Independent Component Analysis

PGM Project

Nathan de Lara
École polytechnique

Florian Tilquin
ENS Cachan

Vincent Vidal
ENS Ulm

January 13, 2016

Contents

Abstract

This paper is dedicated to the study of Independent Component Analysis. We intent to implement, apply and compare several algorithms while being presenting some theoretical aspects such as the link between the likelihood maximisation and the mutual information.

1 Problem statement

1.1 Introduction

The general Independent Component Analysis problem can be formalised this way: Suppose we have some random variables $x \in \mathbb{R}^p$ which correspond to a mix of some primitive sources $s \in \mathbb{R}^n$. The aim is to extract from x every source s_i. Evidently, we will only be able to recover s up to a scaling factor and a permutation of the component. To do so, we will suppose here that:

- the sources are independents.
- the mix is linear and instantaneous
- at most one source has a Gaussian distribution.

We formally define:

$$x = As \quad \text{and} \quad y = Wx, \tag{1}$$

where A is the mixing matrix, W the separation matrix and y the estimation of the sources. The goal is then to find a matrix W that maximise a certain measure of independence of y.

As a measure of independence, we consider, for theoretical purpose, the mutual information, defined in equation 3. However, as it is too hard to compute, we consider other contrast functions, invariant by permutation, scaling on coordinates and maximal for independent ones.

1.2 Information Theory

Let $X \in \mathbb{R}^n$ be a random variable, we note $P(X)$ his density and Σ_X his covariance matrix. In the following, we will write only P when it's possible. Besides we will note $\mathrm{H}(X)$ the entropy of X, defined as $\mathbb{E}\big[-\log P(X)\big]$.

In the space of measures, let $\mathcal{G}$ be the manifold of Gaussian distributions, $\mathcal{P}$ the manifold of "product" distributions and $\mathcal{P} \wedge \mathcal{G}$ the manifold of Gaussian "product" distributions. Note that these manifolds are exponential families. In this space, we can define the **Kullback–Leibler divergence** from Q to P :

$$\mathrm{K}(P \parallel Q) = \int_{\mathbb{R}^n} P(x) \log \frac{P(x)}{Q(x)} \mathrm{d}x. \tag{2}$$

The main advantage of this geometric point of view is that the Kullback-Leibler divergence allows the notion of projection on exponential families. The projection of P on the family $\mathcal{E}$, noted $P^{\mathcal{E}}$, is defined as the vector of $\mathcal{E}$ that minimise the divergence to P. This projection verifies the Pythagorean theorem and thus we will be able to find relation between the main quantities defined with this divergence. Schematised in the figure 1, the main quantities are:

– The **mutual information**:

$$\begin{aligned} \mathrm{I}(Y) &= \mathrm{K}\Big(P(Y) \parallel \Pi_i P_i(Y_i)\Big) = \mathrm{K}\Big(P(Y) \parallel P(Y)^{\mathcal{P}}\Big) \\ &= \sum_i \mathrm{H}\big(P(Y_i)\big) - \mathrm{H}\big(P(Y)\big) \end{aligned} \tag{3}$$

– The **non-gaussianity**:

$$\mathrm{G}(Y) = \mathrm{K}\Big(Y \parallel \mathcal{N}\left(\mathbb{E}\big[Y\big], \Sigma_Y\right)\Big) = \mathrm{K}\Big(P(Y) \parallel P(Y)^{\mathcal{G}}\Big). \tag{4}$$

– The **correlation**:

$$\begin{aligned} \mathrm{C}(Y) &= \mathrm{K}\Big(\mathcal{N}\left(\mathbb{E}\big[Y\big], \Sigma_Y\right) \parallel \mathcal{N}\left(\mathbb{E}\big[Y\big], \operatorname{Diag}\Sigma_Y\right)\Big) \\ &= \mathrm{K}\Big(P(Y)^{\mathcal{G}} \parallel P(Y)^{\mathcal{P}\wedge\mathcal{G}}\Big) \\ &= \frac{1}{2}\log\frac{\det\left(\operatorname{Diag}(\Sigma_Y)\right)}{\det\left(\Sigma_Y\right)}. \end{aligned} \tag{5}$$

