FastISA: A fast fixed-point algorithm for Independent Subspace Analysis

Aapo Hyvärinen and Urs Köster*
Helsinki Institute for Information Technology, Department of Computer Science
University of Helsinki, Finland

Abstract. Independent Subspace Analysis (ISA; Hyvarinen & Hoyer, 2000) is an extension of ICA. In ISA, the components are divided into subspaces and components in different subspaces are assumed independent, whereas components in the same subspace have dependencies.In this paper we describe a fixed-point algorithm for ISA estimation, formulated in analogy to FastICA. In particular we give a proof of the quadratic convergence of the algorithm, and present simulations that confirm the fast convergence, but also show that the method is prone to convergence to local minima.

1 Introduction

Independent Component Analysis (ICA) has successfully been used in the past on a variety of data, but because it is a linear model and it requires independent sources underlying the data, its range is limited. This motivates an extension of ICA, in which certain dependencies between sources can also be modeled. ISA is such an extension, where the inclusion of a pooling stage and a nonlinear transformation augments the linear filtering. The pooling organizes filters into subspaces inside which dependencies are allowed. The estimation is very similar to ICA, as it follows the assumption that the subspaces are mutually independent. It can be performed by maximizing a nonlinear contrast function with gradient descent, which is however quite slow and inefficient. This motivates a fixed-point algorithm for ISA, which we present in this paper. Like the FastICA algorithm [1], the method combines quick convergence with simplicity and usability. Here we discuss some of the mathematical background of the ISA framework, followed by a convergence proof for the algorithm and simulations showing the convergence properties.

2 Model and algorithm

ICA[2] is a method for separating a multivariate signal $\mathbf{x}$ into statistically independent components $\mathbf{s}$. This can be formulated as $\mathbf{x} = \mathbf{A}\mathbf{s}$ where $\mathbf{A}$ is a mixing matrix. Inverting the system to $\mathbf{s} = \mathbf{W}\mathbf{x}$, we can identify $\mathbf{W}$ as the demixing matrix we wish to form, such that the independence of the sources is maximized. For ISA, we introduce ”independent feature subspaces”. We do not require independence of individual sources, but instead between norms of projections on these subspaces. Thus the model can be estimated by maximizing the independence of these norms. We define one such element as

$$u_i = (\sum_{j \in S_i} s_j^2)^{1/2} = (\sum_{j \in S_i} (\mathbf{w}_j^T \mathbf{x})^2)^{1/2} \tag{1}$$

[*] Urs Köster is supported by a scholarship from the Alfried Krupp von Bohlen und Halbach-Foundation.

i.e. we project onto the group S_i of elements which belong to the i-th subspace and compute the norm. Note that taking the norm is a nonlinear mapping, which makes the method capable of modeling complicated dependency structures that linear ICA cannot capture. For the estimation of the model we need to maximize the independence of these norms. To do this, we define the probability distribution of the model as:

$$\log p(s_1,...,s_m) = \sum_{j=1}^{m} \left(-\log Z_j - \frac{G(\sum_{i\in S_j} s_i^2)}{b} \right) \tag{2}$$

where the square root has been replaced by the more general nonlinear contrast function $G(.)$. Z normalizes the distribution and b ensures unit variance, they can be computed in closed form for some choices of $G(.)$ [3]. For our simulations we used the function $G(x) = \sqrt{x+\gamma}$, where γ is a small, arbitrary constant to aid with stability, and was chosen to be 0.1.

