Radical Independent Component Analysis (ICA)

Michael J. Meyer

July 1, 2016

1 Introduction

Let X be a random vector in $\mathbb{R}^d$ with components X_j and finite second moments. X is called *white* if the covariance matrix $cov(X) = I$ is the identity matrix of dimension d. There is always a matrix W such that WX is white and the passage from X to WX is called *whitening*. The same terminology applies if instead of X a finite *sample* of realizations of X is considered.

The whitening matrix W is determined only up to multiplication with an orthogonal matrix Q (since $cov(QX) = cov(X)$ if Q is orthogonal).

Upon whitening the components of X are uncorrelated but in general far from independent. In general only if X is Gaussian are the components already independent once they are uncorrelated.

ICA algorithms first whiten the data (typically with the whitening matrix W computed from the eigenvalue decomposition of the covariance matrix $cov(X)$) and then search for a rotation Q of the whitened data $Y = WX$ such that the components of $Z = QY$ are independent, or rather, as independent as possible.

The measure of independence of the components Z_j employed is called the *contrast function* and is typically (and in this project) the Kullback-Leibler distance $dist_{KL}(P, P^*)$ of the distribution $P = P_Z$ of Z from the product

$$P^* = P_{Z_1} \otimes P_{Z_2} \otimes \cdots \otimes P_{Z_d} \tag{1}$$

of the marginal distributions P_{Z_j}. The two distributions are equal if and only if the components Z_j are independent.

The Kullback-Leibler distance is a very strong notion of distance (stronger than the total variation distance). For discrete distributions P, R it has the following information theoretical intepretation:

Suppose a source emits symbols s randomly from a finite alphabet with distribution P. You do not know the true distribution P and construct an encoding $C = C(s)$ of the symbols s (each symbol mapped to a string of zeros and ones) which is optimal (has shortest expected code length) if the symbols have distribution R.

Then the expected code length $E^P(length(C(s))$ under the true distribution P generating the symbols will be $dist_{KL}(P, R)$ bits longer than an optimal code

constructed for the distribution P of the symbols s. The unit is *bits* if the binary logarithm $\log_2$ is used in the computation of the Kullback-Leibler distance (and called *nats*, if the natural logarithm is used).
With P^* as in (1) this contrast function assumes a particularly simple form

$$dist_{KL}(P, P^*) = c + \sum\nolimits_j H(Z_j) \tag{2}$$

where the constant c does not depend on the rotation Q (in $Z = QY$) and $H(Z_j)$ denotes the entropy of the coordinate Z_j, see [1], p1, eq(5). On the right hand side only one dimensional distributions enter the computation and so we only need estimators for the entropy of one dimensional distributions.

ICA algorithms minimize this contrast function as a function of the rotation Q. Here Q varies over all rotations, a complicated high dimensional manifold. Gradient search methods are liable to get stuck at a local minimum.

The *Radical ICA* algorithm, see [1] notes that each rotation Q can be written as a product of *Jacobi rotations* $J(i, j, \phi)$:
the counterclockwise rotation in the coordinate plane spanned by the standard unit vectors e_j and e_j (coordinate planes in coordinates i, j). For a proof see Remark (A) below. For our purposes the angles ϕ can be restricted to an intervall of length $\pi/2$ and we use the interval $\Phi = [-\pi/4, \pi/4]$

For each pair (i, j) with $0 \leq i < j < d$, a complete search is then performed over an equidistant grid of angles $\phi \in \Phi$ for the angle ϕ such that the Jacobi rotation $J(i, j, \phi)$ leads to the greatest decrease in the contrast function (2).

This Jacobi rotation is then added to the overall rotation Q of the whitened data Y. Several sweeps through the coordinate pairs (i, j) are performed until the contrast function no longer decreases at least a fixed percentage of its value.

Thus this algorithm is a kind of exhaustive search and so naturally slow. The original article [1] suggests the use of the Vasicek entropy estimator modified using spacings. This uses the order statistic and forces us to order the samples of the components Z_j which is of complexity $O(n \log(n))$ in the sample size n. Moreover the algorithm is $O(d^2)$ in the data dimension d and so $O(d^2 n \log(n))$ alltogether.

The entropy estimator however is exchangeable (in fact a parameter to the ICA class constructor in this implementation). We have also provided a naive empirical entropy estimator (using the histogram step function density). With this the rotation search becomes $O(d^2 n)$ and a large speedup is obtained already for sample size 30000 and data dimension 6. The speed gain increases rapidly with increasing data dimension.