Using the Pythagorean theorem and the two decompositions of $\mathrm{K}\big(P \parallel P^{\mathcal{P}\wedge\mathcal{G}}\big)$, through $P^{\mathcal{P}}$ or $P^{\mathcal{G}}$, shown in the Figure 1, we can prove that:

$$\mathrm{I}(Y) + \sum_i \mathrm{G}(Y_i) = \mathrm{G}(Y) + \mathrm{C}(Y). \tag{6}$$

Because the non-gaussianity is invariant under invertible affine transforms, minimising the mutual independence according to W is equivalent to minimise $\mathrm{C}(Y) - \sum_i \mathrm{G}(Y_i)$. We can then define a set of contrast function, for $\alpha \geq 0$:

$$\phi_\alpha(Y) = \alpha\mathrm{C}(Y) - \sum_i \mathrm{G}(Y_i). \tag{7}$$

Let's remark that the FastICA algorithm is based on the minimisation of the marginal non-gaussianity (so $\alpha = 0$). For more information see [Car03].

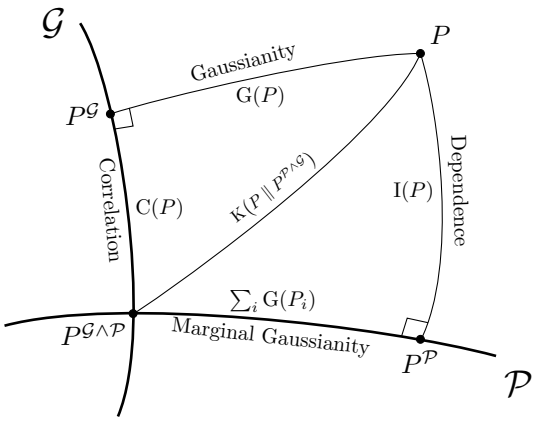

Figure 1: Representation of a distribution P and the different projections on the exponential families $\mathcal{P}$ and $\mathcal{G}$. On the paths between the distributions are the quantities associated to the Kullback-Leibler divergence between those distributions.

1.3 ICA and Maximum Likelihood

As presented in [HO00], it is possible to consider ICA as a maximum likelihood problem linked to the infomax principle. With the previously introduced notations, the log-likelihood is defined as:

$$L = \sum_{t=1}^{T} \sum_{i} \log f_i \left(w_i^{\mathsf{T}} x(t) \right) + T. \log \left(|det(W)| \right), \tag{8}$$

where f_i is the density function of s_i. Then, if we suppose f_i be the actual distribution of $y_i(t) = w_i^{\mathsf{T}} x(t)$, the expectation of this likelihood can be written :

$$\begin{aligned} \mathbb{E}\big[\, L \,\big] &= \log \left| \det W \right| + \sum_i \mathbb{E}\big[\log f_i \left(w_i^{\mathsf{T}} x(t) \right) \big] \\ &= \mathrm{H}(WX) - \mathrm{H}(X) - \sum_i \mathbb{E}\big[-\log P\left(y_i(t) \right) \big] , \end{aligned} \tag{9}$$

which is, up to a constant $\mathrm{H}(X)$, the Mutual Independence given equation 3.

1.4 Performance measure

Because we can only recover the matrix W up to scaling factors and a permutation of the component, we can't norms to evaluate the separation matrix. We use here the "Amari divergence" [ACY+96], written in equation 10, which gives a criterion of proximity between two matrices.

If U and V are two n-by-n matrices, the Amari error is defined by:

$$d(U,V) = \frac{1}{2n} \sum_i \left(\frac{\sum_j |a_{ij}|}{\max\limits_j |a_{ij}|} - 1 \right) + \frac{1}{2n} \sum_j \left(\frac{\sum_i |a_{ij}|}{\max\limits_i |a_{ij}|} - 1 \right), \tag{10}$$

with $a_{ij} = (UV^{-1})_{ij}$. This function, which is not an actual distance, has the advantage to have the invariant wanted: invariant by scaling factors and permutations matrix multiplication.

2 Algorithms for ICA

2.1 Hérault and Jutten (HJ) algorithm

This method is one of the first algorithm to perform ICA. It is based on the neural network principle. We write $W = \left(I_n + \widetilde{W}\right)^{-1}$ and for a pair of given functions (f, g), we adapt $\widetilde{W}$ as follows:

$$\widetilde{W}_{ij} = f(y_i)g(y_j). \tag{11}$$

See [JH91] for more information.

