Independent Component Analysis

Introduction to Probabilistic Graphical Models and Deep Generative Models

Nour Bouayed, Yvann Le Fay, Zineb Bentires

December 2023

Abstract

Independent Component Analysis (ICA) is a special case of blind source separation (BSS) methodology wherein the objective is to recover an unknown mixing process f from noisy mixed observations of the process $\mathbf{x} = f(\mathbf{s}) + \boldsymbol{\varepsilon}$ for some statistically independent sources $\mathbf{s}$ and some noise $\boldsymbol{\varepsilon}$. It typically proceeds by maximising proxies of the statistical independence of the estimated components such as the opposite of the mutual information of the estimates. In this paper, we review existing frameworks which have been introduced over the past two decades, with a focus on the linear mixing process case, $\mathbf{x} = A\mathbf{s} + \boldsymbol{\varepsilon}$. We further exhibit the existing links between ICA and identifiability theory, allowing for exact inference of both, the sources and the latent mixing process using more recent techniques such as Variational Autoencoders. Finally, we experimentally validate the ICA algorithms on toy models, showcasing reasonable results on real-world applications.

1 Introduction

Assume there are m conversations held by several people in a room, and n microphones recording the background sound. In that case, the records can be seen as a complex transformation, let it be f, which is dependent on the room's shape, size and acoustics as well as the orientation of the microphones with respect to the people. The blind source separation problem consists in performing statistical inference to recover the original conversations from the sound records. This problem, known as the *cocktail party problem* was first introduced by Cherry [1953] and later used as a motivation by Hyvärinen and Oja [2000] for developing the framework of ICA. Let us assume we observe n signals $\mathbf{x}(t) = [x_1(t), \dots, x_n(t)]^\top \in \mathbb{R}^n$ which we assume to be a transformation of some latent sources $\mathbf{s}(t) = [s_1(t), \dots, s_m(t)]^\top \in \mathbb{R}^m$ through an unknown but injective mixing process $f : \mathbb{R}^m \to \mathbb{R}^n$ plus an additional independent noise $\boldsymbol{\varepsilon}$, that is,

$$\mathbf{x} = f(\mathbf{s}) + \boldsymbol{\varepsilon}. \tag{1}$$

In the context of ICA, we further make the assumption the s_i's are statistically independent and non-Gaussian variables. Given some observed data $\mathbf{X} = [\mathbf{x}(t_1), \dots, \mathbf{x}(t_T)] \in \mathbb{R}^{n \times T}$, we are interested in estimating the mixing process such that the estimated sources are as independent as possible using different proxies, such as maximising non-Gaussianity measures or minimising the mutual information. Once the transformation is estimated, the sources are obtained by applying the inverse transformation to the observations.

While this statistical framework is a reasonable starting point to perform estimation of the sources, it has some drawbacks. 1) It is built upon assumptions which are difficult to test in real-world scenarios: the non-Gaussianity assumption is reasonable but the independence assumption is difficult to test for. Moreover, the law of $\boldsymbol{\varepsilon}$ and the number of sources have to be specified to perform estimation. 2) It suffers from the identifiability issue: Without further knowledge on f, we are only able to recover the sources up to some transformations that keep invariant (1).

The remainder of this report is organised as follows:

1. In section 2, we focus on the linear case where the observed signals are assumed to be a linear mixture of the sources. In particular, we review the maximum likelihood approach, upon which, the gradient descent (GD) 1 and the FastICA 2 algorithms are built. This section is mostly based on Hyvärinen and Oja [2000].

2. In section 3, we relax the linear hypothesis and show inference is still possible under technical hypotheses and in the presence of observed auxiliary variables, using more recent techniques of Bayesian variational inference and Variational Autoencoders. This section is entirely based on Khemakhem et al. [2020].

3. In section 4, we provide numerical evidence for the statistical performances of, first, the FastICA algorithm and its extension, and second, the VAE approach.

4. Extensions and derivations of the methods are given in the Appendix section 6. The individual contributions are detailed in 7.

2 Linear ICA

The ICA approach to BSS was first proposed by Hyvärinen and Oja [2000], in which the observation process is assumed to be a noise-free linear invertible transformation of the sources, in that case (1) becomes

$$\mathbf{x} = A\mathbf{s}, \tag{2}$$

where $A \in \mathbb{R}^{n\times m}$ is assumed to admit a pseudo-inverse $W = (A^\top A)^{-1}A^\top \in \mathbb{R}^{m\times n}$, i.e.,

$$\mathbf{s} = W\mathbf{x}. \tag{3}$$

To address the scale-indeterminacy issue, we assume the data $\mathbf{X}$ is centred with unit variance and we restrict ourselves to orthogonal matrices W. Hyvärinen and Oja [2000] propose to estimate W such that the recovered sources have minimal mutual information. Indeed, we know the mutual information of the estimated sources will be zero, if and only if, its components are statistically independent, and is otherwise, non-negative, see (10). Thus, the ICA can be seen as a minimisation program of the mutual information of $W\mathbf{x}$ over the set of full-rank matrices $W \in \mathbb{R}^{m\times n}$. Under the orthogonality assumption, minimising the mutual information is equivalent to the maximum likelihood estimator (MLE) for W, (see 2.3). For the sake of simplifying the notation, we do not distinguish the true unmixing matrix, W, from any intermediary estimated unmixing matrix.

2.1 The maximum likelihood estimator (MLE)

Let the p_i's denote the source densities, let $G(\mathbf{s}) = -(\log p_1(s_1), \dots, \log p_m(s_m))$ and $g = (\partial_{s_i} G_i)_{1\le i\le m}$. For the sake of simplicity, we restrict ourselves to $W = [w_1, \dots, w_m]^\top$ being a $m \times m$-square matrix. This allows us to perform a change of variable, we have $p(\mathbf{x} \mid W) = |\det W| \prod_{i=1}^m p_i(w_i^\top \mathbf{x})$, thus the (negative) log-likelihood for W is given by

$$l(W \mid \mathbf{x}) = \mathbb{E}\left(\sum_{i=1}^m G_i(w_i^\top \mathbf{x})\right) - \log|\det W|. \tag{4}$$

The gradient descent ΔW step for this objective function is given by

$$\Delta W = (I - \mathbb{E}[g(W\mathbf{x})(W\mathbf{x})^\top])W, \tag{5}$$

(see Appendix 6.3 for proper derivations). Replacing the expectation by empirical mean over the data $\mathbf{X}$ and performing a Gram-Schmidt orthogonalisation at each step to ensure orthonormality of the rows of W, we obtain Algorithm 1.