Here, we propose the following new algorithm for the estimation of ISA. To estimate the components $\mathbf{s}$, we iteratively update the rows $\mathbf{w}$ of the demixing matrix, which correspond to the feature vectors, with the update rule, which is formulated in analogy to fastICA:

$$\mathbf{w}_j^+ = E\left\{ \mathbf{x}(\mathbf{w}_j^T\mathbf{x})g(\sum_{i\in S_j}(\mathbf{w}_i^T\mathbf{x})^2)\right\} - E\left\{ g(\sum_{i\in S_j}(\mathbf{w}_i^T\mathbf{x})^2) + 2(\mathbf{w}_j^T\mathbf{x})^2 g'(\sum_{i\in S_j}(\mathbf{w}_i^T\mathbf{x})^2)\right\}\mathbf{w}_j \tag{3}$$

where $E\{.\}$ denotes the expectation value, S_j is the set of indices of components belonging to the subspace, $g(.)$ and $g'(.)$ are the first and second derivatives of the nonlinearity $G(.)$. The algorithm requires the data to be whitened. We orthogonalize $\mathbf{W}$ after each step, which is equal to decorrelation since we are in whitened space.

3 Convergence proof

To show the convergence of our new algorithm, we make a change of variables to $\mathbf{z} = \mathbf{A}^T\mathbf{w}$, so the update rule, here given for the k^{th} element of the first vector, becomes:

$$z_1^{k+} = E\left\{ s_k(\mathbf{z}_1^T\mathbf{s})g(\sum_{i=1}^{n}(\mathbf{z}_i^T\mathbf{s})^2)\right\} - z_1^k E\left\{ g(\sum_{i=1}^{n}(\mathbf{z}_i^T\mathbf{s})^2) + 2(\mathbf{z}_1^T\mathbf{s})^2 g'(\sum_{i=1}^{n}(\mathbf{z}_i^T\mathbf{s})^2)\right\} \tag{4}$$

We denote subspace size by n and consider the first subspace for notational simplicity. The lower index on z indicates the vector it is taken from, and the upper index indicates the position within that vector.

Now we assume that we are near to a solution up to a perturbation ε, so $\mathbf{Z} = \mathbf{A}^T\mathbf{W}$ is near diagonal, and the vector $\mathbf{z}_1$ is of the form $\mathbf{z}_1 = (1+\varepsilon_1, \varepsilon_2, \varepsilon_3, ...)^T$. This is a special case, since in general $\mathbf{Z}$ will converge to a permutation of a block-diagonal matrix with blocks of size n. However, we do not lose generality by considering this particular case only. We follow the update step and analyze the dynamics of the perturbation that we introduced. Since we are going to show that the convergence is quadratic, it is sufficient to write out terms that are linear in the perturbation and show that they vanish. Thus we will use the notation $O(||.||^2)$ for all terms of a higher (i.e. smaller) order than linear terms. To evaluate the above expression, we need to make two approximations. At first we expand all occurrences of the square term as follows:

$$\sum_{i=1}^{n} (\mathbf{z}_i^T \mathbf{s})^2 = \sum_{i=1}^{n} (z_i^{iT} s_i)^2 + 2\sum_{\ell=1}^{n} (z_\ell^\ell s_\ell)(\mathbf{z}_{-\ell}^{\ell T} \mathbf{s}_{-\ell}) + O(||\varepsilon||^2) \tag{5}$$

where $\mathbf{z}_{-k}^T \mathbf{s}_{-k}$ denotes the inner product of the vectors with the k-th element removed. To separate the linear term out of the function g, we make a Taylor expansion up to the second term,

$$g(\sum_{i=1}^{n} (\mathbf{z}_i^T \mathbf{s})^2) = g(\sum_{i=1}^{n} (z_i^i s_i)^2) + 2\sum_{\ell=1}^{n} (z_\ell^\ell s_\ell)(\mathbf{z}_{-\ell}^{\ell T} \mathbf{s}_{-\ell}) g'(\sum_{i=1}^{n} (z_i^i s_i)^2) + O(||\varepsilon||^2) \tag{6}$$