2.2 Jade algorithm

Several methods are based on the cumulants. The goal here is to annul all the cross cumulants of order 4, which would assure a certain "independence" up to the order 4. Thus, the idea is to diagonalize the cumulant tensor which is equivalent to minimise the following contrast function:

$$c\left(x\right) = \sum_{i,k,l} \left|\mathrm{Cum}\left(x_i, x_i^*, x_k, x_l\right)\right|^2. \tag{12}$$

See [Car89] for more information about the JADE algorithm.

2.3 FastICA algorithm

This algorithm is based on the maximisation of the marginal non-gaussianity, which is approximated here as follow:

$$\mathrm{G}(Y_i) \simeq C^{te} \times \left(\mathbb{E}\big[\, f(Y_i)\,\big] - \mathbb{E}\big[\, f(\mathcal{N}\left(0,\, 1\right))\,\big]\right)^2, \tag{13}$$

with C^{te} a positive constant and f a non linear quadratic function. In our experiments, we used $f(x) = \frac{x^4}{4}$. But it is possible to use $f(x) = \log\cosh x$ or $f(x) = \exp\left(-\frac{x^2}{2}\right)$ as well. The problem is now an optimisation problem consisting of maximising the expression in equation 13 according to W under the constraint of non correlation of $y = Wx$. Using a Fixed-point algorithm, we could derive the following iteration step:

$$\widetilde{W}_{t+1} = \mathbb{E}\big[\, X.f(W_t^{\mathsf{T}}X)^{\mathsf{T}}\,\big] - \mathbb{E}\big[\, f''(W_t^{\mathsf{T}}X)\,\big]\, W_t, \tag{14}$$

with W_t the normalise vector of $\widetilde{W}_t$. See [Hyv99] for more information.

2.4 Kernel ICA algorithm

Given a reproducing kernel Hilbert space $\mathcal{F}$, this algorithm seeks to minimize the Kernel Generalized Variance defined as:

$$\widehat{\delta}_{\mathcal{F}} = -\frac{1}{2}\log\prod_i (1-\rho_i^2) \tag{15}$$

where the ρ_i are the kernel canonical correlations between the observations components, obtained with computations over the observations Gram matrices.

3 Results

3.1 Experimental design for simulated data

In order to test the different algorithms, we use a pipeline from [BJ03]. We took some distributions, whose density are shown figure 2. We sampled N times a certain amount m

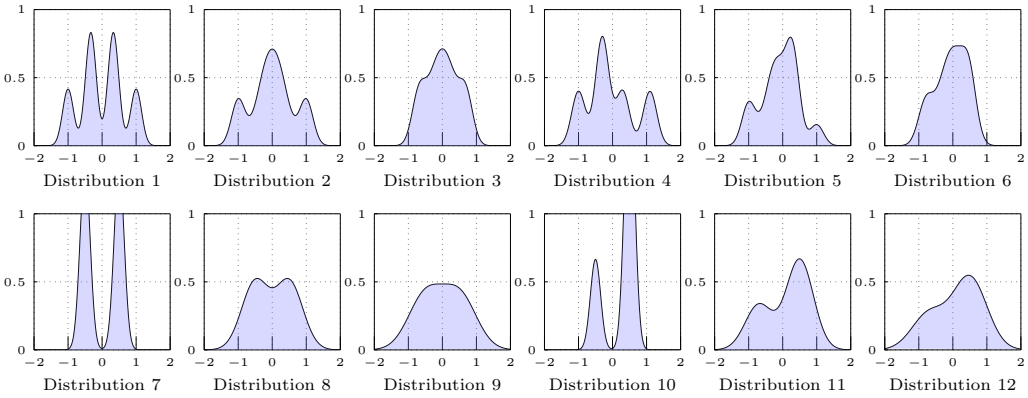

Figure 2: Distributions used to test the algorithms.

of them to create the initial signal s. Note that we may choose a given distribution multiple times. Then, we picked a random bounded matrix A, used to mix the signals. Then, we preprocess the mixed signal Y, by multiply it by a whitening matrix P.

At this point, we applied the ICA algorithms to the signal $Y = PAs$, which gave us the separation matrix W. Eventually we evaluated the performance of the algorithm by taking the "Amari divergence" between W and the real separation matrix $W_0 = (PA)^{-1}$.