Algorithm 1: Gradient descent (GD) Algorithm for the MLE

input : Whitened mixed signals $\mathbf{X} \in \mathbb{R}^{n\times T}$
$g = (\partial_{s_i} G_i)_{1\le i\le m}$ where G_i is minus the logarithm of the density for the i-th source
Tolerance $\varepsilon > 0$, learning rate $\gamma > 0$ `// A backtracking line search could be performed`
output: Estimated inverse of the mixing matrix A, $W \in \mathbb{R}^{m\times m}$
Initialise a matrix $W^{(0)}$ with orthonormal rows
$k \leftarrow 0$
while $||\Delta W||_F > \varepsilon$ **do**
  $\Delta W \leftarrow (I - 1/T g(W^{(k)}\mathbf{X})(W^{(k)}\mathbf{X})^\top)W^{(k)}$
  $W^{(k+1)} \leftarrow W^{(k)} + \gamma \Delta W$ `// Gradient descent`
  Orthonormalise the rows of $W^{(k+1)}$ `// Gram-Schmidt procedure`
  $k \leftarrow k + 1$
end
return $W^{(k)}$

Since the true generative process determined by the g_i's, is not known, we instead resort to a heuristic based on discriminating sub-Gaussian from super-Gaussian sources. Let $g_+(x) = \tanh(x) + x$, which corresponds to a super-Gaussian prior with density function $p_+(x) \propto \exp(-\ln\cosh(x) - x^2/2)$, similarly, let $g_-(x) = -\tanh(x) + x$, which corresponds to a sub-Gaussian prior with density $p_-(x) \propto \exp(\ln\cosh(x) - x^2/2)$ (for the plots of the densities, see Appendix 6.2). Depending on the estimated sub or super-Gaussianity of $w_i^\top \mathbf{x}$, we assign s_i to have the density p_- or p_+. For example, if we assign the $g_\pm$ depending on the Kurtosis, then the learning rule (5) becomes $\Delta W = [I - \mathbb{E}[(\text{sgn}(k_4)\tanh(W\mathbf{x}) + W\mathbf{x})(W\mathbf{x})^\top]]W$, where $\text{sgn}(k_4)$ is the diagonal matrix with entries the sign of the normalised kurtosis for each sources, i.e., $\text{sgn}(k_4) = \text{diag}[\text{sgn}(\mathbb{E}[(w_i^\top \mathbf{x})^4] - 3)]$.

2.2 FastICA

Hyvärinen and Oja [2000] derive a faster estimation algorithm by replacing the GD step by an approximated Newton's step. For any row $w = w_i$ of W, we minimise the penalised objective f given by

$$f(w) = \mathbb{E}[G_i(w^\top \mathbf{x})] + \lambda(1 - w^\top w), \tag{6}$$

where λ is the Lagragian multiplier for the normality constraint on w. The Newton update operation $w \mapsto w^*$ is (see Appendix 6.4 for the complete proof):

$$w^* = \mathbb{E}[\mathbf{x} g_i(w^\top \mathbf{x})] - \mathbb{E}[g_i'(w^\top \mathbf{x})]w. \tag{7}$$

Right after performing this update for the row w_i, we run an online Gram-Schmidt step to ensure the rows of W form an orthogonal set, then project back onto the constraint surface by normalising w_i. Replacing the expectation by empirical mean over the data $\mathbf{X}$, we obtain the FastICA Algorithm 2 given in Appendix 6.4.

2.3 Other estimation principles

Besides maximum likelihood methods, there are several other approaches for estimating W, which are shown to be equivalent to the former:

1. **Maximising non-Gaussianity**, through metrics like kurtosis and negentropy. The latter is null only for Gaussian distributions and is otherwise always non-negative. Our objective is finding W that maximises the negentropy J of $\mathbf{s} = W\mathbf{x}$:
$$J(W) = H(N_\mathbf{s}) - H(\mathbf{s}), \tag{8}$$
where $N_\mathbf{s} \sim \mathcal{N}(\mathbb{E}[\mathbf{s}], \text{var}(\mathbf{s}))$ and $H(Z)$ is the entropy defined as $H(Z) = -\mathbb{E}_{Z\sim p}[\log p(Z)]$ for any random variable Z with density p. The first term is a constant because the data is centred and whitened. Using the independence of the sources, we have
$$J(W) = -H(\mathbf{s}) + \text{const} = -\mathbb{E}\left(\sum_{i=1}^m G_i(w_i^\top \mathbf{x})\right) + \text{const}, \tag{9}$$
which we see is equal (up to constants with respect to W) to the log-likelihood (4).

2. **Minimising mutual information**, which measures the inter-dependence of a set of random variables:
$$I(\mathbf{s}) = \sum_{i=1}^m H(s_j) - H(\mathbf{s}). \tag{10}$$
If we constrain W to be orthogonal, we can drop the last term, since then $H(W\mathbf{x}) = H(\mathbf{x})$ which is a constant with respect to W. Minimising (10) is equivalent to maximising the negentropy, which is equivalent to maximum likelihood.

3 Non-linear case and variational inference through the use of Variational Autoencoders (VAEs)

In recent years, the intersection of deep learning and signal processing has led to innovative approaches for a multitude of tasks, including signal disentanglement using latent representation in data. Khemakhem et al. [2020] propose to

use Variational Autoencoders (VAEs) to retrieve the sources by estimating the mixing process. Recall the model assumption (1)

$$\mathbf{x} = f(\mathbf{s}) + \boldsymbol{\varepsilon}, \tag{11}$$

which can be rewritten as

$$p_\theta(\mathbf{x}, \mathbf{s}) = p_f(\mathbf{x} \mid \mathbf{s}) p_{T,\lambda}(\mathbf{s}). \tag{12}$$

Thus, we are interested in recovering the parameter of interest $\theta = (f, T, \lambda)$, where f is the mixing process, T and λ parametrize the generative model of the sources $\mathbf{s}$. The VAE framework allows the estimation of θ and provides a variational approximation of the posterior $q_\phi(\mathbf{s} \mid \mathbf{x})$ of $p_\theta(\mathbf{s} \mid \mathbf{x})$. For more details on the VAE framework, we refer the reader to Appendix 6.5. To perform the estimation of θ, Khemakhem et al. [2020] additionally assume we have access to an auxiliary observed variable $\mathbf{u}$ upon which the distribution of $\mathbf{s}$ depends on. Consequently, conditioning the model (12) on the latent variables $\mathbf{u}$, the model becomes

$$p_\theta(\mathbf{x}, \mathbf{s} \mid \mathbf{u}) = p_f(\mathbf{x} \mid \mathbf{s}) p_{\mathbf{T},\lambda}(\mathbf{s} \mid \mathbf{u}). \tag{13}$$

