LOCAL CONVERGENCE PROPERTIES OF FASTICA AND SOME GENERALISATIONS

Knut Hüper, Hao Shen, Abd-Krim Seghouane

Department of Information Engineering
Research School of Information Sciences and Engineering
The Australian National University
Canberra ACT 0200, Australia

Systems Engineering and Complex Systems Research Program
National ICT Australia
Canberra Research Laboratory, Locked Bag 8001
Canberra ACT 2612, Australia

Knut.Hueper@nicta.com.au, Hao.Shen@rsise.anu.edu.au, Abd-krim.Seghouane@nicta.com.au

ABSTRACT

In recent years, algorithms to perform Independent Component Analysis in blind identification, localisation of sources or more general in data analysis have been developed. Prominent example certainly is the socalled FastICA algorithms from the Finnish school. In this paper we will generalise the FastICA algorithm considered as a discrete dynamical system on the unit sphere to the case where all units converge simultaneously, i.e., we consider some kind of parallel FastICA algorithm living on the orthogonal group. In addition we present a local convergence analysis for the algorithms proposed in this paper building on earlier work. It turns out that one can treat these type of algorithms in a similar manner as the Rayleigh quotient iteration, well known in numerical linear algebra, i.e. considering the algorithm as a discrete dynamical system on a suitable manifold. The algorithms presented here are compared by several numerical experiments and simulations.

1. INTRODUCTION

Blind Source Separation (BSS) is a challenging problem in Statistical Signal Processing. Since the influential paper [4], in the area of Independent Component Analysis (ICA), several efficient ICA algorithms have been developed to solve the BSS problem successfully. The FastICA algorithm is a prominent ICA algorithm proposed by the Finnish school around Hyvärinen, Karhunen and Oja, see [5].

The standard FastICA algorithm is a self mapping of the unit sphere for solving a one-unit linear ICA problem. Recently the first two authors have shown, that for the ideal linear ICA model, source signals can be recovered at certain fixed points of the algorithmic mapping of FastICA. The algorithm has a local quadratic rate of convergence, see [1].

In practice, one would prefer to reconstruct multiple signals in parallel under certain situations. A socalled symmetric orthogonalisation was proposed to parallelise the FastICA algorithm, see [5]. In this work we generalise the idea of parallelisation of FastICA. For the algorithms we propose here we take the standard FastICA as a starting point and develop several parallel generalisations. The algorithms are defined on the orthogonal group O_m of order m being a straightforward generalistaion of the sphere case. Building on earlier work local quadratic convergence of our algorithms are proved by means of calculus on manifolds, [3],[1],[2]. This in particular means that theoretically all signals are extracted simultaneously locally quadratically fast. One obvious advantage of the methods presented in this paper is that accumulated errors from deflation will not occur. The overall computational complexity seems to be lower than for FastICA including deflation. Due to the page restrictions we cannot further comment on this.

Recall, the whitened demixing ICA model can be formulated by the relation $Z = X^\top W$, where $W \in \mathbb{R}^{m\times n}$ is the whitened observation, the orthogonal matrix $X \in \mathbb{R}^{m\times m}$ is a parameterisation as the demixing matrix, and $Z \in \mathbb{R}^{m\times n}$ is the recovered signal, see [5].

2. ALGORITHMS

Let upper case letters denote matrices. The set S^{n-1} denotes the set of vectors in $\mathbb{R}^n$ with unit norm. Let denote $x \in S^{m-1}$ a column of the matrix $X = [x_1, \ldots, x_m]$ and $w \in \mathbb{R}^m$ one of the columns of W, respectively. By $^\top$ we denote transposition. I denotes the identity matrix.

National ICT Australia is funded by the Australian Government's Department of Communications, Information Technology and the Arts and the Australian Research Council through Backing Australia's Ability and the ICT Centre of Excellence Program.