The Vasicek entropy estimator is also provided. In all tests performed the two entropy estimators perform equally well but the test coverage is not very thorough: a sample size of 30000 (padded or unpadded) is assumed. Generally it is recommended to pad the data to sample size at least 10000 (convolve with a centered Gaussian distribution by adding independent Gaussian perturbations repeatedly to each data item), see [1] for a detailed rationale for this.
Implementations in R and Matlab can be found at [2]

Remarks (A) Let Q be a rotation matrix, i.e. orthogonal with determinant one. The proof of the QR decomposition using Jacobi rotations (also referred to as Givens rotations, see [3] or [4]) applied to the transpose Q' yields Jacobi rotations J_i such that the product

$$J_k J_{k-1} \dots J_1 Q' = R \tag{3}$$

is an upper triangular matrix R which is also orthogonal since so is the left hand side of (3). It is then easy to see that R is a diagonal matrix $R = diag(d_i)$ with diagonal elements $d_i = \pm 1$. Since the left hand side has determinant one (it is a rotation) so does R and it follows that the number of negative signs on the diagonal of R is *even*. The negative signs can thus be arranged in pairs (d_i, d_j) in coordinates i, j. Now multiplication with the Jacobi rotation $J(i, j, \pi)$ has the following effect on the diagonal elements of R

$$d_i \to -d_i,\ d_j \to -d_j, \quad \text{and} \quad d_k \to d_k,\ \forall k \neq i, j.$$

Thus we can find further Jacobi rotations $J_{k+1}, \dots, J_m$ such that the product

$$J_m J_{m-1} \dots J_1 Q' = I$$

is the identity matrix. Multiplication with Q then yields $Q = J_m J_{m-1} \dots J_1$, as desired.

(B) Why do we want to do such a thing as ICA? Assume you can really find an invertible transformation $Z = F(X)$ (with easily computed inverse) such that the components Z_j of Z are independent. Then you can construct the multidimensional distribution of Z in a trivial fashion from the one dimensional marginals Z_j. But that means that we can construct the distribution of X itself from the one dimensional distributions of the Z_j.

To see this consider the problem of sampling from the distribution of X: sample independently from the distribution of the Z_j, obtain a sample of Z as $Z = (Z_1, \dots, Z_d)$ and apply the inverse transformation $X = F^{-1}(Z)$ to obtain a sample of Z.

Clearly we cannot easily reduce a general multidimensional distribution to one dimensional ones. It will take complex nonlinear transformations to obtain true coordinate independence and no easy algorithms will be found to do this in general.

(C) It would be highly desirable to compute the Kullback-Leibler distance (2) of the solution from the product of the marginal distributions since this would give us an indication how close we are to true independence. Unfortunately this is not possible. We cannot compute the constant c in (2) since it contains the entropy of the multidimensional distribution of Y, see [1], p1, eq(5).

Our contrast function drops the constant c since it does not depend on the rotation Q and so is irrelevant for the minimization problem.

(D) Since the components of a Gaussian random variable are independent as soon as they are uncorrelated, the rotation Q to true coordinate independence

cannot be uniquely determined if at least two coordinates Z_i, Z_j of Z are Gaussian. In this case we can multiply Q with arbitrary Jacobi rotations $J(i, j, \phi)$, which preserve the covariance matrix $cov(Z_i, Z_j)$, and maintain independence of Z_i, Z_j (the other coordinates are not affected by such a rotation).

In other words: the rotation Q is unique only if at most one component Z_j is Gaussian. For the practical application of the algorithm this is not very relevant since a contrast function minimizing rotation is always found.

(E) The Vasicek estimator is better than a naive empirical estimator based on a histogram with equal width bins if the values of the random sample are unevenly distributed (densely clustered). Note that the naive empirical distribution derived from the histogram (step function density) is a mixture of uniform distributions in the intervals on which the bins are based but the sample values need not be uniform in these intervals at all. In that case one cannot expect a histogram to yield a good representation of the density of the distribution.

In the Vasicek estimator this problem is less severe since the clustering is taken into account naturally by considering the gaps in the ordered sample. For definiteness consider a sample of size n from the distribution

$$P = \frac{1}{2}Unif(0,1) + \frac{1}{2}Unif(I) \quad \text{where } I = \left[0, 1/n^2\right].$$

with density $f = \frac{1}{2}\left(1 + n^2 1_I\right)$.

Since practically no draws from the $Unif(0,1)$ distribution fall into the intervall I, the Vasicek estimate of the entropy of the mixture is close to the average of the estimates for the two distributions which is close to

$$\frac{1}{2}\left[Entropy(Unif(0,1)) + Entropy(Unif(I))\right]$$

which in turn is close to the true entropy as can easily be verified by direct computation.

On the other hand the histogram with bins of equal width $1/\sqrt{n}$ sees half the sample falling into the first bin $J = [0, 1/\sqrt{n})$ and the other half uniformly in $[0, 1]$ so it believes the distribution to be the mixture $\frac{1}{2}Unif(0,1) + \frac{1}{2}Unif(J)$ and estimates the entropy accordingly. This entropy diverges from the true entropy in unbounded fashion, as $n \uparrow \infty$.

However it is a problem is that sorting to obtain the ordered sample is very expensive. A solution is to apply a bin sort that is fast for uniformly distributed values and slow only when the values are highly clustered in which case there are no fast solutions at all.

References

[1] Erik G. Miller, John W. Fisher III
ICA USING SPACINGS ESTIMATES OF ENTROPY,
`https://people.cs.umass.edu/~elm/papers/learned-miller03a.pdf`,

[2] Prof. E.G. Learned-Miller, Homepage,
https://people.cs.umass.edu/~elm/papers_by_code.html

[3] Unknown lecturer, Oxford university, QR decomposition,
https://www0.maths.ox.ac.uk/system/files/coursematerial/2015/3135/110/lecture4_slides.pdf

[4] Kahan, UC Berkley, Reflection and Rotation
http://people.eecs.berkeley.edu/~wkahan/Math128/ReflRotn.pdf