# Easy Data for Independent Component Analysis

**Ruitong Huang**
Department of Computing Science
University of Alberta
Edmonton, AB T6G2E8 Canada
ruitong@ualberta.ca

**András György**
Department of Electrical and Electronic Engineering
Imperial College London
South Kensington Campus, London SW7 2BT, UK
a.gyorgy@imperial.ac.uk

**Csaba Szepesvári**
Department of Computing Science
University of Alberta
Edmonton, AB T6G2E8 Canada
szepesva@ualberta.ca

## Abstract

Independent component analysis (ICA) is concerned with the reconstruction of sources from their mixture, assuming the sources are independent. Any collection of deterministic numbers are independent of each other. In fact, oftentimes the power of ICA algorithms is illustrated by separating mixtures of periodic, deterministic signals. These signals are "easy" for ICA, even though they show strong temporal dependencies. What then makes some data easy for ICA algorithms? This paper is an attempt to provide answers to these intriguing questions by promoting to replace probabilistic assumptions and analysis with carefully constructed data dependent bounds and a deterministic analysis. We also describe the first algorithm free of unspecific parameters whose runtime is polynomial and whose reconstruction error scales only polynomially with the natural parameters of the problem.

## 1 Introduction

Independent Component Analysis (ICA) is a data analysis technique that attempts to explain an observed $x \in \mathbb{R}^{d\times T}$ array by decomposing it into the product $As$ where $A \in \mathbb{R}^{d\times d}$ is a non-singular matrix and $s \in \mathbb{R}^{d\times T}$ is viewed as $T$ $d$-dimensional vectors such that the components of the vectors are "statistically independent" [Hyvärinen et al., 2001]. Oftentimes, this is illustrated by data as shown in Fig. 1. On the left-hand side of this figure, the bottom three plots depict the $d = 3$ components of the observed signal $x$. The $x$ axis represents time: The numbers shown are scaled by a factor of 50; thus $T = 2500$. This observed data $x$ is generated by mixing the sources shown by the top three graphs on the left-hand side. Reconstruction results by three algorithms are shown on the right-hand side. As can be seen, up to scaling and the ordering of the reconstructed components, the reconstruction is quite successful no matter the algorithms. The curious thing about this example is that the all components of the source data are periodic functions of time. This is quite obvious for the first two components, while the third, being generated using a pseudo-random number generator has a long period and thus "looks random". Are the components $s_1(t)$ and $s_2(t)$ (or $s_1(t)$ and $s_3(t)$) independent of each other as required by the standard ICA modeling assumptions? Of course they are! As the reader may recall, any two numbers $a, b \in \mathbb{R}$ when viewed as random variables, i.e., constant functions from an underlying probability space, are independent of each other, hence $s_1(t)$ and $s_2(t)$ will be independent, as will be $s_1(t)$ and $s_3(t)$, or even $s_1(t)$ and $(s_2(t), s_3(t))$. Does the success of the algorithms on this example imply that they will also work when used on the mixture of *any* sources? Of course not. For example, if one source is a linear function of another source

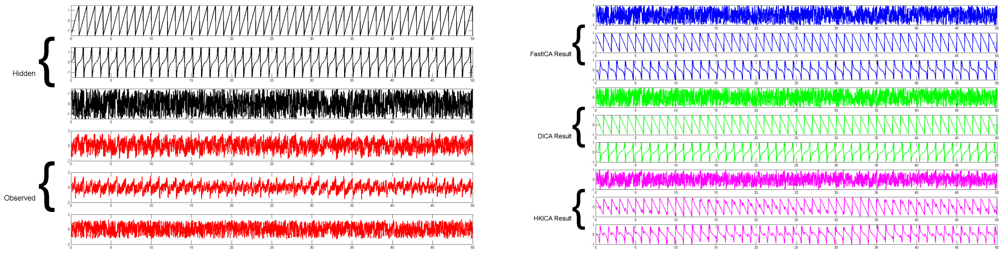

Figure 1: Example of ICA: The black signals are the source signals, red ones observed signals (left). The right figure shows the reconstructed (and rescaled) signals from FastICA, HKICA, and DICA (3 different ICA algorithms, see Section 3.1 and 3 for details ).

then no algorithm can recover the sources from their mixtures. Another question is whether the temporal dependency of the sources may hamper performance. If $s(1) = s(2) = \cdots = s(T)$ then the algorithms effectively need to work with a single vector observation and no algorithm will be able to perform a successful reconstruction. Some data is (as shown) is easy, some will be harder if not impossible to work. What makes then some data easy (or hard) to work with when it comes to independent component analysis? Can we redefine the problem of ICA in a meaningful way so that we can explain the success of the particular ICA algorithms on the above example?