1-4244-0469-X/06/$20.00

The standard one-unit FastICA algorithm can be formulated as a self map

$$\psi : S^{m-1} \to S^{m-1}, x \mapsto \frac{\mathbb{E}[G'(x^\top w)w] - \mathbb{E}[G''(x^\top w)]x}{\|\mathbb{E}[G'(x^\top w)w] - \mathbb{E}[G''(x^\top w)]x\|}. \quad (1)$$

Here the function $G : \mathbb{R} \to \mathbb{R}$ is a user defined contrast function being sufficiently smooth, convex and even, G', G'' being first and second derivatives. The central computational step in our generalisations of FastICA is still a one-unit FastICA step.

Parallel FastICA with so-called symmetric orthogonalisation can be restated as follows, [5]

Algorithm 2.1 *1. Initialise* $X^{(0)} = [x_1^{(0)}, \ldots, x_m^{(0)}] \in O_m$. *Set* $k = 0$.

2. For $i = 1, 2, \ldots, m$, *compute* $x_i^{(k+1)} = \psi(x_i^{(k)})$.

3. Set $\widehat{X}^{(k+1)} = [x_1^{(k+1)}, \ldots, x_m^{(k+1)}]$. *Compute* $X^{(k+1)} = (\widehat{X}^{(k+1)}\widehat{X}^{(k+1)\top})^{-1/2}\widehat{X}^{(k+1)}$ *(polar decomposition).*

4. If $\|X^{(k+1)} - X^{(k)}\|$ *(Frobenius norm) is small enough stop; otherwise set* $k = k + 1$ *and goto 2.*

In the sequel we will call the inner iterations over the number of columns a *sweep*. A few direct modifications to improve the efficiency of Algorithm 2.1 are in order here. In *step 3*, rather than performing reorthogonalisation by a polar decomposition requiring an SVD, we replace it by the Gram-Schmidt orthogonalisation process, efficiently implemented by a QR-decomposition. There are two important differences between these two orthogonalisation approaches. Firstly, the QR-approach will ensure that the iterates of the first column of the matrix X are exactly the same as the ones we would get by iterating one-unit FastICA to this column alone. This is not ensured by the polar decomposition. To the best of the authors' knowledge the convergence of the polar decomposition aproach is not supported by any theory up to now. A further minor improvement is to skip any FastICA iterations on the last column. After finishing the FastICA step of the second last column, the last column is up to sign already uniquely determined by the subsequent Gram-Schmidt process. Any further refinement during the present sweep on the last column would be just wasted time.

Hence, the modified parallel FastICA is as follows

Algorithm 2.2 *1. Initialise* $X^{(0)} = [x_1^{(0)}, \ldots, x_m^{(0)}] \in O_m$. *Set* $k = 0$.

2. For $i = 1, 2, \ldots, m-1$, *compute* $x_i^{(k+1)} = \psi(x_i^{(k)})$.

3. Set $\widehat{X}^{(k+1)} = [x_1^{(k+1)}, \ldots, x_{m-1}^{(k+1)}, x_m^{(k)}]$. *Reorthogonalise* $\widehat{X}^{(k+1)} = QR$ *(QR-dec.). Set* $X^{(k+1)} = Q$.

4. If $\|X^{(k+1)} - X^{(k)}\|$ *(Frobenius norm) is small enough stop; otherwise, set* $k = k + 1$ *and goto 2.*

The Algorithms 2.1 and 2.2 share the feature that during each sweep (*step 2*), FastICA is applied to all columns independently and simultaneously. The parallelism is obvious. To the best of our knowledge there exists up to now no theory which could ensure that after each sweep the matrix X has still full rank. Even worse, closely related but simpler algorithms recently proposed in the numerical linear algebra community to compute eigenvectors of a real symmetric matrix via a parallel version of Rayleigh Quotient Iteration lack such a property as well, see [6],[2] for details. On the other hand, most of the algorithms mentioned in [6],[2] and all algorithms presented here, share the property that they are locally welldefined around fixed points and locally smooth as well. Consequently, we can apply calculus to explore their local convergence properties.

