A FAMILY OF FIXED-POINT ALGORITHMS FOR INDEPENDENT COMPONENT ANALYSIS

Aapo Hyvärinen

Helsinki University of Technology
Laboratory of Computer and Information Science
Rakentajanaukio 2 C, FIN-02150 Espoo, Finland
aapo.hyvarinen@hut.fi

ABSTRACT

Independent Component Analysis (ICA) is a statistical signal processing technique whose main applications are blind source separation, blind deconvolution, and feature extraction. Estimation of ICA is usually performed by optimizing a 'contrast' function based on higher-order cumulants. In this paper, it is shown how almost any error function can be used to construct a contrast function to perform the ICA estimation. In particular, this means that one can use contrast functions that are robust against outliers. As a practical method for finding the relevant extrema of such contrast functions, a fixed-point iteration scheme is then introduced. The resulting algorithms are quite simple and converge fast and reliably. These algorithms also enable estimation of the independent components one-by-one, using a simple deflation scheme.

1. INTRODUCTION

Independent Component Analysis (ICA) [1, 2] is a statistical signal processing technique whose goal is to express a set of random variables as linear combinations of statistically independent component variables. Some applications of ICA are blind source separation [1], feature extraction [3], and, in a slightly modified form, blind deconvolution [4]. In the simplest form of ICA [2], we observe m scalar random variables $x_1, x_2, ..., x_m$ which are assumed to be linear combinations of n unknown independent components, or ICs, denoted by $s_1, s_2, ..., s_n$. These ICs s_i are assumed to be mutually *statistically independent*, and zero-mean. Let us arrange the observed variables x_j into a vector $\mathbf{x} = (x_1, x_2, ..., x_m)^T$ and the IC variables s_i into a vector $\mathbf{s}$, respectively; then the linear relationship is given by

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1}$$

Here, $\mathbf{A}$ is an unknown $m \times n$ matrix of full column rank, called the mixing matrix. The basic problem of ICA is then to estimate both the mixing matrix $\mathbf{A}$ and the realizations of the ICs s_i *using only observations of the mixtures* x_j. Two fundamental restrictions of the model are that, firstly, we can only estimate non-Gaussian ICs (except if just one of the ICs is Gaussian), and secondly, we must have at least as many observed linear mixtures as ICs, i.e. $m \geq n$. The assumption of zero mean of the ICs is in fact no restriction, as this can always be accomplished by subtracting the mean from the random vector $\mathbf{x}$. Moreover, the ICs and the columns of $\mathbf{A}$ can only be estimated up to a multiplicative constant, because any constant multiplying an IC in eq. (1) could be cancelled by dividing the corresponding column of the mixing matrix $\mathbf{A}$ by the same constant. For mathematical convenience, one usually defines that the ICs s_i have unit variance. This makes the (non-Gaussian) ICs unique, up to their signs [2]. Note that this definition of ICA implies no ordering of the ICs.

The current algorithms for Independent Component Analysis can be roughly divided into two categories. The algorithms in the first category, e.g. [2, 5], rely on batch computations minimizing or maximizing so-called contrast functions based on higher-order cumulants. The problem with these algorithms is that they require very complex matrix or tensorial operations. The second category contains adaptive algorithms often based on stochastic gradient methods, which may have implementations in neural networks, e.g. [4, 6, 7, 8]. The main problem with this category is slow convergence, and the fact that convergence depends crucially on the correct choice of the step size (learning rate) parameters. Furthermore, most of the proposed ICA algorithms in both categories are highly non-robust against outliers.

In this paper we introduce a large family of novel algorithms for ICA estimation. First we introduce a general family of 'contrast' functions whose extrema are closely connected to the estimation of ICs. Then we show how the fixed-point algorithm in [4, 9] can be generalized for finding the relevant extrema of such contrast functions.

Our 'generalized fixed-point' algorithms have a number of desirable properties. First, they are easy to use, as they contain no user-defined parameters and require no prewhitening of the data. Second, their convergence is fast, and can be proven analytically. Third, for a suitable choice of the contrast function, the generalized fixed-point algorithm is much more robust against outliers than the great majority of ICA algorithms, which are often based on estimation of fourth-order moments.

2. FIXED-POINT ALGORITHM USING KURTOSIS

In this section, we introduce the principle of fixed-point iteration using the classical contrast function, kurtosis. The introduction of generalized contrast functions is postponed to the following section. The algorithm presented in this section is especially easy to analyze mathematically. For practical purposes, however, the generalization of this algorithm, to be introduced in the following sections, is much better.