Assuming the inference model is well-specified, i.e., $\{q_\varphi(\mathbf{s} \mid \mathbf{x}, \mathbf{u})\}_\varphi$ contains the true posterior $p_\theta(\mathbf{s} \mid \mathbf{x}, \mathbf{u})$ and under technical assumptions on the distribution of $\mathbf{s} \mid \mathbf{u}$, Khemakhem et al. [2020] prove the VAE identifies the true parameter θ, up to a linear invertible transformation in the general case, and under some additional assumptions, up to a permutation and a sign (see Th. 6.1 given in Appendix 6.5).

4 Experiments

The code to reproduce the experiments listed below can be found at the following address: https://github.com/hallelujahylefay/independent_component_analysis. It includes an implementation of the FastICA algorithm, an implementation of the gradient descent algorithm for the MLE estimator, and a JAX iVAE implementation built upon the companion code of Khemakhem et al. [2020][1]. The code is test-based and was designed for just-in-time (JIT) compilation.

In order to assess ICA's capability in separating mixed signals into their constituent components, we conducted several numerical experiments by generating sources, mixing them using a linear or non-linear process, and then applying ICA.

4.1 Experiments on Linear ICA problems using FastICA algorithm

4.1.1 Experiments on synthetic signals

We created controlled mixtures of known univariate signals, which we first ensured were non-Gaussian by visualising QQ-plots of their standardised distributions (see Figure 6), which deviate from the standard normal distribution.

After applying the FastICA Algorithm 2 on the mixed signals, we get the estimated sources depicted in Figure 1, to each of which we associate a unique source among the original ones, by solving a linear sum assignement problem to maximise the overall correlation between each possible pairing.

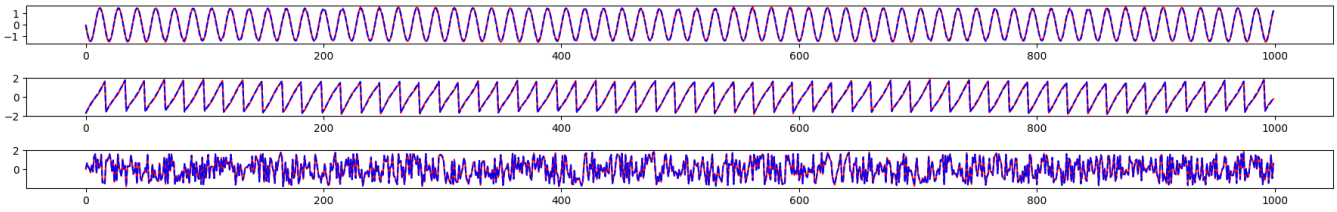


Figure 1: Recovered sources (in blue) superposed to their best matching original source after whitening (in red).

We can see the algorithm has well learnt the invert mixing process, starting from a random initial unmixing matrix. Indeed, we could visualise during the iterations of FastICA the increase of the Mean Correlation Coefficient (MCC) between true and estimated sources as shown in Figure 2.

We repeated the experiment using the GD Algorithm 1, see Figure 7.

[1] available at https://github.com/ilkhem/iVAE

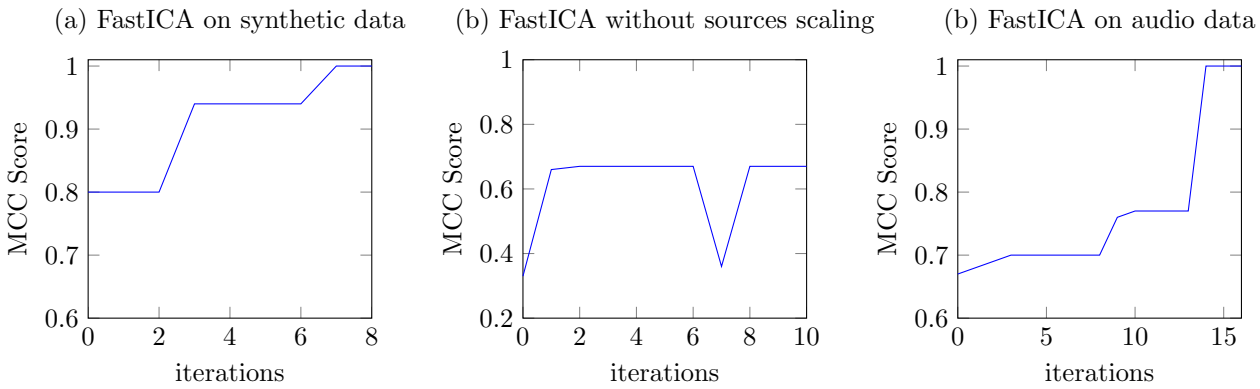
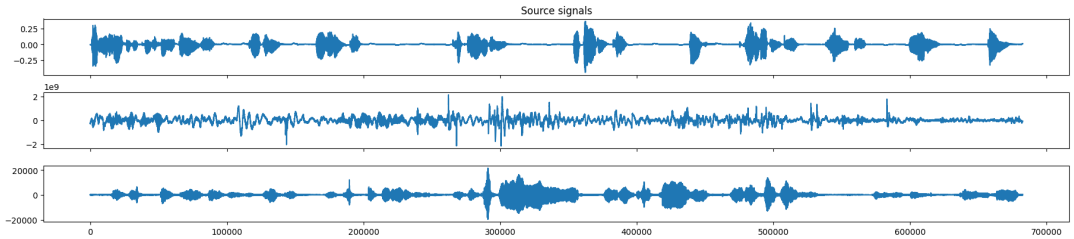


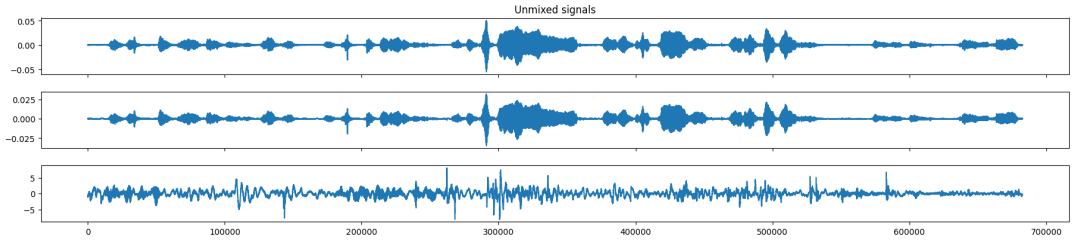
Figure 2: Evolution of the Mean Correlation Coefficient between true sources and estimated sources when using FastICA

4.1.2 Experiments on real audio signals

In order to test how the algorithm performs on real data, we uploaded three speech recordings which we mixed linearly using a random (3×3) matrix[2]. The original and mixed signals are displayed below, see Figure 3.