To the best of our knowledge not very much is known about global convergence properties of the one-unit FastICA algorithm. Even if we would know that FastICA converged globally one could not easily derive from this similar properties for Algorithm 2.1. It is even some kind of miracle that by numerical evidence this algorithm seems to work sometimes well. See the experiments below. The situation is different for Algorithm 2.2. By construction, the sequence of iterates of the first column of the matrix X would certainly converge globally as well being a one-unit FastICA sequence by itself.

Because FastICA seems to converge almost always, we propose the following generalisation.

Algorithm 2.3 *1. Initialise* $X^{(0)} = [x_1^{(0)}, \ldots, x_m^{(0)}] \in O_m$. *Set* $k = 0$.

2. Compute $x_1^{(k+1)} = \psi(x_1^{(k)})$.

3. For $i = 2, \ldots, m-1$
compute $[x_1^{(k+1)}, \ldots, x_{i-1}^{(k+1)}] = QR$ *(QR-dec),*
compute $y = (I_m - QQ^\top)x_i^{(k)}$,
set $\widehat{x}_i^{(k+1)} = y/\|y\|$,
evaluate $x_i^{(k+1)} = \psi(\widehat{x}_i^{(k+1)})$.

4. Set $\widehat{X}^{(k+1)} = [x_1^{(k+1)}, \ldots, x_{m-1}^{(k+1)}, x_m^{(k)}]$. *Reorthogonalise* $\widehat{X}^{(k+1)} = QR$ *(QR-dec.). Set* $X^{(k+1)} = Q$.

5. If $\|X^{(k+1)} - X^{(k)}\|$ *(Frobenius norm) is small enough stop; otherwise, set* $k = k + 1$ *and goto 2.*

Note that in Algorithm 2.3 the QR-decomposition of Step 3 can be considered being unique and smooth even in the case of a rectangular matrix as long as it is full rank.

Experimentally, Algorithms 2.2, 2.3 share a common feature. After the first few sweeps, columns closer to the first one produce better estimates than the ones closer to the last column. Our next algorithm privileges left columns over right ones to intensify this effect.

Algorithm 2.4 *1. Initialise* $X^{(0)} = [x_1^{(0)}, \ldots, x_m^{(0)}] \in O_m$. *Set* $k = 0$.

2. For $j = 2, \ldots, m-1$

(a) For $i = 1, \ldots, j-1$
set $x_i^{(k)} = \psi(x_i^{(k)})$

(b) Compute $[x_1^{(k)}, \ldots, x_{j-1}^{(k)}] = QR$ *(QR-dec).*
Compute $y = (I_m - QQ^\top)x_j^{(k)}$.
Set $x_j^{(k)} = y/\|y\|$.

3. *For* $i = 1, \ldots, m-1$
   compute $x_i^{(k+1)} = \psi(x_i^{(k)})$.
4. *Set* $\widehat{X}^{(k+1)} = [x_1^{(k+1)}, \ldots, x_{m-1}^{(k+1)}, x_m^{(k)}]$.
   Reorthogonalise $\widehat{X}^{(k+1)} = QR$ *(QR-dec.).*
   Set $X^{(k+1)} = Q$.
5. *If* $\|X^{(k+1)} - X^{(k)}\|$ *(Frobenius norm) is small enough stop; otherwise, set* $k = k+1$ *and goto 2.*

All the above algorithms can be further modified. At each sweep, rather than applying one standard FastICA onto each column, we can increase the number of FastICA per column, see the simulations below.

3. NUMERICAL EXAMPLES

To illustrate the performance of our FastICA algorithms, we consider a classical audio signal separation example, see http://www.cis.hut.fi/projects/ica/cocktail/cocktail_en.cgi. The dataset consists of nine different sound signals.

