatible or superior to the conventional algorithms. Especially, our method gives superior performance for super-Gaussian sources and in the presence of noise.

As a future work, we can think of non-iterative version of Lp-ICA-F or solving (10) directly without relaxation. A rigorous proof of the optimality can be another research issue.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF-2016R1A1A1A05005442).

References

[1] I. Jolliffe, Principal Component Analysis, Wiley Online Library, 2005.
[2] P. Comon, Independent component analysis, a new concept? Signal Process. 36 (3) (1994) 287–314.
[3] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, 46, John Wiley & Sons, 2004.
[4] X.L. Li, T. Adali, Independent component analysis by entropy bound minimization, IEEE Trans. Signal Process. 58 (10) (2010) 5151–5164, doi:10.1109/TSP.2010.2055859.
[5] P. Comon, C. Jutten, Handbook of Blind Source Separation: Independent Component Analysis and Applications, Academic press, 2010.
[6] N.C. Rogasch, R.H. Thomson, F. Farzan, B.M. Fitzgibbon, N.W. Bailey, J.C. Hernandez-Pavon, Z.J. Daskalakis, P.B. Fitzgerald, Removing artefacts from TMS-EEg recordings using independent component analysis: importance for assessing prefrontal and motor cortex network properties, Neuroimage 101 (2014) 425–439.
[7] S.-H. Hsu, T. Mullen, T.-P. Jung, G. Cauwenberghs, Validating online recursive independent component analysis on EEg data, in: Neural Engineering (NER), 2015 7th International IEEE/EMBS Conference on, IEEE, 2015, pp. 918–921.
[8] A. Delorme, J. Palmer, J. Onton, R. Oostenveld, S. Makeig, Independent eeg sources are dipolar, PLoS ONE 7 (2) (2012) e30135.
[9] R. Jenssen, T. Eltoft, Independent component analysis for texture segmentation, Pattern Recognit. 36 (10) (2003) 2301–2315. doi: https://doi.org/10.1016/S0031-3203(03)00131-6.
[10] P.C. Yuen, J. Lai, Face representation using independent component analysis, Pattern Recognit. 35 (6) (2002) 1247–1257. doi: https://doi.org/10.1016/S0031-3203(01)00101-7.
[11] I. Dagher, R. Nachar, Face recognition using IPCA-ICAalgorithm, IEEE Trans. Pattern Anal. Mach. Intell. 28 (6) (2006) 996–1000, doi:10.1109/TPAMI.2006.118.
[12] A. Hyvärinen, Fast and robust fixed-point algorithms for independent component analysis, Neural Netw. IEEE Trans. 10 (3) (1999) 626–634.
[13] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, Neural Netw. 13 (4) (2000) 411–430.
[14] T.-W. Lee, M. Girolami, T.J. Sejnowski, Independent component analysis using an extended infomax algorithm for mixed subGaussian and superGaussian sources, Neural Comput. 11 (2) (1999) 417–441.
[15] Z. Shi, H. Tang, Y. Tang, A new fixed-point algorithm for independent component analysis, Neurocomputing 56 (0) (2004) 467–473.
[16] F.R. Bach, M.I. Jordan, Kernel independent component analysis, J. Mach. Learn. Res. 3 (2003) 1–48.
[17] Z. Koldovsky, P. Tichavsky, E. Oja, Efficient variant of algorithm FastiCA for independent component analysis attaining the Cramer-Rao lower bound, Neural Netw., IEEE Trans. 17 (5) (2006) 1265–1277.
[18] V. Zarzoso, P. Comon, Robust independent component analysis by iterative maximization of the kurtosis contrast with algebraic optimal step size, IEEE Trans. Neural Netw. 21 (2) (2010) 248–261.
[19] J. Cheng, Q. Liu, H. Lu, Y.-W. Chen, Ensemble learning for independent component analysis, Pattern Recognit. 39 (1) (2006) 81–88. doi: https://doi.org/10.1016/j.patcog.2005.06.018.
[20] A. Sharma, K.K. Paliwal, Subspace independent component analysis using vector kurtosis, Pattern Recognit. 39 (11) (2006) 2227–2232. doi: https://doi.org/10.1016/j.patcog.2006.04.021.
[21] Z. Szab, B. Pczos, A. Lrincz, Separation theorem for independent subspace analysis and its consequences, Pattern Recognit. 45 (4) (2012) 1782–1791. doi: https://doi.org/10.1016/j.patcog.2011.09.007.
[22] C. Ding, D. Zhou, X. He, H. Zha, R 1-PCA: rotational invariant l1-norm principal component analysis for robust subspace factorization, in: Proceedings of the 23rd International Conference on Machine learning, ACM, 2006, pp. 281–288.
[23] N. Kwak, Principal component analysis based on l1-norm maximization, Pattern Anal. Mach. Intell. IEEE Trans. 30 (9) (2008) 1672–1680.
[24] N. Kwak, Principal component analysis by lp-norm maximization, Cybern. IEEE Trans. 44 (5) (2014) 594–609.
[25] R. Martin-Clemente, V. Zarzoso, On the link between l1-PCA and ICa, IEEE Trans. Pattern Anal. Mach.Intell.https://doi.org/10.1109/TPAMI.2016.2557797.
[26] F.J. Theis, A. Jung, C.G. Puntonet, E.W. Lang, Linear geometric ICA: fundamentals and algorithms, Neural Comput. 15 (2) (2003) 419–439.
[27] J.-F. Cardoso, High-order contrasts for independent component analysis, Neural Comput. 11 (1) (1999) 157–192.
[28] J.V. Stone, Independent Component Analysis: A Tutorial Introduction, The MIT Press Cambridge, Massachusetts, London, England, 2004.
[29] R.J. Samworth, M. Yuan, Independent component analysis via nonparametric maximum likelihood estimation, Ann. Stat. 40 (6) (2012).
[30] M. Aharon, M. Elad, A. Bruckstein, K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation, IEEE Trans. Signal Process. 54 (11) (2006) 4311–4322.
[31] U. Faigle, Greedy algorithms in combinatorial optimization, in: Operations Research Proceedings 1993, in: Operations Research Proceedings, 1993, Springer Berlin Heidelberg, 1994, doi:10.1007/978-3-642-78910-6_161.
[32] A. Delorme, T. Mullen, C. Kothe, Z.A. Acar, N. Bigdely-Shamlo, A. Vankov, S. Makeig, EEGLAB, SIFT, NFT, BCILAB, and ERICA: new tools for advanced eeg processing, Comput. Intell. Neurosci. 2011 (2011) 10.

[6] https://sccn.ucsd.edu/wiki/BSSComparison.

Sungheon Park received his B.S. and M.S. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea in 2012 and 2014 respectively. He is currently pursuing Ph.D. degree in intelligent systems at the Machine Intelligence and Pattern Analysis Laboratory (MIPAL), Seoul National University, Seoul, Korea. His research interests include computer vision and deep learning.

Nojun Kwak was born in Seoul, Korea in 1974. He received the BS, MS, and PhD degrees from the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea, in 1997, 1999 and 2003 respectively. From 2003 to 2006, he was with Samsung Electronics. In 2006, he joined Seoul National University as a BK21 Assistant Professor. From 2007 to 2013, he was a Faculty Member of the Department of Electrical and Computer Engineering, Ajou University, Suwon, Korea. Since 2013, he has been with the Graduate School of Convergence Science and Technology, Seoul National University, Seoul, Korea, where he is currently an Associate Professor. His current research interests are mainly on feature learning by deep neural networks and their applications in various areas of pattern recognition, computer vision, image processing, and so on.