In this short communication we give a summary of our longer paper [Huang et al., 2015b] where we attempted to provide some answers to these questions, while we also streamline the presentation of the results. At a high level, the essence of our approach is to define empirical measures of the "niceness" of data and a measure of the success of the algorithm and then postulating the requirement that good algorithms are those that get better results on "nicer" data. We also require that the niceness measure should behave in a controlled fashion in the classical ICA settings so that the usual statistical results can be recovered from our results. The key feature of the approach is that no probabilistic assumptions are made on the data (the algorithms may randomize though), and thus this work can be thought as the natural extension of online learning where learning algorithms are analyzed without making any probabilistic assumptions [Cesa-Bianchi and Lugosi, 2006].

### 1.1 Notation

We denote the set of real and natural numbers by $\mathbb{R}$ and $\mathbb{N}$, respectively. A vector $v \in K^d$ for a field $K$ is assumed to be a column vector. The 2-norm of $v$ is denoted by $\|v\|_2$ and for any matrix $Z$ we let $\|Z\|_2 = \max_{v:\|v\|_2=1} \|Zv\|_2$ denote the corresponding induced norm. For a tensor (including vectors and matrices) $T$, its Frobenius norm (or $\ell_2$ norm) $\|T\|_F$ is defined as the square root of the sum of the square of all the entries. The transpose of a vector/matrix $Z$ is denoted by $Z^\top$, while the inverse of the transpose is denoted by $Z^{-\top}$. The outer product of two vectors $v, u \in K^d$ is denoted by $u \otimes v = uv^\top$. Let $v^{\otimes k}$ denote the $k$-fold outer product of $v$ with itself, that is, $v \otimes v \otimes v \ldots \otimes v$, which is a k-dimensional tensor. Finally, $\text{Poly}(\cdot, \cdots, \cdot)$ denotes a polynomial function of its argument.

## 2 Main Results

Let $x$, $s$, and $\varepsilon$ be $\mathbb{R}^d$-valued functions from $[T] \doteq \{1, 2, \ldots, T\}$, and $A$ be $d \times d$ nonsingular "mixing" matrix, such that

$$x(t) = As(t) + \varepsilon(t), \qquad 1 \le t \le T\,. \tag{1}$$

We consider the problem of reconstructing $A$ having observed $x$ provided that $(A, s, \varepsilon)$ are "nice" as it will be defined later. Intuitively, $s$ is the source whose components are "independent" while $\varepsilon$ is "noise". We measure how well a matrix $\hat{A}$ returned by an algorithm working on data $x$ recovers

$A$ by the reconstruction error defined as

$$d(\hat{A}, A) = \inf_{\substack{\pi \in \mathrm{Perm}([d]) \\ c \in \mathbb{R}^d}} \max_k ||c_k A_{:\pi(k)} - A_{:k}||_2 ,$$

where $A_{:i}$ stands for the $i$th column of $A$ and $\mathrm{Perm}([d])$ is the set of all the permutations on the set $[d]$. This measure compensates for the inherent indeterminacy in reconstructing the scale and ordering of sources.

Let us now develop the "niceness" measure of the $(x, A, s, \varepsilon)$. Starting from the classical setting, we will define this tuple nicer if the respective *empirical* distributions approximately satisfy the usual assumptions: *a*) the source ($s$) components are independent; *b*) they have high absolute excess kurtosis; *c*) they have zero mean; *d*) the noise ($\varepsilon$) has zero mean; *e*) it has low absolute excess kurtosis; *f*) the noise and source are independent.

For a function $u : [T] \to \mathbb{R}^k$, we let $\nu^{(u)}$ stand for the empirical distribution of $u$, defined by $\nu^{(u)}(B) = \frac{1}{T}|\{t \in [T] : u(t) \in B\}|$ for Borel sets $B \subset \mathbb{R}^k$. For a distribution $\mu$ over the reals we let $\kappa(\mu)$ be the (absolute excess) kurtosis of $\mu$: $\kappa(\mu) = |\int x^4 \mu(dx) - 3(\int x^2 \mu(dx))^2|$. For a product distribution $\mu = \mu_1 \otimes \ldots \otimes \mu_d$ over $\mathbb{R}^d$, we let $\kappa_{\min}(\mu) = \min_{1 \le i \le d} \kappa(\mu_i)$ to denote the minimum kurtosis of the components of $\mu$. When $\mu$ is a distribution over $\mathbb{R}^d$, we define the $d$-dimensional excess absolute kurtosis of $\mu$ by $\kappa^2(\mu) = \max_{1 \le i,j \le d} \sum_{k,l} \left\{M_{i,j,k,l}(\mu) - (M_{i,j}(\mu)M_{k,l}(\mu) + 2M_{i,k}(\mu)M_{j,l}(\mu))\right\}^2$, where $M_{a,\ldots,z}(\mu) = \int y_a \ldots y_z \mu(dy)$. We will also use $N(\nu) = \| \int x\nu(dx)\|$.

