A Family of Gradient-based Independent Component Analysis Techniques for Dynamic Channel Environment: An Overview

THOMAS YANG
Department of Electrical and Systems Engineering
Embry-Riddle Aeronautical University
Daytona Beach, Florida 32114
USA
yang482@erau.edu

Abstract: - This paper provides an overview of a family of gradient-based Independent Component Analysis (ICA) techniques. The advantage of these algorithms is the capability to perform blind source separation in dynamic channel environment, a scenario often encountered in many practical applications. A unified presentation is attempted to coherently introduce both block and sequential algorithms. The inherent connections among these various methods are illustrated, which provides a guideline for the selection of a particular method in practice.

Key-Words: independent component analysis, gradient method, blind source separation, dynamic environment

1 Introduction
Independent Component Analysis (ICA) is a powerful statistical technique that has attracted extensive research efforts in areas such as image and speech Processing, telecommunications, financial engineering, and biomedical signal processing. ICA is able to blindly extract statistically independent components from a set of observations that are linear combinations of these components, with little knowledge about the mixing process.

The classic Fast-ICA algorithm has been a highly successful technique adopted in many areas. However, due to its inherent fixed-point assumption, Fast-ICA exhibits poor convergence properties in applications where dynamic mixing process is involved in producing the observations. In this scenario, gradient-based methods are preferred due to its superior ability to adapt to time variations.

Recently, we developed various gradient based algorithms to address the problem of time-varying channel conditions in ICA processing [1-5]. In this paper, an overview of these techniques is presented in a unified framework.

2 Background
The basic ICA model is: \( X = AS \). Here, \( X \) is the observation matrix, \( A \) is the mixing matrix, and \( S \) is the source signal matrix consisting of independent components. The objective of ICA is to find a separation matrix \( W \), such that \( S \) can be recovered when the observation matrix \( X \) is multiplied by \( W \). This is achieved by making each component in \( WX \) as statistically independent as possible. Many principles have been adopted to accomplish this task, such as Maximization of Nongaussianity, Maximum Likelihood Estimation, Minimization of Mutual Information and Tensorial Methods.

The Newton-based Fast-ICA algorithm [6], based on Maximization of Nongaussianity, is a highly efficient algorithm that typically converges within less than ten iterations in a stationary environment. In most cases, the choice of the learning rate is avoided. However, when the mixing matrix is dynamic, Fast-ICA exhibits much degraded convergence properties.

In the following section, various gradient ICA methods we recently proposed to perform blind signal separation in dynamic environment are presented. The algorithms developed are used for estimating one row, \( w \), of the demixing matrix \( W \), and the estimation is performed repeatedly to obtain all rows of \( W \). The cost function adopted is the square of kurtosis, which is the simplest quantitative measure of nongaussianity. It is assumed that at most one source signal is Gaussian. The only difference between Fast-ICA and the presented techniques is the update equation for \( w \). Other procedures in the iterative process, such as preprocessing mean centering/whitening and post-processing orthogonalization, are identical.

3 A Family of Gradient ICA Methods
To present our proposed algorithms, the following parameters need to be defined first:
$$j$$: iteration index.
$$M$$: number of observations.

$$L$$: length of the processing block.

$$w(j) = [w_1(j), w_2(j), ..., w_M(j)]^T$$: the current row of the separation matrix for the $$j$$th iteration. ($$i = 1, 2, ..., M$$)

$$x_i(j)$$: the $$i$$th signal in the $$l$$th observation data vector for the $$j$$th iteration. ($$l = 1, 2, ..., L$$)

$$X_l(j) = [x_{l,1}(j), x_{l,2}(j), ..., x_{l,M}(j)]^T$$: $$l$$th signal observation for the $$j$$th iteration.

$$G_j = [X_1(j), X_2(j), ..., X_L(j)]^T$$: Observation matrix for the $$j$$th iteration.

### 3.1 Optimum Block Adaptive ICA with Individual Adaptation (OBAI-ICA) [3]

OBAI-ICA automatically computes the optimum convergence factors that are individualized for each component in the gradient vector. Also, these convergence factors are updated in every iteration.

The $$l$$th kurtosis value for the $$j$$th iteration is

$$kurt(j) = E\{[w^T(j)X_l(j)]^4\} - 3$$

(1)

where it is assumed that the signals and $$w(j)$$ both have been normalized to unit variance.