(a) Source signals (unscaled).



(b) Estimated source signals.

Figure 3: Blind Sources Separations for Audio Signals

The results presented suggest that when Independent Component Analysis (ICA) is employed on a mixture of source signals with disproportionate magnitudes, the algorithm encounters difficulties in effectively isolating components of significantly lower amplitude compared to others. This situation is similar to the difficulties experienced in the cocktail party problem, where picking out a distant conversation in a loud and crowded room is hard. The struggle to detect a fainter source is reflected in the lower MCC scores, as shown in Figure 2 in scenario (b).

Testing on the same audio sources which we reduce to the same scale order before mixing, we get a better result, and the algorithm recovers all sources, see Appendix 6.6.2. This shows the sensitivity of FastICA to variations in source magnitude.

4.1.3 Practical considerations

Working on real data imposed some constraints on the implemented algorithms' computational complexity. Mainly, we adapted two parts of the method to our need :

[2]To listen to the original audio signals, their mixtures and the result of ICA algorithm on this problem, visit https://github.com/ylefay/independent_component_analysis/tree/main/experiments

- Firstly, the Mean Correlation Coefficient computation, which normally entails solving a linear sum assignment problem to assign each estimated source to the true source that best correlates with it, thus testing all permutations. We instead used a greedy approach on the correlation matrix (between true and estimated sources): iteratively, we retrieved the maximum of each row of the correlation matrix, removing afterwards the selected column index from subsequent iterations to obtain a bijection between indices of the estimated sources and the true sources.
- Secondly, to accelerate the FastICA algorithm, we implemented an early stopping mechanism, which stopped the algorithm inner iterations (on a given component of the unmixing matrix) when correlation between the updated and previous versions of the component remained constant.

4.2 Experiments on Non Linear ICA problems using iVAE

We repeated the experiments conducted by Khemakhem et al. [2020] on synthetic data. We generated a number of non-stationary Gaussian univariate signals, which we divided into $M = 40$ segments of $L = 1000$ samples each. The conditioning variable $\mathbf{u}$ is the segment label, and its distribution is uniform on the integer set $[\![1, M]\!]$. Within each segment, the conditional prior distribution $s \mid u$ is Gaussian. Following Khemakhem et al. [2020], we mix the sources using a multi-layer perceptron (MLP) with an hidden dimension of 100 and add no noise. We implemented the iVAE model as described in the original paper, and trained it using similar values for the hyperparameters. The learning rate for the Adam optimiser was set to 0.01. The batch size is 64, and we trained the model during 200 epochs. The obtained MCC performance curve Figure 4. We can see the model is indeed learning latent variables that are closer and closer to the sources. However, we did not manage to reach the performance that was described by the authors (with an MCC of 0.9 which we were able to reproduce using their code). We believe we were not able to obtain high MCC because of the optimiser. A learning-rate scheduler could be implemented to avoid stagnation of the MCC.

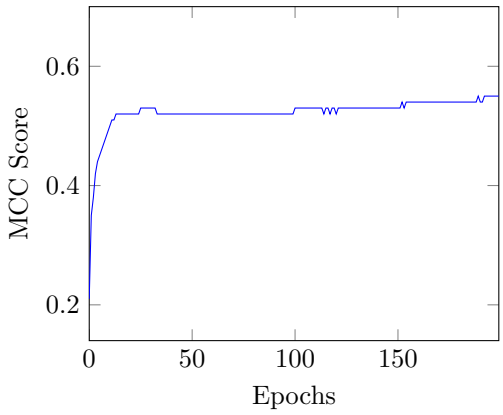


Figure 4: Evolution of MCC during iVAE training on synthetic data

5 Conclusion

This report has presented a comprehensive evaluation of Independent Component Analysis (ICA), covering both linear and nonlinear mixing processes. In the linear case, the FastICA and the Gradient descent are simple but convenient and powerful algorithms for tackling the Blind source separation problem. Indeed, our theoretical and empirical analyses underscore the efficiency of those algorithms. However, the linear mixing model is a strong assumption which is rarely encountered in real-world datasets. Advanced techniques, such as VAEs, enable source retrieval for a non-linear mixing process. Indeed, the VAE method offers a more flexible approach, particularly useful when dealing with complex data distributions. However, the success of this method hinges on certain technical assumptions about the generative model as well as the presence of auxiliary variables. Yet, we believe the model assumption is flexible since It includes approximations of a broad class of distributions.

6 Supplementary Material

6.1 Justification of non-Gaussianity hypothesis

One key assumption is that none of the sources is Gaussian. To understand why, let us restrict ourselves to 2 latent and observed signals with $s = (s_1, s_2) \sim \mathcal{N}(0, I_2)$. Then for any orthogonal matrix A, i.e., such that $AA^\top = I_2$, we have

$$\mathbf{x} = A\mathbf{s} \sim \mathcal{N}(0, I_2). \tag{14}$$

Thus, A is not identifiable: the joint distribution is completely invariant by rotation or symmetry.