To measure the degree of independence of the components of the source $s$, we define a family of "distances" between distributions (strictly speaking, these are only pseudo-distances). In particular, given two distributions $\nu_1$ and $\nu_2$ over $\mathbb{R}^d$ and an integer $k \ge 1$, we let $D_k(\nu_1, \nu_2) = \sup_{f \in \mathcal{F}} |\int f(s)\nu_1(ds) - \int f(s)\nu_2(ds)|$, where $\mathcal{F} = \{f : \mathbb{R}^d \to \mathbb{R} : f(s) = \prod_{j=1}^k s_{i_j}, 1 \le i_1, \ldots, i_k \le d\}$ is the set of all monomials up to degree $k$. When $\mu$ is a product measure, $D_k(\mu, \nu)$ measures how close the components of $X \sim \nu$ are to being independent. When $\nu$ is a measure of $p + q$ variables (i.e., $X \in \mathbb{R}^{p+q}$) we also need a measure that quantifies the degree of independence of the vectors $(X_1, \ldots, X_p)$ and $(X_{p+1}, \ldots, X_{p+q})$. We will denote this measure by $D_k^{(p,q)}(\nu)$ and is defined as $D_k^{(p,q)}(\nu) = \inf_{\mu_1,\mu_2} D_k(\mu_1 \otimes \mu_2, \nu)$, where $\mu_1$ ranges over all measures on $\mathbb{R}^p$ and $\mu_2$ ranges over all measures on $\mathbb{R}^q$. Finally, we let

$$L = \max\left(\| \int [y^{\otimes 2}]\nu^{(\varepsilon)}(dy)\|_F,\ \| \int [y^{\otimes 3}]\nu^{(\varepsilon)}(dy)\|_F\right),$$

which captures the magnitude of second and third moments of the noise. We let $\Pi_0$ to be the set of zero mean product distributions over $\mathbb{R}^d$.

Now we are ready to state our main result.

**Theorem 2.1.** *There exists a randomized algorithm such that for any $A \in \mathbb{R}^{d \times d}$, and $x, s, \varepsilon : [T] \to \mathbb{R}^d$ satisfying Equation* (1)*, the algorithm returns $\hat{A}$ such that with probability at least $1 - \delta$,*

$$d(\hat{A}, A) \le \inf_{\mu \in \Pi_0} \mathcal{C}(\mu) \min\left(D_4(\nu^{(s)}, \mu) + \kappa(\nu^{(\varepsilon)}) + D_4^{(d,d)}(\nu^{(As,\varepsilon)}) + N(\nu^{(\varepsilon)}) + N(\nu^{(s)}), \Theta(\mu)\right),$$

*where $\mathcal{C}(\mu)$ and $\Theta(\mu)$ are problem dependent constants, polynomial in $(\sigma_{\max}(A), 1/\sigma_{\min}(A), 1/\kappa_{\min}(\mu), 1/\delta, d, L)$. Further, the computational complexity of the algorithm is $O(d^3T)$ when used on any data $x$ of dimensions $T \times d$.*

Note that the above result implies that in the standard stochastic setting with independent sources and Gaussian noise, independently generated from the sources, with probability at least $1 - \delta$,

$$d(\hat{A}, A) \le \mathcal{C} \min\left(\tfrac{1}{\sqrt{T}}, \Theta\right),$$

for some problem dependent constants $\mathcal{C}$ and $\Theta$. The next section describes the algorithm, introduced in our previous paper, that achieves this bound.