Then, the kurtosis vector for the $$j$$th iteration is

$$kurt(j) = [kurt_1(j), kurt_2(j), ..., kurt_L(j)]^T$$

(2)

Now the updating formula can be written as

$$w(j+1) = w(j) + [MU]_j \nabla_b(j)$$

(3)

where

$$\nabla_b(j) = \frac{\partial [kurt^T(j)kurt(j)]}{\partial w(j)}$$

$$= \frac{1}{L} \left[ \frac{\partial [kurt^T(j)kurt(j)]}{\partial w_1(j)}, \frac{\partial [kurt^T(j)kurt(j)]}{\partial w_2(j)}, ..., \frac{\partial [kurt^T(j)kurt(j)]}{\partial w_M(j)} \right]^T$$

(4)

and

$$[MU]_j = \begin{bmatrix} \mu_{11}(j) & 0 \\ \vdots & \vdots \\ 0 & \mu_{MM}(j) \end{bmatrix}$$

(5)

To evaluate (4), we have

$$\frac{\partial [kurt^T(j)kurt(j)]}{\partial w_i(j)} = \sum_{l=1}^{L} \frac{\partial [E\{[w^T(j)X_l(j)]^4\} - 3]}{\partial w_i(j)}$$

$$= 8 \sum_{l=1}^{L} [w^T(j)X_l(j)]^3 kurt_i(j) x_{i,l}(j)$$

(6)

In the derivation of (6), the expectation operator was dropped. The block gradient vector can be written as

$$\nabla_b(j) = \frac{8}{L} \left[ \sum_{l=1}^{L} [w^T(j)X_l(j)]^3 kurt_i(j) x_{i,l}(j) \right]$$

$$= \frac{8}{L} [G_j^T [C]_j kurt(j)]$$

(7)

Where

$$[C]_j = \begin{bmatrix} w^T(j)X_1(j) & \ldots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \ldots & w^T(j)X_L(j) \end{bmatrix}$$

(8)

From (7), the updating formula (3) becomes

$$w(j+1) = w(j) + \frac{8}{L} [MU]_j [G_j^T [C]_j kurt(j)]$$

(9)

Now, the task is to identify the matrix $$[MU]_j$$ in an optimal sense, so that the total squared kurtosis $$kurt^T(j)kurt(j)$$ is maximized. We express the $$l$$th kurtosis value in the $$(j+1)$$th iteration by Taylor’s series expansion.

$$kurt_l(j+1) = kurt_l(j) + \sum_{i=1}^{M} \frac{\partial kurt_l(j)}{\partial w_i(j)} \Delta w_i(j)$$

$$+ \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{l} \frac{\partial^2 kurt_l(j)}{\partial w_i(j) \partial w_j(j)} \Delta w_i(j) \Delta w_j(j) + ...$$

$$l = 1, 2, ..., L$$

(10)

where

$$\Delta w_i(j) = w_{i}(j+1) - w_i(j) \quad i = 1, 2, ...,$$
In (10), the complexity of the terms increases as the order of the derivative increases. However, if $\Delta w_i(j)$ is small enough, higher order derivative terms can be omitted.

The expectation operator in (1) is dropped. Thus,

$$\frac{\partial \text{kurt}_t(j)}{\partial w_i(j)} = 4x_i(j)[w^T(j)X_i(j)]^3$$  \hfill (12)

Then, (10) becomes

$$\text{kurt}_t(j+1) = \text{kurt}_t(j) + 4[w^T(j)X_i(j)]^3[X_i^T(j)\Delta w(j)]$$  \hfill (13)

Writing (13) for every $k$, the Taylor expansion becomes

$$\text{kurt}_t(j+1) = \text{kurt}_t(j) + 4[C_j]^T[G_j]_{i,L} \Delta w(j)$$  \hfill (14)

From (9),

$$\Delta w(j) = \frac{8}{L}[G_j]_{i,L} [C_j]^T kurt(j)$$  \hfill (15)

Substituting (15) into (14), one obtains

$$\text{kurt}_t(j+1) = \text{kurt}_t(j) + \frac{32}{L}[C_j]^T[G_j][M_j][G_j]_{i,L} [C_j]^T kurt(j)$$  \hfill (16)

Defining $q(j)$ and $[R_j]$ as

$$q(j) = [G_j]^T[C_j] kurt(j) = [q_1(j) ... q_m(j)]^T$$  \hfill (17)

$$[R_j] = [G_j]^T[C_j]^T[G_j] = [R_{mn}(j)] \quad 1 \leq m, n \leq N$$  \hfill (18)