6.2 Examples of sub-Gaussian and super-Gaussian distributions

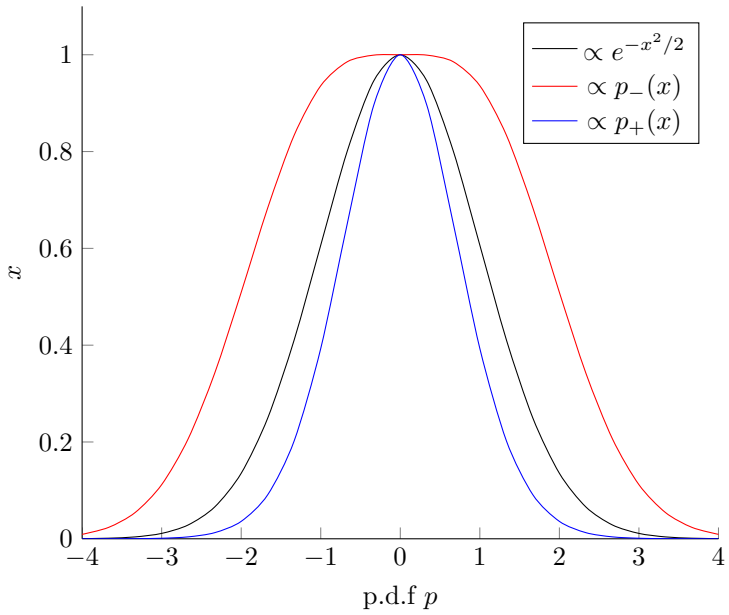


Figure 5: Illustration of sub-Gaussian, super-Gaussian and Gaussian density functions

6.3 Deriving the gradient descent (GD) for the MLE

The equation of the likelihood given in (4) enables us to have the gradient descent given by the step ΔW in equation (5). The derivative of the negative log-likelihood (4) is

$$\partial_W l(W \mid \mathbf{x}) = \mathbb{E}(g(W\mathbf{x})\mathbf{x}^\top) - (W^\top)^{-1}, \tag{15}$$

where g is the component-wise derivative function of G, $g(\mathbf{s}) = (g_1(s_1), \ldots, g_m(s_m))^\top$ and $g_i = G_i'$. With the convention of a gradient step given by $W \leftarrow W + \gamma \Delta W$ with $\gamma > 0$, we have

$$\Delta W = -\partial_W l(W \mid \mathbf{x}) = (W^\top)^{-1} - \mathbb{E}(g(W\mathbf{x})\mathbf{x}^\top). \tag{16}$$

Multiplying by the right, both sides by $W^\top W \approx I$, we get (5). As a minor contribution, we derive the gradient descent update in the general case where $m \leq n$, which we have not been able to find. Suppose W is a full-rank $m \times n$ matrix. Using the generalisation of Jacobi's change of variable formula given in [Ben-Israel, 1999], we have $p(\mathbf{x} \mid W) = \sqrt{\det WW^\top} \prod_{i=1}^m p_i(w_i^\top \mathbf{x})$. Using $\nabla(\log \det WW^\top) = 2(WW^\top)^{-1}W$, (16) becomes

$$\Delta W = (WW^\top)^{-1}W - \mathbb{E}(g(W\mathbf{x})\mathbf{x}^\top). \tag{17}$$

Replacing the gradient incremental step in 1 by the latter gives rise to a new GD algorithm in the general setup.

6.4 Deriving the FastICA algorithm

We rely on Murphy [2012] and Hyvärinen and Oja [2000] for the following derivations. Restricting the problem on one source estimation, the constrained objective, its gradient, and Hessian are given by

$$f(w) = \mathbb{E}[G(w^\top x)] + \lambda(1 - w^\top w), \quad \nabla f(w) = \mathbb{E}[xg(w^\top x)] - 2\lambda w, \quad \nabla^2 f(w) = \mathbb{E}[xx^\top g'(w^\top x)] - 2\lambda I, \tag{18}$$

where λ is the Lagrange multiplier. The Newton's step is $w \mapsto w^* = w - \Delta w$ where $\Delta w = -[\nabla^2 f(w)]^{-1}\nabla f(w)$, i.e.,

$$w^* \approx \mathbb{E}[xg(w^\top x)] - \mathbb{E}[g'(w^\top x)]w, \tag{19}$$

where we made the useful approximation $\mathbb{E}[xx^\top g'(w^\top x)] \approx \mathbb{E}[xx^\top]\mathbb{E}[g'(w^\top x)] = \mathbb{E}[g'(w^\top x)]$. After performing this update, we project back w^* onto the constraint surface by normalising it.

Algorithm 2: FastICA Algorithm

input : Whitened mixed signals $\mathbf{X} \in \mathbb{R}^{n \times T}$
$g = \nabla G$
Tolerance $\varepsilon > 0$
output: Estimated inverse of the mixing matrix A, $W = [w_1, \dots, w_m]^\top \in \mathbb{R}^{m \times n}$
Initialise a matrix $W^{(0)}$ with orthonormal rows
for $i = 1$ **to** m **do**
  $k \leftarrow 0$
  while $||w_i^{(k+1)} - w_i^{(k)}||_2 > \varepsilon$ **do**
    $w_i^{(k+1)} \leftarrow \mathbf{X}g_i(w_i^{(k),\top}\mathbf{X})^\top/T - (g_i'(w_i^{(k),\top}\mathbf{X})\mathbb{1}_T/T)w_i^{(k)}$ `// Newton update`
    $w_i^{(k+1)} \leftarrow w_i^{(k+1)} - \sum_{j=1}^{i-1}(w_i^{(k+1),\top}w_j)w_j$ `// online Gram-Schmidt procedure`
    $w_i^{(k+1)} \leftarrow w_i^{(k+1)}/||w_i^{(k+1)}||_2$ `// Normalisation`
    $k \leftarrow k + 1$
  end
  $w_i \leftarrow w_i^{(k)}$
end
return $W = [w_1, \dots, w_m]^\top$

6.5 Non Linear ICA using Variational Autoencoders

The statistical model is described by (11), where $\theta \in \Theta$ represents the vector of interest. In the context of variational inference, we compute an approximation of the posterior $p_\theta(\mathbf{s} \mid \mathbf{x})$ denoted by $q_\phi(\mathbf{s} \mid \mathbf{x})$ and maximise a proxy for the log-likelihood, called Evidence Lower Bound (ELBO) to compute an estimator of θ. In the next section, we state the main result on the identifiability of the model given in Khemakhem et al. [2020].

Identifiability results

We denote by $p_\theta(\mathbf{s})$ our usual prior on $\mathbf{s}$. The mixing process f is considered as a parameter to be learned by the VAE and is included in θ.