2.1. Kurtosis as a contrast function

Kurtosis, or the fourth-order cumulant [4], is defined for a zero-mean random variable v as $\text{kurt}(v) = E\{v^4\} - 3(E\{v^2\})^2$. Kurtosis is a contrast function for ICA in the following sense. Consider a linear combination of the observed mixtures $\mathbf{x}$, say $\mathbf{w}^T\mathbf{x}$, where the vector $\mathbf{w}$ is constrained so that $E\{(\mathbf{w}^T\mathbf{x})^2\} = 1$. When $\mathbf{w}^T\mathbf{x} = \pm s_i$ for some i, i.e. when the linear combination equals, up to the sign, one of the ICs, the kurtosis of $\mathbf{w}^T\mathbf{x}$ is locally minimized or maximized [4, 8]. This property is widely used in ICA algorithms, and forms also the basis of the fixed-point algorithm presented in this section.

2.2. Derivation of a fixed-point algorihm

Now we derive a fixed-point algorithm to find the relevant extrema of kurtosis. This is a modification for non-whitened data of the algorithm presented in [4, 9]. First note that the gradient of the kurtosis of $\mathbf{w}^T\mathbf{x}$ with respect to $\mathbf{w}$ is

$$\nabla_{\mathbf{w}} \text{kurt}(\mathbf{w}^T\mathbf{x}) = 4[E\{\mathbf{x}(\mathbf{w}^T\mathbf{x})^3\} - 3\mathbf{C}\mathbf{w}E\{(\mathbf{w}^T\mathbf{x})^2\}] \quad (2)$$

where $\mathbf{C} = E\{\mathbf{x}\mathbf{x}^T\}$ is the covariance of the data. Let us now optimize this kurtosis under the constraint $E\{(\mathbf{w}^T\mathbf{x})^2\} = \mathbf{w}^T\mathbf{C}\mathbf{w} = 1$. By the classical conditions of Kuhn-Tucker, we have in the extrema:

$$2\lambda\mathbf{C}\mathbf{w} = 4[E\{\mathbf{x}(\mathbf{w}^T\mathbf{x})^3\} - 3\mathbf{C}\mathbf{w}E\{(\mathbf{w}^T\mathbf{x})^2\}] \quad (3)$$

$$\Leftrightarrow$$

$$\mathbf{w} = \frac{2}{\lambda}[\mathbf{C}^{-1}E\{\mathbf{x}(\mathbf{w}^T\mathbf{x})^3\} - 3\mathbf{w}] \quad (4)$$

where λ is the Lagrangian coefficient. This clearly suggests a fixed-point algorithm in which the vector $\mathbf{w}(k-1)$ is updated in the k-th step as follows:

$$\mathbf{w}^*(k) = \mathbf{C}^{-1}E\{\mathbf{x}(\mathbf{w}(k-1)^T\mathbf{x})^3\} - 3\mathbf{w}(k-1) \quad (5)$$
$$\mathbf{w}(k) = \mathbf{w}^*(k)/\sqrt{\mathbf{w}^*(k)^T\mathbf{C}\mathbf{w}^*(k)}$$

where the expectation is estimated using a sufficiently large sample of data. Though the conditions of Kuhn-Tucker only give a necessary condition of optimality, and only certain extrema of kurtosis provide estimation of the ICs, this algorithm does converge globally to one of the right extrema, finding always one of the ICs as the linear combination $\mathbf{w}^T\mathbf{x}$. This is proven in [9, 10].

Note that in (5) it is implicitly assumed that the covariance matrix $\mathbf{C}$ is not singular. If this is not the case, the dimension of the data must be reduced, e.g., with PCA, before running the algorithm.

2.3. Estimating several ICs

The algorithm in (5) estimates just one of the ICs. To estimate several ICs, we need to do the fixed-point iteration step (5) using several vectors $\mathbf{w}_1(k-1), ..., \mathbf{w}_N(k-1)$, and decorrelate the corresponding linear combinations $\mathbf{w}_1(k)^T\mathbf{x}, ..., \mathbf{w}_N(k)^T\mathbf{x}$ at every iteration. Methods for accomplishing this will be presented in Section 4 in connection with the generalized fixed-point algorithm.

3. GENERAL CONTRAST FUNCTIONS

Above, we used kurtosis, the classical contrast function, for estimating the ICs. In this section, we show how almost any non-quadratic function can be used to derive a 'local' contrast function for ICA. This is very important for many practical applications, because the fourth power inherent in kurtosis grows very fast and is thus quite sensitive to outliers. Using contrast functions that grow slower than the fourth power, we can find algorithms that are more robust against outliers.