The total squared kurtosis for the $(j+1)^{th}$ iteration is

$$\text{kurt}_t^T(j+1)\text{kurt}_t(j+1) = \text{kurt}_t^T(j)\text{kurt}(j) + \frac{64}{L} \sum_{i=1}^{L} q_i(j) \mu_{gb}(j) + \frac{1024}{L} q^T(j)[M_j][R_j][M_j] q(j)$$  \hfill (19)

In order to identify $[M_j]_i$ optimally, the following condition must be met:

$$\frac{\partial kurt_t^T(j+1)\text{kurt}(j+1)}{\partial \mu_{gb}(j)} = 0 \quad i = 1, 2, ..., M$$  \hfill (20)

Substituting (19) into (20), and using the symmetry property of the matrix $[R_j]$ given in (18), the following is obtained:

$$\sum_{k=1}^{M} q_k(j) \mu_{gb}(j) = -\frac{L}{32} q(j)$$  \hfill (21)

where * denotes the optimal value.

Writing (21) for every $i$, one obtains:

$$[R_j][M_j]^T q(j) = -\frac{L}{32} q(j)$$  \hfill (13)

From (17), (22) and (9), the optimum weight update equation is obtained:

$$w(j+1) = w(j) + \frac{8}{L} (-\frac{L}{32}) [R_j]^{*T} q(j) - w(j) - 0.25[R_j]^{*T} q(j)$$  \hfill (23)

where $[R_j]$ and $q(j)$ are given by (17) and (18).

### 3.2 Online Gradient ICA and Block ICA

It can be shown that block ICA can be obtained from OBAI-ICA, equation (23), by adopting a single uniform time-invariant convergence factor in equation (5). The resulting equation is:

$$w(j+1) = w(j) + \frac{8}{L} \mu_{gb}[G_j]^T[C_j] kurt(j)$$  \hfill (24)

The online gradient ICA can be obtained by further setting block size $L=1$, resulting in [7]:

$$w(j+1) = w(j) - \mu(\text{sign}(\text{kurt}(j))X(j)[w^T(j)X(j)]^3)$$  \hfill (25)

### 3.3 Optimum Block Adaptive ICA (OBA-ICA) [1]

In the derivation of OBAI-ICA, if we adopt a single uniform convergence factor in (5) but update it in each iteration, the OBA-ICA algorithm can be obtained after some derivation as:

$$w(j+1) = w(j) - 0.25 \frac{q^T(j)q(j)}{q^T(j)[R_j] q(j)} q(j)$$  \hfill (26)
The advantage of OBA-ICA is that, unlike OBAI-ICA, it does not require matrix inversion operation. This desirable fact makes OBA-ICA not only computationally efficient for high-order systems, but also applicable in scenarios in which the processing block size is less than the system order.

3.4 General Optimum Block Adaptive ICA (GOBA-ICA) [5]
For most applications, the probability distribution of the source signals is known in advance. Thus, a more straightforward method can be obtained by, at each iteration, setting the kurtosis vector in the next iteration directly to the vector consisted of the kurtosis values of individual source signals. For example, if the independent components are binary signals taking values of either -1 or 1 with equal probability, the kurtosis of the discrete uniform distribution is -2. Therefore, \( \text{kurt}(j+1) \) should be a vector of length \( L \) in which every element is -2. That is,

\[
\text{kurt}(j+1) = -K = [-2 \ -2 \ \ldots \ -2]^T \tag{27}
\]

Define

\[
[R]_j = 4[C]_j[G]_j \tag{28}
\]

Then, the change in the weight vector becomes

\[
\Delta w(j) = -[R]_j (Kurt(j) + K) \tag{29}
\]

where

\[
[R]_j = \begin{cases} ([R]_j ([R]_j)^{-1} & L < N \\ ([R]_j ([R]_j)^{-1}[R]_j)^{T} & L \geq N \end{cases} \tag{30}
\]

is the pseudo-inverse of the matrix \([R]_j\).

It is clear that (29) can be applied whether the block size \( L \) is greater than the system order \( N \) or not, thus the method is named GOBA-ICA. It can be considered the Least Square (LS) solution for the weight update equation exploiting apriori signal distribution constraint expressed by (27). It is also worth mentioned that, when \( L > N \), GOBA-ICA is the same as OBAI-ICA.