We applied Algorithms 2.1-2.4 using up to 16 FastICA iterations per column. Figure 1-4 illustrate the convergence properties of these algorithms measured by the distance of the accumulation point to the current iterate, i.e. by $\|X^{(k)} - X^*\|$, with X^* the demixing matrix. By increasing the number of FastICA iterations per column in Algorithm 2.1 the convergence rate slows down. The opposite is true for Algorithms 2.2-2.4, the number of sweeps required to reach a certain accuracy is significantly smaller. Certainly, increasing the number of FastICA iterations per column increases the computational burden per sweep as well.

We then fixed the number of FastICA iterations per column to the value of four. In Fig. 5-8 the distance $\|x_i^{(k)} - x_i^*\|$ for $i = 1,\ldots,9$ is plotted against the number of sweeps required. What one can observe from the last four figures is that Algorithm 2.1 converges, if at all, extremely slowly, producing oscilations for all the signals. We conjecture that this is due to the wrong reorthogonalisation process used (polar decomposition). However, Algorithms 2.2-2.4 do converge without oscillation phenomena. These three algorithms share the feature of reconstructing the individual source signals with different speed.

4. LOCAL CONVERGENCE

We conclude by briefly discussing local quadratical convergence for Algorithms 2.2-2.4. The number of FastICA iterations per column will not matter our analysis as long as it is greater or equal to 1. The following Theorem is a theoretical result, usually not verified by real world applications or simulations due to the finiteness of the sample space. This is in particular the case for a set of nonperiodic signals to be decomposed into independent ones.

Heuristically, Algorithms 2.2-2.4 decouple asymptotically into independent and individual FastICA iterations on each column. As shown in [1] one-unit FastICA is locally quadratically convergent. One is tempted to conclude that the overall algorithm would then have the same convergence properties. This is indeed the case.

Theorem 4.1 *If Algorithm 2.2, 2.3, or 2.4 converges to the demixing matrix* X^**, it will converge locally quadratically fast.*

PROOF (SKETCH). We will consider each of the algorithms as a self map $\psi : O_m \to O_m$ on the set of orthogonal matrices. It is easily seen that the demixing matrix X^* is indeed a fixed point. The derivative of this map evaluated at X^* is the linear map $\mathrm{D}\,\psi(X^*) : T_{X^*}O_m \to T_{X^*}O_m$ with $T_{X^*}O_m$ the tangent space of O_m at X^*. Each iteration consists of a composition of FastICA transformations and projections followed by a reorthogonalisation step using a QR-decomposition. By the chain rule, the derivative of one iteration step is therefore the composition of the linear maps corresponding to the derivatives of the individual transformations. Each partial step, either FastICA or projection affects only a certain column leaving all the other columns invariant. Using Corollary 3.1 in [1] the derivative of such a partial step annihilates first order perturbations in the corresponding column leaving invariant all other first order perturbations. It is easily seen that the derivative of a projection evaluated at the fixed point acts as the identity. The same holds true for the derivative of the QR-decomposition. The result follows by a Taylor-type argument, i.e.,

$$\|\psi(X^k) - X^*\| \leq \sup_{Z \in \bar{U}} \|\,\mathrm{D}^2\,\psi(Z)\| \cdot \|X^{(k)} - X^*\|^2 \quad (2)$$

with $\bar{U}$ being the closure of a sufficiently small open neighborhood of $X^* \in O_m$. □

5. REFERENCES

[1] H. Shen and K. Hüper, "Local Convergence Analysis of FastICA," submitted to ICA 2006, Charleston, USA.

[2] K. Hüper, *Mathematical Systems Theory in Biology, Communications, Computation*, vol. 134 of *The IMA Volumes in Mathematics and its Applications*, chapter: A Dynamical System Approach to Matrix Eigenvalue Algorithms, pp. 257–274, Springer, New York, 2003.