Note that the whitening matrix P correspond to the inverse of the square root of the covariance matrix of $X = As$.

3.2 Results on experimental data

The results are showed in the table 1.

For the left table, we chose two sources following the same distribution. The proximity of the distribution 8, 9 and 12 to the gaussian, prohibited for more than one source, explain the overall bad results obtained for this distributions.

For the right table, we varied the number of distribution m and the number of samples N, with some randomly selected distributions (one distribution is potentially selected multiple times). We can see here the influence of computational cost. The Kernel ICA and the HJ algorithm couldn't manage to achieve results as the number of sources grows. Let's remark that the poor results of the HJ algorithm on 4 sources can be explain by the non convergence of this algorithm.

Overall, kernel ICA and HJ algorithm perform better than the two other, but need a lot of computation time.

3.3 Results on real data

We took 4 images and mixed them with a random bounded matrix. The 4 mixed images were given to the ICA algorithm and the 4 separate images were normalise and display. All the images for the JADE algorithm are showed in the figure 3.

The results are quite good, even if we can still see some artefacts of the other images. Let's remark that, as expected, the recovered images are permuted compared to the initial ones and that 2 of them are in negative mode.

We applied the different algorithm to a mix of sound signal, as the problem of ICA is often use as a solution for the cocktail party problem. The overall performance were good and the small artefact that was left were covered by the small noise of the initial sounds.

	JADE	HJ	FastICA	Kernel ICA
1	5.67	6.37	6.51	**3.35**
2	9.47	8.15	11.54	**5.11**
3	6.76	**6.74**	8.39	10.55
4	6.21	6.82	7.66	**3.32**
5	5.99	6.42	7.30	**2.93**
6	9.32	**9.25**	12.18	9.37
7	2.90	**2.28**	3.68	2.74
8	8.52	**8.38**	10.31	12.53
9	17.52	**11.65**	21.95	28.21
10	13.94	10.30	17.71	**2.95**
11	9.57	8.98	12.31	**6.57**
12	19.19	**12.33**	22.93	13.76

m	N	JADE	HJ	FastICA	Kernel ICA
2	250	8.35	7.56	10.45	**6.06**
	1000	3.66	3.39	4.42	**2.38**
4	1000	11.88	58.52	12.20	**9.72**
	4000	5.33	83.07	5.80	**3.86**
8	2000	19.75	X	19.40	**19.15**
	4000	13.06	X	13.29	**9.71**
16	4000	31.81	X	**28.92**	X
	8000	**20.56**	X	27.89	X

Table 1: **Left:** Average Amari divergence re-scaled by 100 obtained with the listed algorithms for random mix $m = 2$ sources of size $N = 250$ sampled with twelve different distributions. **Right:** Same measure for m sources of size N whose distributions are randomly selected among the twelve. The best results are in bold font. An X is put when a standard desktop computer could not compute the result.

Figure 3: Application of JADE algorithm to images separation. The first line presents the original sources, the second one the mix and the last one the estimations.

!! ATTENTION !!
Il manque les références de : [BS95], [Car97], [Com94] et [LeB].
!! ATTENTION !!

References

[ACY+96] Shun-ichi Amari, Andrzej Cichocki, Howard Hua Yang, et al. A new learning algorithm for blind signal separation. *Advances in neural information processing systems*, pages 757–763, 1996.

[BJ03] Francis R Bach and Michael I Jordan. Kernel independent component analysis. *The Journal of Machine Learning Research*, 3:1–48, 2003.

[BS95] Anthony J Bell and Terrence J Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995.

[Car89] Jean-Francois Cardoso. Source separation using higher order moments. In *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*, pages 2109–2112. IEEE, 1989.

[Car97] Jean-Francois Cardoso. Infomax and maximum likelihood for blind source separation. 1997.

[Car03] Jean-François Cardoso. Dependence, correlation and gaussianity in independent component analysis. *The Journal of Machine Learning Research*, 4:1177–1203, 2003.

[Com94] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[HO00] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4):411–430, 2000.

[Hyv99] Aapo Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.

[JH91] Christian Jutten and Jeanny Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal processing*, 24(1):1–10, 1991.

[LeB] Hervé LeBorgne. Analyse en composantes indépendantes. chapter 3.