The same expansion applied to $g'(.)$ gives, evaluating the series to the same order,

$$g'(\sum_{i=1}^{n} (\mathbf{z}_i^T \mathbf{s})^2) = g'(\sum_{i=1}^{n} (z_i^i s_i)^2) + 2\sum_{\ell=1}^{n} (z_\ell^\ell s_\ell)(\mathbf{z}_{-\ell}^{\ell T} \mathbf{s}_{-\ell}) g''(\sum_{i=1}^{n} (z_i^i s_i)^2) + O(||\varepsilon||^2) \tag{7}$$

now we can substitute these expressions into the original formula for the update step. We have split $\mathbf{z}^T \mathbf{s}$ into a perturbation and an unperturbed term, so we get

$$\begin{aligned} z_k^{1+} = E\Bigg\{ s_k (z_k^1 s_k + \mathbf{z}_{-k}^{1T} \mathbf{s}_{-k}) \Bigg[g(\sum_{i=1}^{n} (z_i^i s_i)^2) + 2\sum_{\ell=1}^{n} (z_\ell^\ell s_\ell)(\mathbf{z}_{-\ell}^{\ell T} \mathbf{s}_{-\ell}) g'(\sum_{i=1}^{n} (z_i^i s_i)^2) \Bigg] \Bigg\} \\ - z_k^1 E\Bigg\{ \Bigg[g(\sum_{i=1}^{n} (z_i^i s_i)^2) + 2\sum_{\ell=1}^{n} (z_\ell^\ell s_\ell)(\mathbf{z}_{-\ell}^{\ell T} \mathbf{s}_{-\ell}) g'(\sum_{i=1}^{n} (z_i^i s_i)^2) \Bigg] \\ + 2(z_k^1 s_k + \mathbf{z}_{-k}^{1T} \mathbf{s}_{-k})^2 \Bigg[g'(\sum_{i=1}^{n} (z_i^i s_i)^2) + 2\sum_{\ell=1}^{n} (z_\ell^\ell s_\ell)(\mathbf{z}_{-\ell}^{\ell T} \mathbf{s}_{-\ell}) g''(\sum_{i=1}^{n} (z_i^i s_i)^2) \Bigg] \Bigg\} + O(||\varepsilon||^2) \end{aligned}$$

We can now analyze the behavior of this for values of k that are either within the subspaces under consideration or outside of it. For $k \leq n$, the subspace under consideration is the one that we have nearly converged to. In this case we do not require that the change in an individual variable goes to zero, since the algorithm can only determine each subspace up to an arbitrary rotation. Thus a linear term may remain. For k being larger that n however, we shall show the quadratic convergence in the following. We separate and expand the sums and take them out of the expectations, so we can clearly see the order of the individual terms. For clarity, we split $z_k^{1+} = A + B + O(||\varepsilon||^2)$.