[3] M. Nikpour, K. Hüper, and J.H. Manton, "Generalizations of the Rayleigh quotient iteration for the iterative refinement of the eigenvectors of real symmetric matrices.," in *Proceedings of the* 30^{th} *IEEE ICASSP*, Philadelphia, USA, 2005, pp. V1041–V1044.

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

[5] A. Hyvärinen, J. Karhunen, and E Oja, *Independent Component Analysis*, Wiley, New York, 2001.

[6] M. Nikpour, *Reduced Rank Signal Processing*, Ph.D. thesis, University of Melbourne, Australia, 2002.

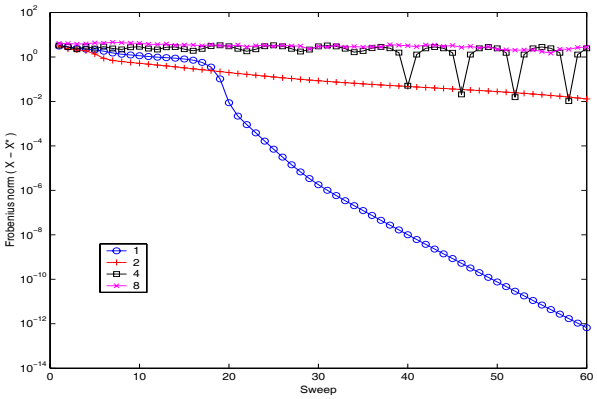

Fig. 1. Convergence of Algorthm 2.1.

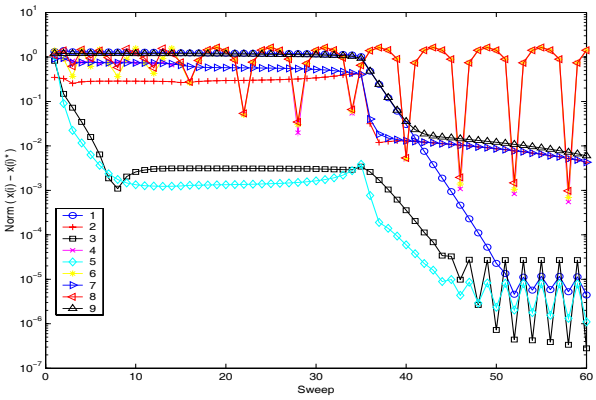

Fig. 5. Convergence of all nine columns using Algorithm 2.1.

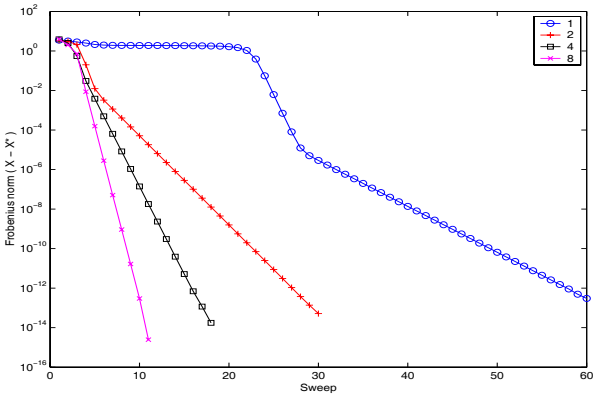

Fig. 2. Convergence of Algorithm 2.2.

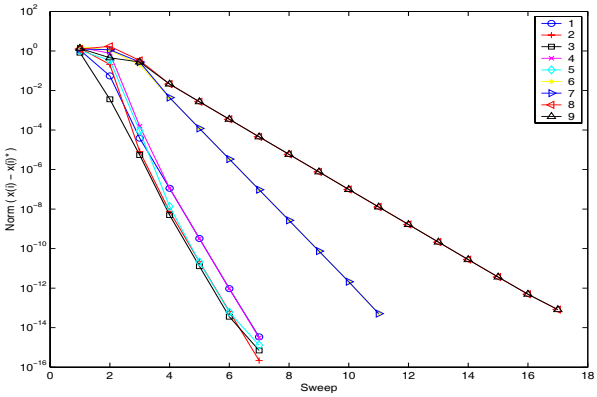

Fig. 6. Convergence of all nine columns using Algorithm 2.2.