# 3 A “Deterministic” ICA Algorithm

The algorithm builds on the works of Frieze et al. [1996], Hsu and Kakade [2013], Arora et al. [2012]. For a measure $\mu$ over $\mathbb{R}^d$, define $g_\mu, f_\mu : \mathbb{R}^d \to \mathbb{R}$ by $g_\mu(u) = \int (u^\top x)^4 \mu(dx) - 3(\int (u^\top x)^2 \mu(dx))^2$ and $f_\mu(u) = \int (u^\top As)^4 \mu(ds) - 3(\int (u^\top As)^2 \mu(ds))^2$. One can show that if $\mu = \mu_1 \otimes \cdots \otimes \mu_d$ is a product measure then for any $u \in \mathbb{R}^d$, $\nabla^2 f_\mu(u) = AKD_uA^\top$, where $D_u = \operatorname{diag}\left((u^\top A_1)^2, \cdots, (u^\top A_d)^2\right)$ and $K = \operatorname{diag}(\ldots, \kappa(\mu_i), \ldots)$. Fix some $\psi \in \mathbb{R}^d$ and let $B$ be an arbitrary “square root” of matrix $\nabla^2 f_\mu(\psi)$. Thus, $\nabla^2 f_\mu(\psi) = BB^\top$ and $B = AK^{1/2}D_\psi^{1/2}R^\top$ for some orthonormal matrix $R$. Now choose $\phi_i \in \mathbb{R}^d$, $i = 1, 2$ distinct vectors. Defining $T_i = \nabla^2 f_\mu(B^{-\top}\phi_i)$, one can calculate that $T_i = AK^{1/2}D_\psi^{-1/2}\Lambda_i A^\top$ where $\Lambda_i = \operatorname{diag}\left((\phi_i^\top R_1)^2, \ldots, (\phi_i^\top R_d)^2\right)$ and $R_i$ denotes the $i$th column of $R$. Then $M = T_1T_2^{-1} = A\Lambda A^{-1}$ with $\Lambda = \Lambda_1\Lambda_2^{-1} = \operatorname{diag}\left(\left(\frac{\phi_1^\top R_1}{\phi_2^\top R_1}\right)^2, \ldots, \left(\frac{\phi_1^\top R_d}{\phi_2^\top R_d}\right)^2\right)$. Thus, $A_i$ are the eigenvectors of $M$. Note that the eigenvalues of $M$ are defined in terms of the orthogonal matrix $R$, and so it is easy to handle the resulting minimum spacing

$$\gamma_R = \min_{i,j:i\neq j} \left| \left(\frac{\phi_1^\top R_i}{\phi_2^\top R_i}\right)^2 - \left(\frac{\phi_1^\top R_j}{\phi_2^\top R_j}\right)^2 \right|.$$

We show in the full version [Huang et al., 2015a] that, when $\phi_1, \phi_2, \psi$ are independent standard $d$-dimensional normally distributed random vectors then $\gamma_R \geq \frac{\delta}{2d^2}$ with probability at least $1 - \delta$. The resulting algorithm, called Deterministic ICA (DICA), is shown in Algorithm 1. The name is somewhat misleading, as the algorithm randomizes; the intention is to emphasize that the algorithm does not require probabilistic assumptions on the data.

**Algorithm 1** Deterministic ICA (DICA)

```
input  $x(t)$ for $1 \leq t \leq T$.
output  An estimation of the mixing matrix $A$.
  Sample $\psi, \phi_1, \phi_2$ from a $3d$-dimensional standard Gaussian distribution;
  Evaluate $\nabla^2 \hat{g}(\psi)$ where $\hat{g} = g_{\nu(x)}$;
  Compute $\hat{B}$ such that $\nabla^2 \hat{g}(\psi) = \hat{B}\hat{B}^\top$;
  Compute $\hat{T}_1 = \nabla^2 \hat{g}(\hat{B}^{-\top}\phi_1)$ and $\hat{T}_2 = \nabla^2 \hat{g}(\hat{B}^{-\top}\phi_2)$;
  Compute the eigenvectors $\{\mu_1, \ldots, \mu_d\}$ of $\hat{M} = \hat{T}_1\hat{T}_2^{-1}$;
  Return $\hat{A} = \{\mu_1, \ldots, \mu_d\}$.
```

## 3.1 Related Works

Perhaps FastICA is the most popular ICA method [Hyvarinen, 1999], which is quite well understood by now. In particular, recently Miettinen et al. [2014] showed that in the noise-free case (i.e., when $X = AS$), FastICA’s error (when it uses a particular fourth-moments-based contrast function) vanishes at a rate of $1/\sqrt{T}$. Similar results (mostly asymptotic) exists for several other ICA methods [e.g., Eriksson and Koivunen, 2003, Samarov et al., 2004, Chen and Bickel, 2005, Chen et al., 2006], However, no finite-sample bounds are available for these methods in the “noisy” case, which seems to better suit the so-called moment methods, a class that our method also belongs to.