We assume that the prior on $\mathbf{s}$ is conditionally factorial. Each component s_i has a univariate exponential family distribution given some observed conditioning variable $\mathbf{u}$. More precisely, each source s_i given $\mathbf{u}$ is distributed as an exponential distribution with sufficient statistics $T_{i,1}, \dots, T_{i,k}$, i.e.,

$$p_{T,\lambda}(\mathbf{s} \mid \mathbf{u}) = \prod_{i=1}^{m} \frac{Q_i(s_i)}{Z_i(\mathbf{u})} \exp\left[\sum_{j=1}^{k} T_{i,j}(s_i)\lambda_{i,j}(\mathbf{u})\right], \tag{20}$$

where

1. the Q_i's are the base measures,
2. the $Z_i(\mathbf{u})$ are normalizing constants,

3. the $T_i = (T_{i,1}, \ldots, T_{i,k})$ are the sufficient statistics,

4. $\lambda_i(\mathbf{u}) = (\lambda_{i,1}(\mathbf{u}), \ldots, \lambda_{i,k}(\mathbf{u}))$ are the natural parameters, crucially depending on u,

5. the number of statistics k is fixed.

We note that this prior on $\mathbf{s}$ given $\mathbf{u}$ is not restrictive, as demonstrated in Sriperumbudur et al. [2017]. A broad class of densities can be approximated by infinite-dimensional exponential densities.

Now, we aim to learn the parameter of interest $\theta = (f, T, \lambda)$. The idea is to introduce theorems based on our previous hypothesis to obtain identifiability up to classes of equivalence and conclude about the convergence of the VAE.

The identifiability results will be defined up to an equivalence class; we can retrieve signals up to an invertible matrix. Two parameters $\theta = (f, T, \lambda)$, $\theta' = (f', T', \lambda')$ are said to be equivalent if

$$(f, T, \lambda) \sim (\tilde{f}, \tilde{T}, \tilde{\lambda}) \iff \exists A, c \text{ s.t. } T(f^{-1}(\mathbf{x})) = A\tilde{T}(\tilde{f}^{-1}(\mathbf{x})) + c, \forall \mathbf{x} \in \mathbb{R}^n, \tag{21}$$

where A is an $mk \times mk$ invertible matrix.

Now, we can introduce the main contribution of Khemakhem et al. [2020]:

Theorem 6.1 (Identifiability of deep generative model up to equivalence)**.** *Assume that we observe data sampled from the generative model defined by (13) and (20), with parameters $\theta = (f, T, \lambda)$. Assume the following holds:*

1. *The set $\varphi_{\boldsymbol{\varepsilon}}^{-1}(\{0\})$ has measure zero, where $\varphi_{\boldsymbol{\varepsilon}}$ is the characteristic function of the noise density $p_{\boldsymbol{\varepsilon}}$.*

2. *The mixing function f is injective.*

3. *The sufficient statistics $T_{i,j}$ in (20) are differentiable almost everywhere, and $(T_{i,j})_{1 \leq j \leq k}$ are linearly independent.*

4. *There exist $mk + 1$ distinct points $\mathbf{u}_0, \ldots, \mathbf{u}_{mk}$ such that the matrix*

$$\begin{bmatrix} \lambda(\mathbf{u}_1) - \lambda(\mathbf{u}_0)^\top \\ \vdots \\ \lambda(\mathbf{u}_{mk}) - \lambda(\mathbf{u}_0)^\top \end{bmatrix},$$

is invertible.

Then, the parameters (f, T, λ) are $\sim$-identifiable up to a invertible linear transformation.

This theorem and its proof can be found in Khemakhem et al. [2020]. Let us now explain the motivation for each hypothesis:

1. We introduce the first hypothesis to facilitate the inference process. To transform the noise setting into a noiseless one, we employ the convolutional trick. For this purpose, identifying the distribution of the noise is crucial, and the satisfaction of assumption (1) is necessary to achieve this transformation.

2. The second assumption is imperative for concluding the full-rank properties of Jacobians.

3. The regularity assumption (3) is needed for the same reason as above: to ensure that the Jacobian of $T \circ f$ are full rank.

4. The fourth assumption is the core of the theorem. If not satisfied, it implies that for

$$\mathbf{h}(\mathbf{u}) = \begin{bmatrix} \lambda_{1,1}(\mathbf{u}) - \lambda_{1,1}(\mathbf{u}_0) \\ \vdots \\ \lambda_{m,k}(\mathbf{u}) - \lambda_{m,k}(\mathbf{u}_0), \end{bmatrix} \tag{22}$$

the vectors $\mathbf{h}(\mathbf{u}_1), \ldots, \mathbf{h}(\mathbf{u}_{mk})$ are linearly dependent in $\mathbb{R}^{mk}$, i.e., $\mathbf{h}(\mathcal{U})$ (where $\mathcal{U}$ is the support of the distribution of $\mathbf{u}$) has dimension at most $mk-1$ and hence measure zero. If the hypothesis is satisfied, this implies that as long as the $\lambda_{i,j}(\mathbf{u})$ are generated randomly and independently, then almost surely, $\mathbf{h}(\mathcal{U})$ won't be included in any such subset with measure zero. Consequently, the invertibility assumption holds.

For inference of the mixing process up to a permutation and sign signature, see Khemakhem et al. [2020, Th. 2, 3]. The variational inference procedure is done using VAEs. In the next section, we briefly introduce them.

VAEs framework

Variational Autoencoders (VAEs) are a class of generative models designed for learning latent representations of data in a probabilistic framework. They operate on the principle of encoding observed data into a low-dimensional latent space, followed by the generation of new samples from this latent space. In the context of a VAE, u represents additional conditioning information, which could include attributes, labels, or any relevant features associated with the input data. The goal is to train the VAE to generate data that not only captures the inherent variability but also whose distribution depends upon $\mathbf{u}$. The training process involves optimizing the model parameters to maximize the Evidence Lower Bound (ELBO), a key objective that balances the reconstruction of input data and adherence to the latent space distribution.

1. **Encoder Network:** The encoder network, denoted by $q_\phi(\mathbf{s} \mid \mathbf{x}, \mathbf{u})$, takes input data $\mathbf{x}$ and conditioning information $\mathbf{u}$ and outputs the parameters of the approximate posterior distribution $q_\phi(\mathbf{s} \mid \mathbf{x}, \mathbf{u})$, typically parameterised by ϕ.

2. **Decoder Network:** The decoder network, denoted by $p_\theta(\mathbf{x} \mid \mathbf{s}, \mathbf{u})$, takes a sample $\mathbf{s}$ from the latent space and conditioning information $\mathbf{u}$ to generate a reconstruction of the input data $\mathbf{x}$, typically parameterised by θ.

3. **Latent Space:** The latent variable $\mathbf{s}$ is sampled from the approximate posterior $q_\phi(\mathbf{s} \mid \mathbf{x}, \mathbf{u})$ during the encoding phase.