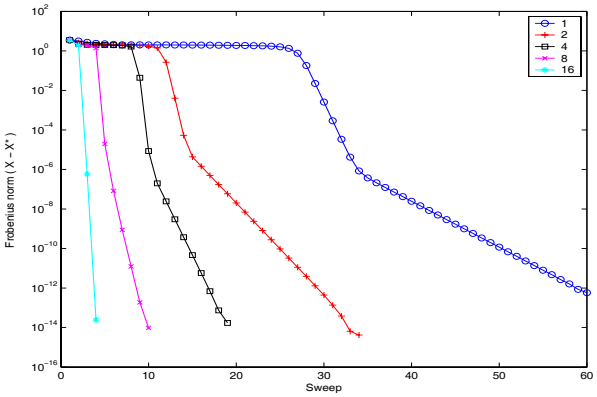

Fig. 3. Convergence of Algorithm 2.3.

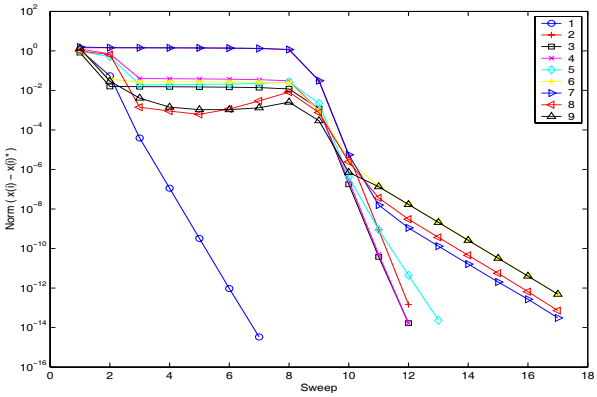

Fig. 7. Convergence of all nine columns using Algorithm 2.3.

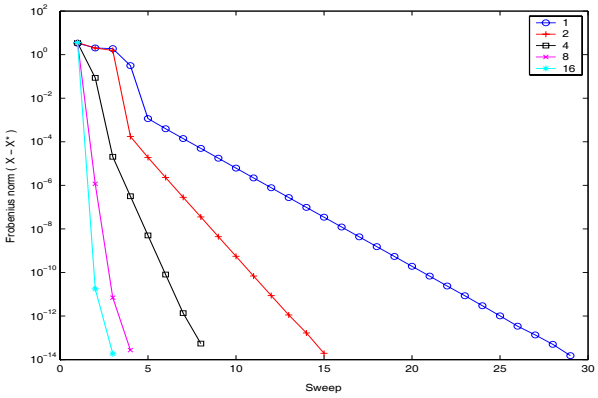

Fig. 4. Convergence of Algorithm 2.4.

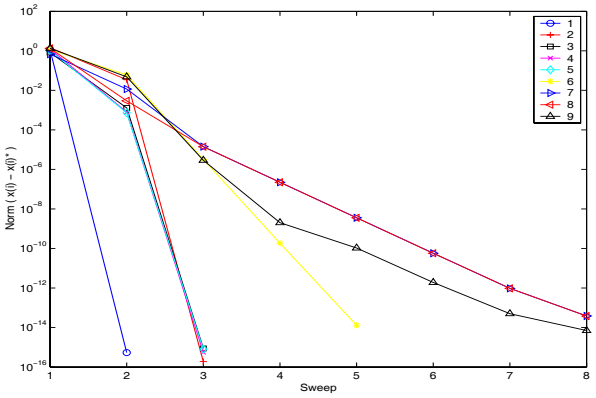

Fig. 8. Convergence of all nine columns using Algorithm 2.4.