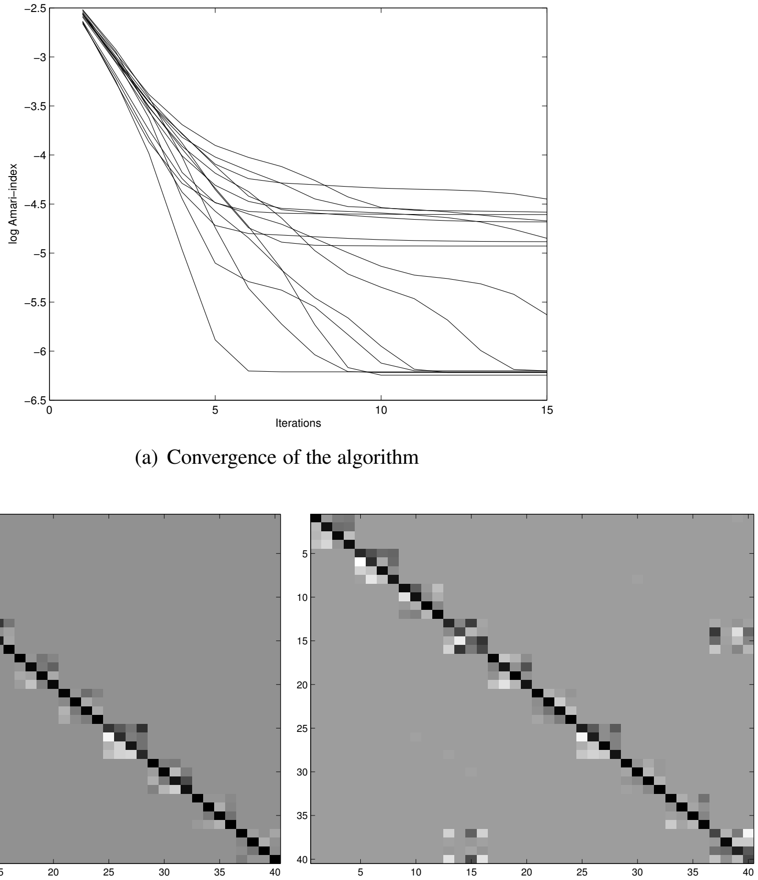
$$\begin{aligned} A \quad = \quad & E\left\{ z_k^1 (s_k)^2 g(\sum_{i=1}^{n} (z_i^i s_i)^2) \right\} + \sum_{h \neq k} E\left\{ z_h^1 s_h s_k g(\sum_{i=1}^{n} (z_i^i s_i)^2) \right\} \\ + \quad & \sum_{i=1}^{n} \sum_{j \neq i} E\left\{ z_k^1 (s_k)^2 2 (z_i^i s_i) z_j^i s_j g'(\sum_{i=1}^{n} (z_i^i s_i)^2) \right\} \\ + \quad & \sum_{h \neq k} \sum_{i=1}^{n} \sum_{j \neq i} E\left\{ z_h^1 s_h s_k 2 (z_i^i s_i) z_j^i s_j g'(\sum_{i=1}^{n} (z_i^i s_i)^2) \right\} \end{aligned}$$

$$\begin{aligned}
B = \quad & - \quad z_k^1 E\left\{ g(\sum_{i=1}^{n} (z_i^i s_i)^2) \right\} - 2 z_k^1 \sum_{i=1}^{n} \sum_{j \neq i} E\left\{ (z_i^i s_i) z_j^i s_j g'(\sum_{i=1}^{n} (z_i^i s_i)^2) \right\} \\
& - \quad 2 z_k^1 E\left\{ (z_k^1 s_k)^2 g'(\sum_{i=1}^{n} (z_i^i s_i)^2) \right\} - 4 z_k^1 \sum_{\ell=1}^{n} \sum_{h \neq k} E\left\{ z_h^1 s_h (z_k^1 s_k) g'(\sum_{i=1}^{n} (z_i^i s_i)^2) \right\} \\
& - \quad 2 z_k^1 \sum_{\ell \neq k} \sum_{h \neq k} E\left\{ (z_h^1 s_h)(z_\ell^1 s_\ell) g'(\sum_{i=1}^{n} (z_i^i s_i)^2) \right\} \\
& - \quad 4 z_k^1 \sum_{i=1}^{n} \sum_{j \neq i} \sum_{\ell=1}^{n} E\left\{ (z_k^1 s_k)^2 (z_\ell^\ell s_\ell) z_j^i s_j g''(\sum_{i=1}^{n} (z_i^i s_i)^2) \right\} \\
& - \quad 8 z_k^1 \sum_{h \neq k} \sum_{i=1}^{n} \sum_{j \neq i} \sum_{\ell=1}^{n} E\left\{ z_h^1 s_h (z_k^1 s_k)(z_\ell^\ell s_\ell) z_j^i s_j g''(\sum_{i=1}^{n} (z_i^i s_i)^2) \right\} \\
& - \quad 4 z_k^1 \sum_{\ell \neq k} \sum_{h \neq k} \sum_{i=1}^{n} \sum_{j \neq i} \sum_{\ell=1}^{n} E\left\{ (z_h^1 s_h)(z_\ell^1 s_\ell)(z_i^i s_i) z_j^i s_j g''(\sum_{i=1}^{n} (z_i^i s_i)^2) \right\}
\end{aligned}$$