To begin with, note that one intuitive interpretation of contrast functions is that they are measures of non-normality [2]. Therefore, to obtain a contrast function based on an arbitrary error function G, it is natural to consider the difference of the expectation of G for actual data from what it would be for a Gaussian variable. In other words, we can define a contrast function J that measures the non-normality of a zero-mean random variable x using any even, sufficiently regular (non-quadratic) function G as follows

$$J_G(x) = E_x\{G(x)\} - E_\nu\{G(\sigma_x \nu)\} \quad (6)$$

where ν is a standardized Gaussian variable, $\sigma_x = \sqrt{E\{x^2\}}$, and thus $\sigma_x\nu$ is a Gaussian variable of the same variance as x.

Clearly, J_G can be considered a generalization of kurtosis. For $G(x) = x^4$, J_G becomes simply the kurtosis of x. Note that G must not be quadratic, because then J_G would be trivially zero for all distributions.

Thus, it seems plausible that J_G in (6) could be a contrast function, in the sense discussed in subsection 2.1. The fact that J_G is indeed a contrast function locally, i.e. near a solution, is proven in [10]. The only condition is that the IC s_i in question must fulfill $E\{s_i g(s_i) - g'(s_i)\} \neq 0$, which is reminiscent of the classical condition of non-zero kurtosis [2]. We also conjecture that for a 'reasonable' choice of G, J_G is a global contrast function. Moreover, it seems reasonable to presume that then also J_G is a global contrast function, for most distributions of the ICs, in a more strict sense: the maxima of $|J_G(\mathbf{w}^T\mathbf{x})|$ are obtained when $\mathbf{w}^T\mathbf{x}$ equals one of the independent components. Numerical simulations seem to confirm these conjectures. For $G(t) = \ln\cosh(t)$ (whose derivative is the tanh function), some simulation results are depicted in Fig. 1.

4. THE GENERALIZED FIXED-POINT ALGORITHMS

The actual search for the extrema of a general contrast function J_G as in (6) may then be performed in many different ways, e.g. by (stochastic) gradient descent, as was done with kurtosis in, e.g. [4, 8]. This adaptive, neural-like approach is developed elsewhere. However, as was explained in Section 2, a highly efficient, reliable and simple algorithm for finding the relevant extrema can be obtained using the

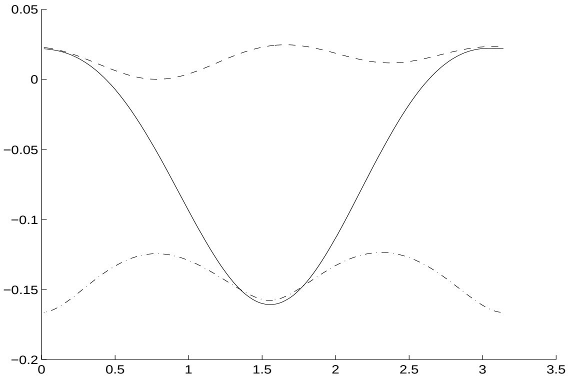

Figure 1: A numerical example illustrating that J_G may be a contrast function in a more global and strict sense. Here $G(x) = \ln\cosh(x)$. In the 2-D case, J_G was plotted as a function of the angle between $\mathbf{w}$ and one of the columns of $\mathbf{A}$, which was here orthogonal. Three sets of ICs were used. Continuous line: one super-Gaussian and one sub-Gaussian IC. Dash-dot: Two super-Gaussian ICs. Dashed line: Two sub-Gaussian ICs. The maxima of the absolute value of J_G were always obtained when the angle was 0 or $\pi/2$, which were exactly the directions of the ICs.

fixed-point method. Thus we obtain an important generalization of the fixed-point algorithm presented in [4, 9].

4.1. Estimating one of the ICs

To derive the generalized fixed-point algorithm, first note that when $E\{(\mathbf{w}^T\mathbf{x})^2\} = \mathbf{wCw} = 1$, the gradient of $J_G(\mathbf{w}^T\mathbf{x})$ equals [10]

$$\nabla_{\mathbf{w}} J_G(\mathbf{w}^T\mathbf{x}) = E_{\mathbf{x}}\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - E_\nu\{g'(\nu)\mathbf{Cw}\}, \qquad (7)$$

where $g = G'$ is the derivative of G, and g' is the derivative of g. Now we can use the same logic as in Section 2. Thus we use the Kuhn-Tucker conditions to obtain the following *generalized fixed-point algorithm*:

$$\begin{aligned} &\mathbf{w}^*(k) = \\ &\mathbf{C}^{-1}E\{\mathbf{x}g(\mathbf{w}(k-1)^T\mathbf{x})\} - E\{g'(\mathbf{w}(k-1)^T\mathbf{x})\}\mathbf{w}(k-1) \\ &\mathbf{w}(k) = \mathbf{w}^*(k)/\sqrt{\mathbf{w}^*(k)^T\mathbf{C}\mathbf{w}^*(k)} \end{aligned} \qquad (8)$$

where g can thus be chosen to be any odd, sufficiently regular, non-linear function, and the two expectations are, in practice, estimated using the average of a large sample of $\mathbf{x}$. Note that ν in the latter expectation has been replaced by $\mathbf{w}^T\mathbf{x}$ to enhance the convergence of the algorithm; this modification does not change the validity of the Kuhn-Tucker conditions. It is proven in [10] that using the algorithm in (8), $\mathbf{w}(k)$ converges, up to the sign, to one of the rows of the inverse of the mixing matrix $\mathbf{A}$. This enables estimation of one of the ICs as $\mathbf{s} = \mathbf{A}^{-1}\mathbf{x}$. (Note that the non-singularity of $\mathbf{C}$ also implies that $\mathbf{A}$ can be assumed to be an invertible square matrix.) The only condition of convergence is that $E\{s_i g(s_i) - g'(s_i)\} \neq 0$ for all s_i that we want to estimate. This can be considered a generalization of the condition, valid when kurtosis is used as contrast, that kurtosis of the ICs must be non-zero [2, 8, 4]. We prove in [10] analytically only the *local* convergence of the algorithms, i.e. convergence for initial points near a solution. Our simulations, however, indicate that if g is a 'nice' function in some intuitive sense (smooth, does not have many local extrema), the algorithms do converge globally, i.e. starting from any (random) initial point $\mathbf{w}(0)$.

The convergence of (8) is shown in [10] to be cubic, and experiments show that usually less than 10 iterations is enough. This means that these algorithms are very fast. They are also very reliable, because no parameters need to be tuned for good convergence.

4.2. Choice of Non-Linearity

As a concrete choice of the non-linearity g in (8), we propose the following functions that give algorithms that are *robust against outliers*:

$$g_1(u) = \tanh(u), \quad g_1'(u) = \cosh^{-2}(u) \qquad (9)$$

$$g_2(u) = u\exp(-u^2/2), \quad g_2'(u) = (1-u^2)\exp(-u^2/2) \quad (10)$$

The robustness is due to the fact that these functions do not give large values for arguments far from 0. In fact, g_2 becomes 0 for large arguments, and thus provides an extremely robust algorithm. However, to benefit from the exceptional robustness offered by g_2 requires that the estimation of the covariance matrix $\mathbf{C}$ is also done in a highly robust way, which is out of the scope of our paper. Thus, g_1 is already as robust as one can get using ordinary estimation of the covariance matrix.

4.3. Estimating several ICs

The algorithm in (8) estimates just one of the ICs. To estimate several ICs, we need to run the algorithm (8) using several vectors $\mathbf{w}_1, ..., \mathbf{w}_N$. To prevent different vectors from converging to the same extrema, we must *decorrelate* the linear combinations $\mathbf{w}_1^T\mathbf{x}, ..., \mathbf{w}_N^T\mathbf{x}$ after every iteration of (8). A simple way of achieving this is a deflation scheme based on a Gram-Schmidt-like decorrelation. This means that we estimate the ICs one by one. When we have estimated p ICs, or p vectors $\mathbf{w}_1, ..., \mathbf{w}_p$, we run (8) for $\mathbf{w}_{p+1}$, and after every iteration step subtract from $\mathbf{w}_{p+1}(k)$ the 'projections' of the previously estimated p vectors, and then renormalize $\mathbf{w}_{p+1}(k)$:

1. Let $\mathbf{w}_{p+1}(k) = \mathbf{w}_{p+1}(k) - \sum_{j=1}^{p} \mathbf{w}_{p+1}(k)^T\mathbf{C}\mathbf{w}_j\mathbf{w}_j$
2. Let $\mathbf{w}_{p+1}(k) = \mathbf{w}_{p+1}(k)/\sqrt{\mathbf{w}_{p+1}(k)^T\mathbf{C}\mathbf{w}_{p+1}(k)}$

(11)

In certain applications, however, it may be desired to use a symmetric decorrelation, in which no vectors are 'privileged' over others [11]. This can be accomplished, e.g., by the classical methods involving matrix square roots, or by the following simple iterative algorithm, where $\mathbf{W}(k)$ is the matrix $(\mathbf{w}_1(k), ..., \mathbf{w}_N(k))$ of the vectors:

1. Let $\mathbf{W}(k) = \mathbf{W}(k)/\sqrt{\|\mathbf{W}(k)^T\mathbf{C}\mathbf{W}(k)\|}$
Repeat 2. until convergence: (12)
2. Let $\mathbf{W}(k) = \frac{3}{2}\mathbf{W}(k) - \frac{1}{2}\mathbf{W}(k)\mathbf{W}(k)^T\mathbf{C}\mathbf{W}(k)$

The norm in step 1 can be almost any ordinary matrix norm, e.g., the largest absolute row (or column) sum (but not the Frobenius norm). Note that the normalization in the fixed-point step can be omitted when using (12). The convergence of the orthonormalization method in (12) is proven in [10].

5. SIMULATION RESULTS

To demonstrate the convergence of our algorithms, and especially their robustness, we applied our algorithm to blind separation of four artificially generated source signals in the presence of some disturbing outliers. Two of the signals were super-Gaussian, and two were sub-Gaussian. These source signals were mixed using several random square matrices, whose elements were drawn from a standardized Gaussian distribution, so as to obtain different mixed signals. To test the robustness of our algorithms, four outliers whose values were ±10 were added in random locations in the mixtures. Then we used our generalized fixed-point algorithm to estimate the original signals. Three different non-linearities were used: the cubic function as in (5), and the two non-linearities in (9) and (10). Moreover, two different estimators of the covariance matrix were used: the conventional estimator using the sample average of $\mathbf{x}\mathbf{x}^T$, and a theoretical robust estimator, which was simulated by estimating the covariance matrix without the outliers. In all the runs, the following were observed:

- 10 iterations were always enough for convergence
- No convergence to non-desired point was observed, i.e. the convergence was always global.
- Estimates based on kurtosis (i.e. the cubic non-linearity) were essentially worse than the others
- Estimates using (10) were slightly better than those using (9).
- The two preceding effects were much stronger if the covariance matrix was estimated in a robust way. In fact then the estimations using (10) were practically perfect in spite of the added outliers.

For details, see [10].

6. CONCLUSIONS

We introduced a generalized version of the fixed-point algorithms presented in [4, 9] for ICA estimation. These generalized fixed-point algorithms have a number of desirable properties:

- They contain no parameters that need to be defined by the user.
- They are very simple to program.
- They require no prewhitening of the data.
- Their fast, cubic convergence can be proven analytically.
- For some choices of the non-linearity, for example those in (9) and (10), they are much more robust against outliers than conventional ICA algorithms.

Some applications of ICA using the generalized fixed-point algorithm, or the original fixed-point algorithm in [4, 9], are described in [11].

7. REFERENCES

[1] C. Jutten and J. Herault, "Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, pp. 1–10, 1991.

[2] P. Comon, "Independent component analysis – a new concept?," *Signal Processing*, vol. 36, pp. 287–314, 1994.

[3] A. Bell and T. J. Sejnowski, "Edges are the independent components of natural scenes," in *NIPS*96*, (Denver, Colorado), 1996.

[4] A. Hyvärinen and E. Oja, "One-unit learning rules for independent component analysis," in *NIPS*96*, (Denver, Colorado), 1996.

[5] J.-F. Cardoso, "Iterative techniques for blind source separation using only fourth-order cumulants," in *Proc. EUSIPCO*, pp. 739–742, 1992.

[6] A. Bell and T. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[7] A. Cichocki, S. I. Amari, and R. Thawonmas, "Blind signal extraction using self-adaptive non-linear hebbian learning rule," in *Proc. NOLTA'96*, pp. 377–380, 1996.

[8] N. Delfosse and P. Loubaton, "Adaptive blind separation of independent sources: a deflation approach," *Signal Processing*, vol. 45, pp. 59–83, 1995.

[9] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," Tech. Rep. A35, Helsinki University of Technology, Laboratory of Computer and Information Science, 1996. Submitted to a journal.

[10] A. Hyvärinen, "A family of fixed-point algorithms for independent component analysis," Tech. Rep. A40, Helsinki University of Technology, Laboratory of Computer and Information Science, 1996. Can be found at http://nucleus.hut.fi/~aapo.

[11] J. Karhunen, A. Hyvärinen, R. Vigario, J. Hurri, and E. Oja, "Applications of neural blind separation to signal and image processing," in *Proc. ICASSP'97*, (Munich, Germany), 1997.