While recent years have seen much advance in the design and analysis of these methods, up to the present work all these works lacked in some respects. In particular, the algorithms of Arora et al. [2012] and Goyal et al. [2014] make use of some parameters ($\beta$ in the paper of Arora et al. [2012] and $\sigma$ in the paper of Goyal et al. [2014]) and their guarantees hold only when these parameter belong to some unobservable interval of an uncontrolled measure, preventing the easy escape of choosing these parameters randomly. In other methods [Anandkumar et al., 2012a,b, Hsu and Kakade, 2013] a common problem is that the minimum gap between the eigenvalues is uncontrolled; it could potentially be exponentially large in $d$. Although this is avoided by Vempala and Xiao [2014], their algorithm still inherits a free parameter from Goyal et al. [2014]. The problem of separating mixture of deterministic signals is also consider in [Kirimoto et al., 2011] and [Forootan and Kusche, 2013]. But their analysis is restricted to particular signals, while our result is applicable to general ones.

## References

A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky. Tensor decompositions for learning latent variable models. *CoRR*, abs/1210.7559, 2012a.

A. Anandkumar, D. Hsu, and S. M. Kakade. A method of moments for mixture models and hidden markov models. *arXiv preprint arXiv:1203.0683*, 2012b.

S. Arora, R. Ge, A. Moitra, and S. Sachdeva. Provable ica with unknown gaussian noise, with implications for gaussian mixtures and autoencoders. In *Advances in Neural Information Processing Systems*, pages 2375–2383, 2012.

N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.

A. Chen and P. J Bickel. Consistent independent component analysis and prewhitening. *Signal Processing, IEEE Transactions on*, 53(10):3625–3632, 2005.

A. Chen, P. J Bickel, et al. Efficient independent component analysis. *The Annals of Statistics*, 34 (6):2825–2855, 2006.

A. Dermoune and T. Wei. FastICA algorithm: Five criteria for the optimal choice of the nonlinearity function. *IEEE transactions on signal processing*, 61(5-8):2078–2087, 2013.

J. Eriksson and V. Koivunen. Characteristic-function-based independent component analysis. *Signal Processing*, 83(10):2195–2208, 2003.

Ehsan Forootan and Jürgen Kusche. Separation of deterministic signals using independent component analysis (ica). *Studia Geophysica et Geodaetica*, 57(1):17–26, 2013.

A. Frieze, M. Jerrum, and R. Kannan. Learning linear transformations. In *37th IEEE Annual Symposium on Foundations of Computer Science*, pages 359–359. IEEE Computer Society, 1996.

N. Goyal, S. Vempala, and Y. Xiao. Fourier PCA and robust tensor decomposition. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 584–593. ACM, 2014.

D. Hsu and S. M. Kakade. Learning mixtures of spherical Gaussians: moment methods and spectral decompositions. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 11–20. ACM, 2013.

R. Huang, A. György, and Cs. Szepesvári. Deterministic independent component analysis. in preparation, 2015a.

R. Huang, A. György, and Cs. Szepesvári. Deterministic independent component analysis. In *ICML*, pages 2521–2530, 2015b.

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

Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent Component Analysis*. Wiley, 2001.

Tetsuo Kirimoto, Takeshi Amishima, and Atsushi Okamura. Separation of mixtures of complex sinusoidal signals with independent component analysis. *IEICE transactions on communications*, 94(1):215–221, 2011.

S. Miettinen, J.and Taskinen, K. Nordhausen, and H. Oja. Fourth moments and independent component analysis. *arXiv preprint arXiv:1406.4765*, 2014.

E. Oja and Z. Yuan. The FastICA algorithm revisited: Convergence analysis. *Neural Networks, IEEE Transactions on*, 17(6):1370–1381, 2006.

E. Ollila. The deflation-based FastICA estimator: statistical analysis revisited. *Signal Processing, IEEE Transactions on*, 58(3):1527–1541, 2010.

A. Samarov, A. Tsybakov, et al. Nonparametric independent component analysis. *Bernoulli*, 10(4): 565–582, 2004.

P. Tichavsky, Z. Koldovsky, and E. Oja. Performance analysis of the FastICA algorithm and Cramér-Rao bounds for linear independent component analysis. *Signal Processing, IEEE Transactions on*, 54(4):1189–1203, 2006.

S. Vempala and Y. Xiao. Max vs min: Independent component analysis with nearly linear sample complexity. *CoRR*, abs/1412.2954, 2014. URL `http://arxiv.org/abs/1412.2954`.

T. Wei. The convergence and asymptotic analysis of the generalized symmetric FastICA algorithm. *arXiv preprint arXiv:1408.0145*, 2014.