Independant Componant Analysis

Nathan de Lara, Florian Tilquin, Vincent Vidal

Master Mathématiques, Vision et Apprentissage

Problem statement

Let $x \in \mathbb{R}^p$ be some random variables whose components correspond to different mix of some primitive sources $s_i \in \mathbb{R}^n$. The aim is to retrieve an estimation y of every source s_i, given only x. We note A the mixing matrix and W the separation matrix such that:

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

In order to retrieve the sources, we suppose that:

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

Measure of independence

We want to find W that maximises the independence of $y = Wx$.
The information theory provides a measure of independence based on the Kullback-Leibler divergence:

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

With $\mathcal{G}$ the Gaussian distribution manifold and $\mathcal{P}$ the product one, we define

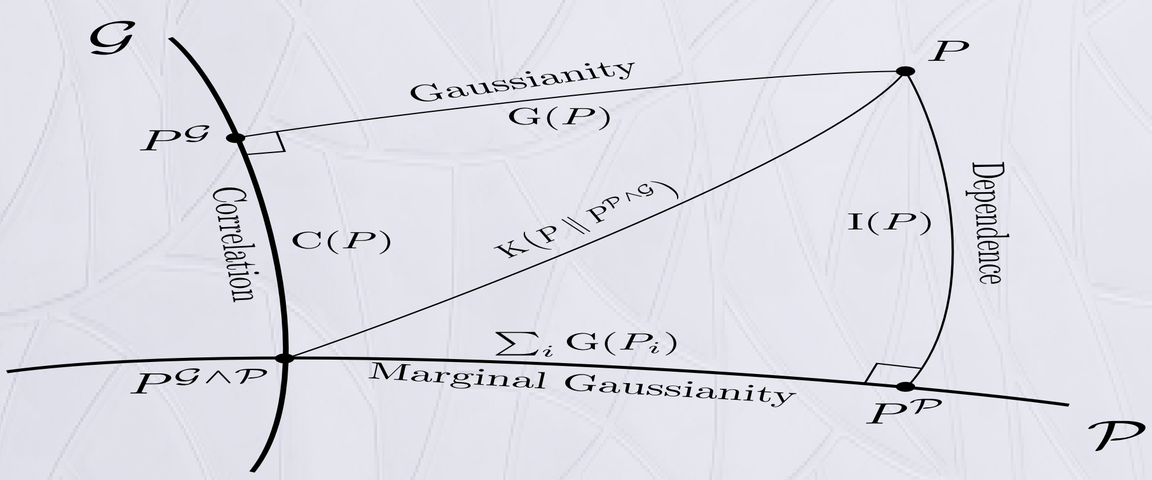

Figure: Representation of the distribution and the different projections on the manifolds $\mathcal{P}$ and $\mathcal{G}$ and the different quantities that can be defined with the Kullback Leibler divergence.

The Pythagorean theorem implies:

$$\mathrm{I}(Y) + \sum_i \mathrm{G}(Y_i) = \mathrm{G}(Y) + \mathrm{C}(Y)\,. \tag{3}$$

If the mutual information $\mathrm{I}(P)$ appears to be the best one to use, it is too hard to compute. The equation 3 justifies the use of the non-gaussianity, correlation or even negentropy as contrast functions.

Performance evaluation

The "Amari distance", equation 4, gives a criterion of proximity between two matrices, to evaluate the performance of an algorithm. If U and V are two n-by-n matrices, the Amari distance is defined by:

$$d(U,V) = \frac{1}{2n}\sum_{i=1}^{n}\left(\frac{\sum_{j=1}^{n}|a_{ij}|}{\max_j |a_{ij}|} - 1\right) + \frac{1}{2n}\sum_{j=1}^{n}\left(\frac{\sum_{i=1}^{n}|a_{ij}|}{\max_i |a_{ij}|} - 1\right) \tag{4}$$

with $a_{ij} = (UV^{-1})_{ij}$. This function, which is not an actual distance, has the advantage to be invariant by scaling factors and permutation of the components of the matrices.

Algorithms

Several algorithms have been developed to perform ICA, among those we can cite:

- HJ: one of the first algorithms for ICA by Hérault and Jutten who pioneered the *blind source separation problem* in the 1980s. It is inspired from the neural network principle.
- JADE: for Joint Approximate Diagonalization of Eigenmatrices. It belongs to the family of the cumulants algorithms initially introduced by Comon in the early 1990s and has a complexity of $\mathcal{O}(n^4)$.
- FastICA: proposed by Hyvärinen in the late 1990s, it uses non-gaussianity as an approximation of independence and performs approximate Newton iterations.
- KGV: for Kernel Generalized Variance. Introduced in the early 2000s by Bach and Jordan, this method outperforms most of the previous algorithms and is particularly resistant to outliers. However, it is more computationally expensive.

Hérault and Jutten (HJ) algorithm

This method is based on the neural network principle. Writing $W = \left(I_n + \widetilde{W}\right)^{-1}$, for a pair of given functions $(f,\ g)$, the algorithm estimates:

$$\widetilde{W}_{ij} = f(y_i)g(y_j). \tag{5}$$

JADE algorithm

Several methods are based on the cumulants. The aim in JADE is to annul all the cross cumulants of order 4. The cumulant tensor is diagonalized which is equivalent to minimizing the following contrast function:

$$c\,(x) = \sum_{i,k,l} \left|\mathrm{Cum}\,(x_i, x_i^*, x_k, x_l)\right|^2. \tag{6}$$

FastICA algorithm

The FastICA algorithm is based on the information theory. Non-gaussianity is used a proxy for independence. Given a quadratic function f, the algorithm performs:

$$\widetilde{W}_{t+1} = \mathbb{E}\left[\, X.f(W_t^{\mathsf{T}}X)^{\mathsf{T}}\,\right] - \mathbb{E}\left[\, f''(W_t^{\mathsf{T}}X)\,\right] W_t, \tag{7}$$

where W_t is the normalized vector of $\widetilde{W}_t$. For experimentation, we use $f(x) = \frac{x^4}{4}$.

Kernel Generalized Variance algorithm

Given a RKHS $\mathcal{F}$, this algorithm seeks to ??minimize/maximize?? the Kernel Generalized Variance defined as:

$$\widehat{\delta}_{\mathcal{F}} = \prod_i (1 - \rho_i^2) \tag{8}$$

where the ρ_i are the canonical kernel correlations between the observations components.

Results

	JADE	HJ	FastICA	KGV
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
rand	8.35	7.56	10.45	6.06

Figure: This table presents the average Amari distance obtained with the listed algorithms for random mix two sources of size 250 sampled with twelve different distributions. Finally, the last line presents the same measure for sources whose distribution is randomly selected among the twelve.

Figure: Application of !!!algo!!! to images separations. The first line presents the original sources, the second one the mix and the last one the estimations.