4. **Loss Function (Variational Lower Bound):** Instead of maximising the log-likelihood which is intractable in the general case, we instead resort to the evidence lower bound (ELBO) denoted by $\mathcal{L}(\theta, \phi)$, which is the negative of the KL divergence between the approximate posterior $q_\phi(\mathbf{s} \mid \mathbf{x}, \mathbf{u})$ and the true posterior $p(\mathbf{s} \mid \mathbf{x}, \mathbf{u})$ plus the log-likelihood of the data:

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{s}|\mathbf{x},\mathbf{u})}\left[\log \frac{p_\theta(\mathbf{x} \mid \mathbf{s}, \mathbf{u})p(\mathbf{s})}{q_\phi(\mathbf{s} \mid \mathbf{x}, \mathbf{u})}\right] = \mathbb{E}_{\mathbf{s}\sim q_\phi(\cdot|\mathbf{x},\mathbf{u})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{s}, \mathbf{u})\right] - \underbrace{D_{\mathrm{KL}}\left(q_\phi(\mathbf{s} \mid \mathbf{x}, \mathbf{u}) \,||\, p_\theta(\mathbf{s})\right)}_{\geq 0} \tag{23}$$

   Intuitively, these two terms illustrate the dual objective we are trying to satisfy:

   - **KL Divergence:** The KL divergence term encourages the approximate posterior to be close to the true posterior.
   - **Log-likelihood term:** Represents the expected log-likelihood of the observed data under $q_\phi(\mathbf{s} \mid \mathbf{x}, \mathbf{u})$.

   Rigorously, the ELBO is a lower bound of the log-likelihood by Jensen's inequality:

$$\log p_\theta(\mathbf{x}) = \log\left(\mathbb{E}_{\mathbf{s}\sim q_\phi(\cdot|\mathbf{x},\mathbf{u})}\left[\frac{p_\theta(\mathbf{x}, \mathbf{s})}{q_\phi(\mathbf{s} \mid \mathbf{x}, \mathbf{u})}\right]\right) \geq \mathbb{E}_{q_\phi(\mathbf{s}|\mathbf{x},\mathbf{u})}\left[\log\left(\frac{p_\theta(\mathbf{x} \mid \mathbf{s}, \mathbf{u})p(\mathbf{s})}{q_\phi(\mathbf{s} \mid \mathbf{x}, \mathbf{u})}\right)\right] = \mathcal{L}(\phi, \theta). \tag{24}$$

6.6 Experiments results

6.6.1 Synthetic Univariate Signals

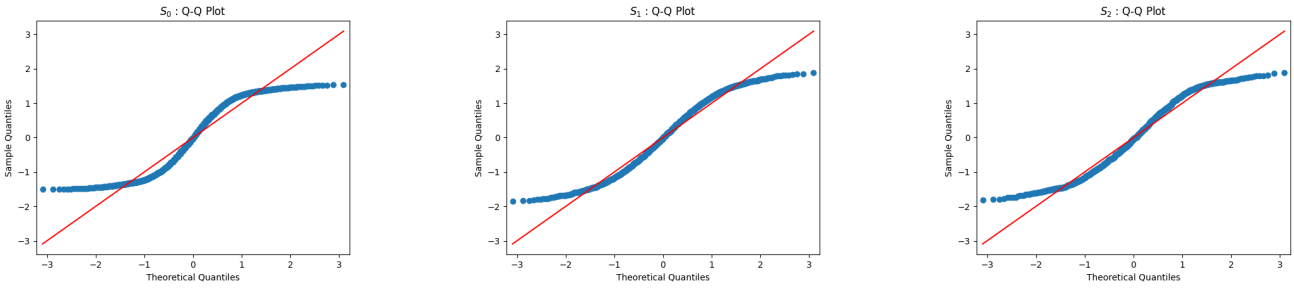


Figure 6: QQ-plots of the synthetic source signals.

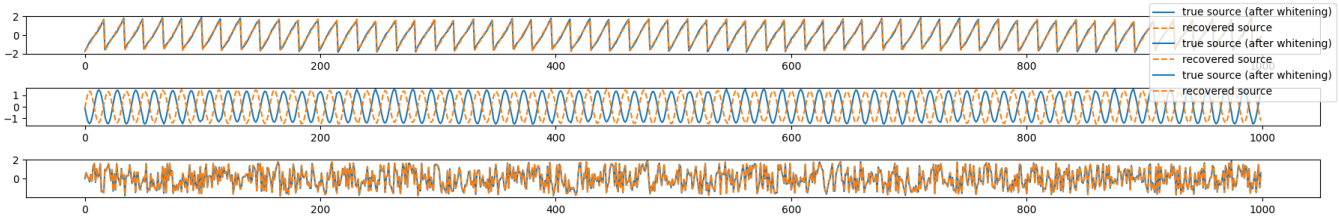
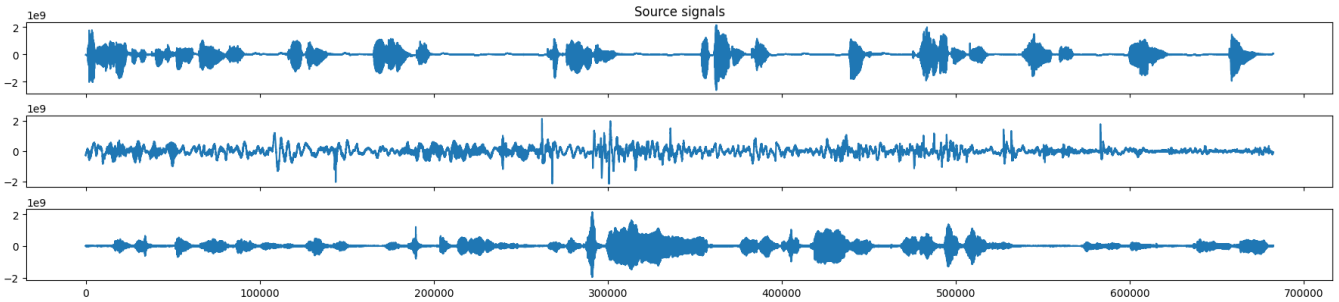


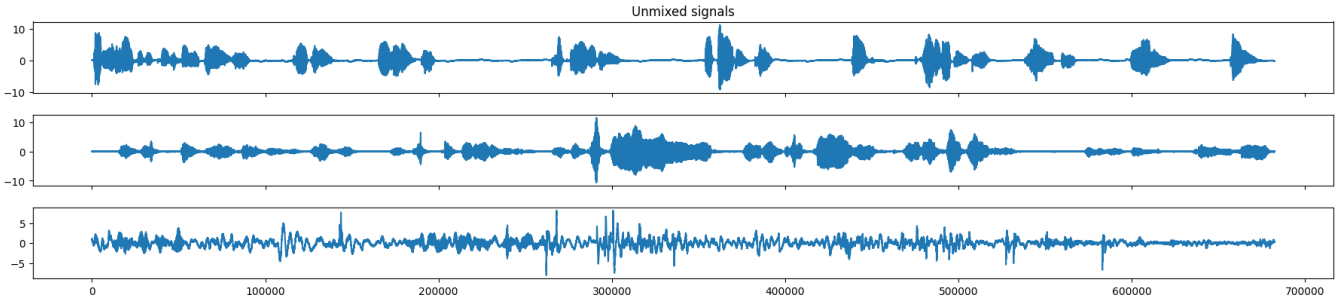
Figure 7: Recovered sources (in orange) superposed to their best matching original source after whitening (in blue) using the Gradient Descent Algorithm 1.

6.6.2 Audio Sources Seperation

When sources have the same scaling magnitude (Homogeneous scaling):



(a) Scaled source signals.



(b) Recovered signals.

Figure 8: Blind source separation when original sources have been scaled before mixing.

7 Individual contributions

	ZB	YLF	NB
Theory	• Lecture and recap of Hyvärinen and Oja [2000] and Khemakhem et al. [2020] - review of the proofs.	• Derived the general gradient descent rule (17).	• Derived FastICA through Newton optimisation method (relying on Murphy [2012] textbook) and wrote it in pseudo-code (2).
Report	• Wrote the MLE section 2.1, reviewed [Bell and Sejnowski, 1995] on the GD approach to the MLE. Wrote the associated appendix - explanation and intuition behind the sub/super Gaussian priors. • Wrote the VAE section 3, and the appendix on variational autoencoders 6.5, reviewed [Ben-Israel, 1999, Sriperumbudur et al., 2017]).	• Wrote the abstract, the introduction and the explicit algorithms 1 and 2 (as well as the beautiful plot 5).	• Wrote about the FastICA method (2.2 and 6.4) and its variants (2.3) • Wrote the Experiments section (4), including plots and figures it contains, and the associated appendix.
Implementation	• Wrote a first version of GD algorithm. • Attempts at feature extraction for both the linear and non-linear ICA.	• Implemented in JAX, the FastICA algorithm in JAX, the FLAX version of iVAE and with Zineb, the GD algorithm.	• Contributed to the project source code with an accelerated version of the FastICA algorithm (in Numpy) and a greedy version of the Mean Correlation Coefficient Score (in numpy). • Tested the implemented algorithms on synthetic and real univariate signals.

The implementation and experimental evaluation is due to all three members, in particular, FastICA algorithm is due to YLF, the gradient-based algorithm for the MLE is due to ZB and YLF. The VAE implementation is due to YLF. The toy model on which all algorithms were tested was written by NB. The Abstract and Motivation sections were written by YLF, the linear ICA section was written by both NB and ZB with inputs from YLF, the VAE section and the supplementary section associated with was written by ZB, and the supplementary material section was mainly written by NB and ZB, after which all authors reviewed the manuscript. See the above table.

8 What is next? Remarks, difficulties and questions.

During the whole project, we faced different challenges. Here is a list with the difficulties encountered within each approach:

1. Linear ICA:
    - Providing a more rigourous explanation for the choice of the prior functions, the g_i's. Empirically, assigning the priors using the Kurtosis-based rule seems sufficient. However, can different priors be used to further upgrade the quality of the retrieved sources?
    - We tried to use Linear ICA as a method for extraction of features on real data. To do so, we used the Labelled Face in the Wild dataset provided by sklearn, we noticed that our whitening function is not as efficient as the sklearn implementation. In fact, our whitening implementation is not designed for edge cases, such as ill-conditioned matrices. Despite first attempts to correct it (use a threshold, discard certain values...) we were not able to obtain unit variance signals on both train and test datasets. To improve our project, we would like to inspire from sklearn techniques of whitening that rely on singular value decomposition instead of eigenvalues decomposition.
2. VAE ICA:
    - Reproduce the results given by the authors: obtain 95% of MCC on the toy data. We run the Adam update step with the loss computed as the mean over the different batches' loss, while the original implementation update the optimiser state for each batch. Why?
    - Find relevant use cases: classic VAE already enable to denoise data without the use of an additional observed variable $\mathbf{u}$.
    - Use this method as an extraction of relevant features for dimension reduction. Can we have a latent variable $\mathbf{u}$ that doesn't introduce any leak for further work.

References

A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995. URL https://api.semanticscholar.org/CorpusID:1701422. 12

A. Ben-Israel. The change-of-variables formula using matrix volume. *SIAM Journal on Matrix Analysis and Applications*, 21(1):300–312, 1999. doi: 10.1137/S0895479895296896. URL https://doi.org/10.1137/S0895479895296896. 7, 12

E. C. Cherry. Some Experiments on the Recognition of Speech, with One and with Two Ears. *The Journal of the Acoustical Society of America*, 25(5):975–979, 06 1953. ISSN 0001-4966. doi: 10.1121/1.1907229. URL https://doi.org/10.1121/1.1907229. 1

A. Hyvärinen and E. Oja. Independent component analysis: algorithms and applications. *Neural Networks*, 13 (4):411–430, 2000. ISSN 0893-6080. doi: https://doi.org/10.1016/S0893-6080(00)00026-5. URL https://www.sciencedirect.com/science/article/pii/S0893608000000265. 1, 2, 3, 8, 12

I. Khemakhem, D. Kingma, R. Monti, and A. Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In S. Chiappa and R. Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2207–2217. PMLR, 26–28 Aug 2020. URL https://proceedings.mlr.press/v108/khemakhem20a.html. 1, 3, 4, 6, 8, 9, 12

K. Murphy. *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine Learning series. MIT Press, 2012. ISBN 9780262018029. URL https://books.google.fr/books?id=NZP6AQAAQBAJ. 8, 12

B. Sriperumbudur, K. Fukumizu, A. Gretton, A. Hyvärinen, and R. Kumar. Density estimation in infinite dimensional exponential families, 2017. 9, 12