The first terms of A and B cancel since s_k^2 in A is independent and has an expectation of unity due to the unit norm properties, e.g. $E\{s_k^2\} = 1$. The second term in A is zero because again s_k can be separated due to independence, and it has zero mean, i.e.$E\{s_k\} = 0$. This leaves only terms that are either proportional to z_k^2 or to other squared terms of off-diagonal elements. Since these are of the same order, they fall into the category of $O(||\varepsilon||^2)$. We have thus shown that the linear terms in ε vanish, so we have established quadratic convergence. It should be noted that the proof did not depend on the assumption of spherical symmetry that is typically made with ISA. The algorithm converges for more general dependency structures as well.

4 Simulations

To investigate the convergence properties of FastISA, we generated mixtures of supergaussian data with an embedded subspace structure, and used the algorithm to identify the sources. The data was generated by first taking 50,000 samples from a 40-dimensional white Gaussian distribution with zero mean and unit variance. We then divided this into subspaces of dimensionality four, and multiplied each member of a subspace by a random variable drawn from a uniform distribution. This serves a dual purpose, as it produces the required supergaussian distribution, and also introduces dependencies in the subspaces. We randomly generated a mixing matrix to obtain the observed mixtures, which were then whitened. As can be seen in figure 2a, only few steps are required to achieve convergence. To investigate whether the algorithm converged to a local minimum instead of the global solution, we computed the matrix $\mathbf{Z} = \mathbf{A}^T\mathbf{W}$. Initializing $\mathbf{W}$ randomly, it was observed that convergence was always to a local minimum. Therefore we validated the convergence properties by starting the optimization not on a random point on the error surface, but close to the optimal solution,



(a) Convergence of the algorithm

(b) Z reaching the global minimum, 40-dim. data

(c) Z after reaching a local minimum, 40-dim. data

Figure 1: Simulations on the algorithm: **(a)** The convergence is fast, the algorithm usually converges to a minimum in 5-15 steps. The algorithm was initialized with the correct solution perturbed by white noise of unit norm. Under these conditions, convergence is to the global minimum for 6 of the 15 random trials. **(b)** The product **Z** of the mixing and filter matrices is plotted, which gives a block-diagonal matrix for the global optimum. The residual log Amari-index is -6.2 which corresponds to an residual error of the order 10^{-3}. This is mainly due to the assumption of infinite expectations. **(c)** It cannot be guaranteed in general that the global minimum of the error surface is found. Here a local minimum is reached, indicated by multiple blocks in the bottom and leftmost position. The log Amari-index [4] here is -5, confirming that this is a local minimum.

which is known in this case since the mixtures are artificially generated. Under these conditions we get convergence to the global minimum, given that the starting point was close enough. This is depicted in Fig. 2c for data with a dimensionality of 40. $\mathbf{Z}$ should converge to a permuted block-diagonal matrix, since rotations inside subspaces do not affect the likelihood. Here, the Amari-index[4] for subspaces was computed by adding up the absolute values of all elements of the blocks of the main diagonal.

5 Conclusion

We presented a fixed-point algorithm for ISA, analogous to the ones presented in [1, 5]. The convergence of the algorithm was proven to be quadratic. Simulations show that the convergence is fast, but they also point out the problem of local minima. The problem of local minima is probably more related to the model specification itself because it was already encountered in [6], and not due to our particular algorithm.

References

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

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

[3] U. Köster A. Hyvärinen. Complex cell pooling and the statistics of natural images. *submitted to Network: Computation in Neural Systems*, available online: cs.helsinki.fi/u/koster/koster05.pdf, 2005.

[4] S. Cichocki, A.; Amari. *Adaptive Blind Signal and Image Processing. Learning Algorithms and Applications*. Wiley, 2002.

[5] A. Hyvärinen E. Bingham. A fast fixed-point algorithm for independent component analysis of complex valued signals. *Journal of Neural Systems*, 10:1–8, 2000.

[6] A. Hyvärinen and P. O. Hoyer. Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705–1720, 2000.