3.5 ICA with Individual Adaptation (IA-ICA) [2] and Homogeneous Adaptation (HA-ICA)

When the time-variation of the mixing process is very rapid, it is more appropriate to apply sequential versions of the above presented OBAI-ICA and OBA-ICA. Set block size \( L = 1 \), the sequential version of OBAI-ICA, named IA-ICA, can be obtained as:

\[
w(j + 1) = w(j) - 0.25 \text{kurt}(j) \left[ X(j)x^T(j) [w^T(j)x(j)] \right] \tag{31}
\]

Similarly, the sequential version of OBA-ICA, named HA-ICA, can be obtained as:

\[
w(j + 1) = w(j) - 0.25 \frac{\text{kurt}(j)X(j)}{[w^T(j)x(j)]^T[X(j)x(j)]} \tag{32}
\]

It is worth mentioning that in both HA-ICA and IA-ICA, the kurtosis vector degenerates into a scalar.

3.6 Block Conjugate-Gradient ICA (BC-ICA) [4]
In adaptive filtering research, conjugate-gradient techniques have been used to improve the convergence properties of the classic gradient methods. Here we present the BC-ICA, which adopts the conjugate gradient principle in block ICA adaptation.

Suppose \( w(j) \) is updated along direction \( r(j) \), i.e.,

\[
\Delta w(j) = \alpha r(j) \quad \text{ (33) }
\]

we need to calculate the optimum value for \( \alpha \) given \( r(j) \).

From (33), the Taylor series expansion (14) becomes

\[
kurt(j + 1) = kurt(j) + 4\alpha [C]_j^T [G], r(j) \tag{34}
\]

Defining \( q(j) \) and \( [R]_j \) as in (17) and (18), the total squared kurtosis can be expressed as:

\[
kurt^T(j + 1)kurt(j + 1) = kurt^T(j)kurt(j) \\
+ 8\alpha r^T(j)q(j) + 16\alpha^2 r^T(j)[R]_j r(j) \tag{35}
\]

Taking the derivative of the total squared
kurtosis (35) with respect to $\alpha$, and setting the resulting expression to zero, one obtains the optimum value of $\alpha$ given by
\[
\alpha = -0.25 \frac{r^T(j)q(j)}{r'(j)[R_j]r(j)}
\]
(36)

The remaining task is to identify the suitable weight update direction according to the conjugate-gradient method. In the first iteration, the conjugate-gradient method uses the gradient direction given by (7). Now, the BC-ICA algorithm can be described as follows:

(1). Initialize $w(0), j = 0$.

(2). According to (7), find the direction
\[
g(j) = \nabla_{\beta} = \frac{8}{L} [G]_j^T [C]_j^{T} kurt(j)
\]
(37)

\textbf{a}. If $w(j)$ has converged, then terminated the algorithm and return $w(j)$;

\textbf{b}. If $w(j)$ has not converged and $j = 0$, then $r(j) = g(j)$, and proceed to step (3);

\textbf{c}. If $w(j)$ has not converged and $j > 0$, then compute $r(j)$ from
\[
r(j) = g(j) + \frac{g^T(j)g(j)}{g^T(j-1)g(j-1)} r(j-1)
\]
(38)

(3). Update the weight vector according to
\[
w(j+1) = w(j) + \alpha r(j),
\]
where $\alpha$ is given by (36).

(4). $j = j + 1$, go to step (2).

This technique can be considered as intermediate lying between the steepest descent and Newton’s methods, in terms of complexity and convergence properties.

3.7 Further Extension to Complex-valued Signal Processing [8-10]
Many practical applications involve complex-valued signal processing. Recently, the complex versions of the above presented algorithms were also developed. The major idea is to use unique convergence factors for the real and imaginary parts of the demixing vector $w(j)$.

4 Practical Applications / Conclusion
To verify the effectiveness of all the presented ICA techniques including their complex versions, computer simulations were conducted in MATLAB, adopting these ICA methods to perform interference suppression for wireless communication systems [1-5, 8-10]. The obtained performance in time-varying channel environment is compared with Fast-ICA. It is clearly seen from the simulation results that the family of gradient-based ICA algorithms outperforms Fast-ICA in dynamic channel environment in terms of both signal-to-interference ratio and convergence properties, especially for large block sizes. There are various issues associated with practical implementation of these algorithms for such a specific application. Those are out of the scope of this paper, but some of the relevant discussions can be found in [1-5] and [8-10].

In conclusion, the recently proposed gradient-based ICA techniques are a family of highly efficient methods for blind signal separation in dynamic environment. These algorithms display superior convergence properties despite the time-variation in the mixing process. We believe there are numerous applications in which these techniques are the preferred methods for source separation.

References:


