QUANTITATIVE METHODS IN ECONOMICS
Vol. XV, No. 1, 2014, pp. 142 – 152

A COMPARATIVE STUDY OF FastICA AND GRADIENT ALGORITHMS FOR STOCK MARKET ANALYSIS

Kesra Nermend
Institute of IT in Management, University of Szczecin, Szczecin 71-415, Poland
email: kesra@wneiz.pl
Yasen Rajihy
Department of Computer Science, University of Babylon, Babylon 09332, Iraq
email: yasenasad@yahoo.com

Abstract: In this paper we proved that a fast fixed point algorithm known as FastICA algorithm depending on maximization the nongaussianity by using the negentropy approach is one of the best algorithm for solving ICA model. We compare this algorithm with Gradient algorithm. The Abu Dhabi Islamic Bank (ADIB) used as illustrative example to evaluate the performance of these two algorithms. Experimental results show that the FastICA algorithm is more robust and faster than Gradient algorithm in stock market analysis.

Keywords: independent component analysis, nangaussianity, negentropy, stock market analysis

INTRODUCTION

Independent Component Analysis (ICA) is a mathematical and computational technique for revealing hidden factors that underlie sets of random signals (variables) [Comon 1994; Jutten, Herault 1991; Hyvärinen et al. 2001]. These underlying latent variables are called sources or independent components (ICs) and they are assumed to be statistically independent of each other and nongaussian. The technique of ICA is a relatively new invention. In the middle of 1990s, some highly successful new algorithms for solving the ICA model were introduced by several research groups [Hyvärinen, Oja 2000; Hyvärinen 1999; Cardoso, Souloumiac 1993; Choi et al. 2001; Pham, Cardoso 2000; Pham, Garat 1997; Belouchrani et al. 1996; Jutten 2000; Jutten, Herault 1991]. The main mathematical problem of ICA can be described as follows: we observe m random variables $x_1, x_2, \cdots x_m$, which are modeled as linear combinations of n random variables $s_1, s_2, \cdots s_m$:

$$x_i = a_{i1}s_1 + a_{i2}s_2 \cdots a_{in}s_n \tag{1}$$

for all $i = 1, 2, ..., m$, where $a_{i,j}$ are some real mixing coefficients. The s_i are statistically mutually independent. The matrix representation of equation (1) can be expressed as:

$$\mathbf{x} = \mathbf{As} \tag{2}$$

where **x** is an observed m-dimensional vector, **s** is n-dimensional (latent) random vector whose components are assumed mutually independent, and **A** is a constant $m \times n$ matrix to be estimated. It is usually further assumed that the dimensions of **x** and **s** are equal, i.e. $m=n$. Basic ICA model describes how the observed data are generated by a process of mixing the independent components s_i. The independent components s_i are the latent variables which means that they are not observable directly. Also the mixing coefficients a_{ij} are unknown. ICA uses solely the observed data x_i to estimate both the ICs s_i and the mixing coefficients a_{ij}. The task in ICA is to find both the latent variables or sources s_i and the mixing process; in the linear case, the latter task consists of finding the mixing matrix **A**. A popular approach is to find a demixing matrix **W** so that variables y_i in **y=Wx** are estimates of s_i up to scaling and permutation. Hence W is an estimate of the (pseudo)inverse of **A** up to scaling and permutation of the rows of **W**. Often the latent variables s_i are estimated one by one, by finding a column vector w_i (this will be stored as a row of **W**) such that $y_i = w_i^T x$ is an estimate of s_i.

In order to calculate a demixing matrix ***W*** (i.e. to estimate ICs), numerous ICA algorithms have been developed with various approaches. In this paper we try to review the most important two algorithms to solve the ICA model based on maximization of nongaussianity by using negentropy approach, namely Gradient algorithm and FastICA algorithm [Hyvarinen 1997; Hyvärinen 1999; Hyvarinen, Oja 1997]. In practice, before application of these two algorithms, suitable preprocessing is often compulsory i.e. centering and whitening. The observed vector **X** is first centered by removing its mean. A zero-mean random vector $z=(z_1,z_2,\ldots,z_n)^T$ is said to be white or sphere if its components z_i are uncorrelated and have unit variances. This means that the covariance matrix (as well as the correlation matrix) of z equals the identity matrix. Centering and whitening can be accomplished by principal component analysis (PCA).

Abu Dhabi Islamic Bank (ADIB) used as illustrative example to evaluate the performance of these two algorithms. Experimental results show that FastICA is more robust and faster than Gradient algorithm in stock market analysis.

RESEARCH METHODOLOGY

Estimation the independent components (ICs) is a challenging task because ICA uses solely the observed data x_i to estimate both the ICs s_i and the mixing coefficients $a_{i,j}$. Several approaches for solving ICA model are presented during last decade. Maximization of nongaussianity based on negentropy is one of these approaches. Maximization of nongoussianity based on negentropy is a simple and intuitive principle for estimating the model of independent component analysis (ICA). Nongaussian components are independent. Nongaussianity is actually of high importance in ICA estimation. If the nongaussianity does not valid, then the estimation is not possible at all. To use nongaussianity in ICA estimation, we must have a quantitative measure of nongaussianity of a random variable, say x. To simplify things, let us assume that x is centered (zero-mean) and has variance equal to one. Actually, one of the functions of preprocessing in ICA algorithms is to make this simplification possible [Hyvärinen, Oja 2000]. One of the most important quantitative measures of nongaussianity is a negentropy as we show below.

Negentropy

Negentropy is based on the information theoretic quantity of (differential) entropy [Hyvärinen et al. 2001]. The entropy of a random variable can be interpreted as the information degrees of a given observe variable. The entropy H of a random vector $\boldsymbol{x}$ with density $f(\boldsymbol{x})$ is defined as:

$$H(\mathbf{x}) = -\int f(x) \log f(x) dx \tag{3}$$

A fundamental result of the information theory is that a gaussian variable has the largest entropy among all random variables of equal variance [Hyvarinen 1999]. The entropy is small when variables are far from the gaussian, hence it can be used in the measure of nongaussianity. The negentropy N of a nongaussian random vector X is defined as:

$$N(X) = H(Xg) - H(X) \tag{4}$$

where Xg is a gaussian random vector whose covariance matrix is equal to that of X. Note that negentropy is non-negative and zero if and only if the vector X has a gaussian distribution. The main problem in using negentropy is that it is computationally very difficult. Hence simpler approximations of negentropy are very useful. A classical method to approximate negentropy is using higher-order cumulants, for example as follows [Jones, Sibson 1987]:

$$N(X) \approx \frac{1}{12} E\{X^3\}^2 + \frac{1}{48} kurt\{X\}^2 \tag{5}$$

where X is assumed to be a zero mean and a unit variance. If the random variable X has a symmetric distribution, then the first term in the right-hand side of Eq. (8) is equal to zero, and so this approximation often leads to the use of kurtosis as in the preceding section. To avoid this problem we can perform the approximation by using a non-quadratic function *G* as follows:

$$N(X) \propto [\, E\{G(X)\} - E\{G(v)\}]^2 \tag{6}$$

where v is a gaussian variable with a zero mean and a unit variance (i.e. standardized). Here, one must choose G wisely. In particular, choosing *G* that does not grow too fast one obtains more robust estimators. The following choices of *G* have proved very useful

$$G_1(X) = \frac{1}{a_1} \log cosh\, a_1 X \tag{7}$$

$$G_2(X) = -Exp\left(\frac{-X^2}{2}\right) \tag{8}$$

Where $1 \leq a_1 \leq 2$ is a constant often taken equal one.

Negentropy, based on the information theoretic quantity of entropy is a best method of measuring nongaussianity, it can be conceptually simple, fast to compute, more robust, enable the deflationary (i.e. one-by-one estimation of independent components), and force the estimations of the independent components to be uncorrelated.

Gradient algorithm using negentropy

The main task in the independent component analysis (ICA) problem is to estimate a demixing matrix ***W*** that will give us the independent components. In this subsection we derive a simple gradient algorithm for maximizing negentropy. Taking the gradient of the approximation of negentropy in (6) with respect to *w*, and taking the normalization $E\{(w^T z)^2\} = \|w\|^2$ into account, one obtains the following algorithm [Hyvärinen et al. 2001]

$$\Delta w \propto \gamma E\{z g(w^T z)\} \tag{9}$$

$$w \leftarrow \frac{w}{\|w\|} \tag{10}$$

where $\gamma = E\{G(w^T z)\} - E\{G(v)\}$. The function g is the derivative of the function G used in equations (7) and (8). The parameter γ can be estimated on-line as follows:

$$\Delta\gamma \propto [G(w^T z) - E\{G(v)\}] - \gamma \tag{11}$$

The final form of the gradient algorithm is summarized as follows:

1. Center the data to make its mean zero.
2. Whiten the data to give z.
3. Choose an initial random vector w of unit norm, and an initial value for γ.
4. Update $\Delta w \propto \gamma z g(w^T z)$.
5. Normalize $w \leftarrow \frac{w}{\|w\|}$.
6. If the sign of γ is not known a priori, update $\Delta\gamma \propto [G(w^T z) - E\{G(v)\}] - \gamma$.
7. If not converged, go back to step 4.

FastICA algorithm using negentropy

In this subsection we derive the fixed-point algorithm (FastICA) using negentropy for maximizes the nongaussianity [Hyvärinen et al. 2001]. The resulting FastICA algorithm finds a direction, i.e., a unite vector w, such that the projection $w^T z$ maximizes the nongaussianity. Here nongaussianity measured by the approximation of negentropy $N(w^T z)$ given in (6). Recall that $w^T z$ must constraining to have a unit variance, this is equivalent to $\|w\| = 1$.
Looking at the gradient method in (9) immediately suggests the following fixed-point iteration:

$$z \leftarrow E\{zg(w^T z)\} \quad (12)$$

$$w \leftarrow \frac{w}{\|w\|} \quad (13)$$

The iteration in (12) has to be modified because it doesn't have a good convergence. This can easily do as follows:

$$(1+\alpha) = E\{zg(w^T z)\} + \alpha w \quad (14)$$

where α is a constant. One must choose α wisely to obtain an algorithm that converges faster than gradient algorithm.

FastICA can be found using Newton's method approximation. To derive the approximative Newton method, first note that the maxima of the approximation of the negentropy of $w^T z$ are typically obtained at certain optima of $E\{G(w^T z)\}$. According to the Lagrange conditions, the optima of $E\{G(w^T z)\}$ under the constraint $E\{(w^T z)^2\} = \|w\|^2 = 1$ are obtained at points where the gradient of the Lagrangian is zero [Hyvärinen et al. 2001]:

$$E\{zg(w^T z)\} + \beta w = 0 \quad (15)$$

To simply solve equation (15) by Newton's method, let $F = E\{zg(w^T z)\} + \beta w$, we obtain its gradient as:

$$\frac{\partial \mathrm{F}}{\partial \mathrm{w}} = \mathrm{E}\{\mathrm{zz}^\mathrm{T} \acute{g}\,(w^T z)\} + \beta I \quad (16)$$

Since the data is whitened, we can simplify the inversion of this matrix by approximate the first term in (16) as follows:

$$\mathrm{E}\{\mathrm{zz}^\mathrm{T} \acute{g}\,(w^T z)\} \approx E\{\mathrm{zz}^\mathrm{T}\} E\{\acute{g}\,(w^T z)\} = E\{\acute{g}\,(w^T z)\} I.$$

Thus the gradient becomes diagonal, and can easily be inverted. Thus we obtain the following approximative Newton iteration:

$$w \leftarrow w - \frac{E\{zg(w^T z)\} + \beta w}{\mathrm{E}\{\acute{g}\,(w^T z)\} + \beta} \quad (17)$$

After straightforward algebraic simplification we give the basic fixed-point iteration in FastICA:

$$w \leftarrow E\{zg(w^T z)\} - \mathrm{E}\{\acute{g}\,(w^T z)\} w \quad (18)$$

Then the basic form of the FastICA algorithm can be described as follows:

1. Center the data to make its mean zero.
2. Whiten the data to give z.
3. Let $w \leftarrow E\{zg(w^T z)\} - \mathrm{E}\{\acute{g}\ (w^T z)\}w$.
4. Let $w \leftarrow \frac{w}{\|w\|}$.
5. If not converged, go back to step 4.

These two algorithms just give estimates only one independent component. In practice, we have many more dimensions, and therefore, we usually want to estimate more than one independent component. This can be done by several methods.

EMPIRICAL RESEARCH

Forecasting stock market has been one of the biggest challenges to the scientific community. It requires the use of a possibly large set of input variables. Selection of a useful subset of input variables is a difficult task. ICA has been widely applied to financial time series analysis. It is use to extract the independent components (ICs) from a very complex data set, these ICs are statistically independent from each other. The ICA procedure reduces the number of input variables to a much smaller set of ICs. These ICs are expected to capture most of the useful information of original data.

Artificial Neural network (ANN) technique is regarded as more suitable for stock market forecasting than other techniques, they are able to learn and detect patterns or relationships from the data itself. Since properly estimated ICs are statistically independent from each other, we can use them as an input of neural network that can be used to forecasts of the stock market. In empirical study we use the Independent Component Analysis (ICA) as a preprocessing algorithm to forecast the stock market.

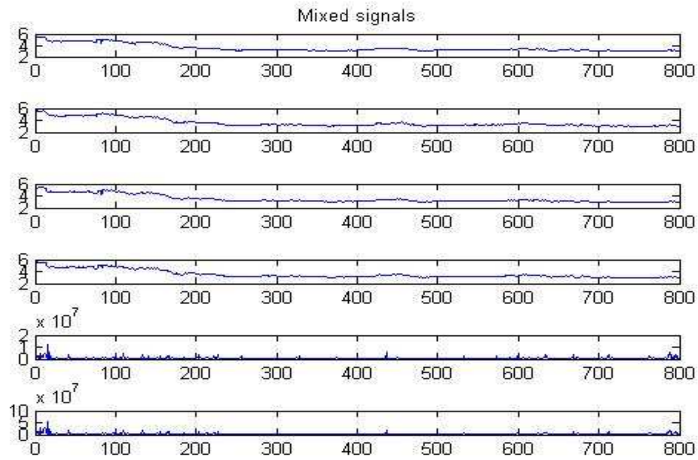
In our empirical, ICA is firstly applied to analyze the financial time series data to get statistically mutually independent components. The analyzed ICs are conducted as the input of NN for constructing a forecasting model. We will try to apply the historical data of the last trading day, including daily open, highest, lowest, closing price, daily volume and daily turnover as the input of NN, the output of the NN include the closing price of the next trading day.

For compering the performance of Gradient algorithm and FastICA algorithm, we select the data of ADIB trading day from October 05, 2010 to December 31, 2013. We will use three different types of data as input variables of NN. These types are:

- Type 1: the original six time series include daily open price, daily highest price, daily lowest price, daily closing price, daily volume and daily turnover of the previous period (Figure 1).

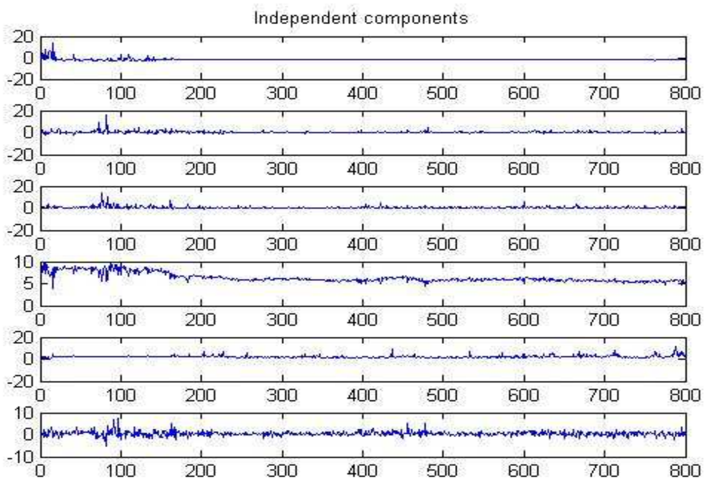
- Type 2: the six ICs obtained by applying Gradient algorithm to original time series (Figure 2).
- Type 3: the six ICs obtained by applying FastICA algorithm to original time series (Figure 3).

Figure 1: original data of ADIB from October 05, 2010 to December 31



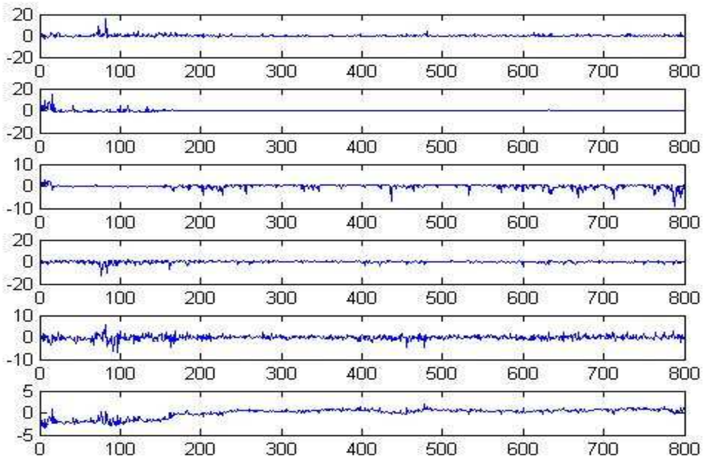
Source: own elaboration

Figure 2. Analyzed data of ADIB from October 05, 2010 to December 31, 2013 using the Gradient algorithm



Source: own elaboration

Figure 3. Analyzed data of ADIB from October 05, 2010 to December 31, 2013 using the FastICA algorithm



Source: own elaboration

Using the previous different types of input data, we obtain three different prediction models (Original-NN, Gradient-NN and FastICA-NN) respectively. A three-layer Back Propagation neural network which contains input layer, one hidden layer, and output layer is chosen in this study.

The performance is evaluated by using the following performance measures: the root mean square error (RMSE), the normalized mean square error (NMSE), the prediction error (PE) and the correlation coefficient (R). The smaller RMSE, NMSE and PE values and the larger R value represent the less deviation, that is, the best performance. Table 1 illustrates the empirical results of those three different models.

Table 1. The ADIB closing price forecasting results

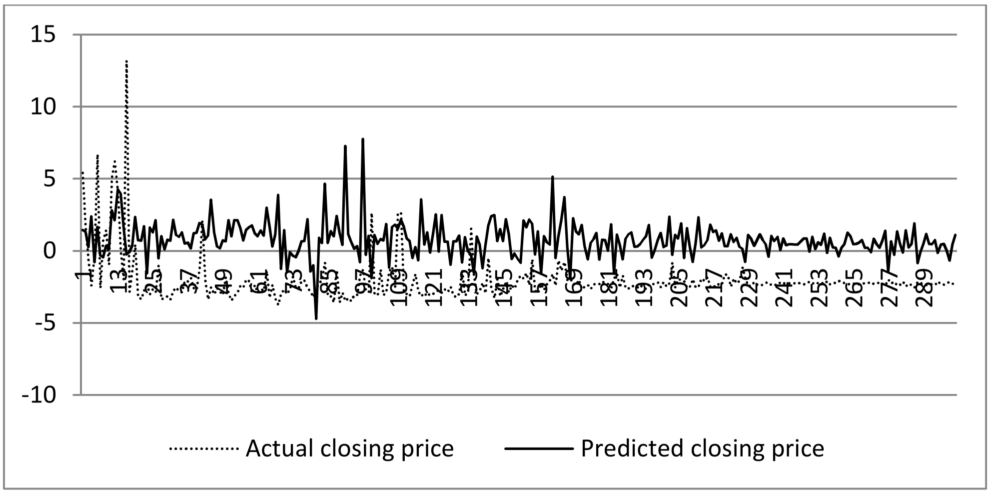
Algorithm	RMSE	NMSE	PE	R
Original-NN	0.98902	0.25287	0.13325	0.59253
Gradient-NN	0.23536	0.09855	0.09899	0.75547
FastICA-NN	**0.09271**	**0.01448**	**0.07375**	**0.96036**

Source: own elaboration

From table 1we can observe that the FastICA-NN model have smallest values of RMSE, NMSE, PE and have a largest R from other models. Thus, the FastICA-NN model can produce lower prediction error and higher prediction accuracy of the closing price forecasting. Thus, we can summarize that the FastICA algorithm outperforms the Gradient algorithm in analyzing time series data.

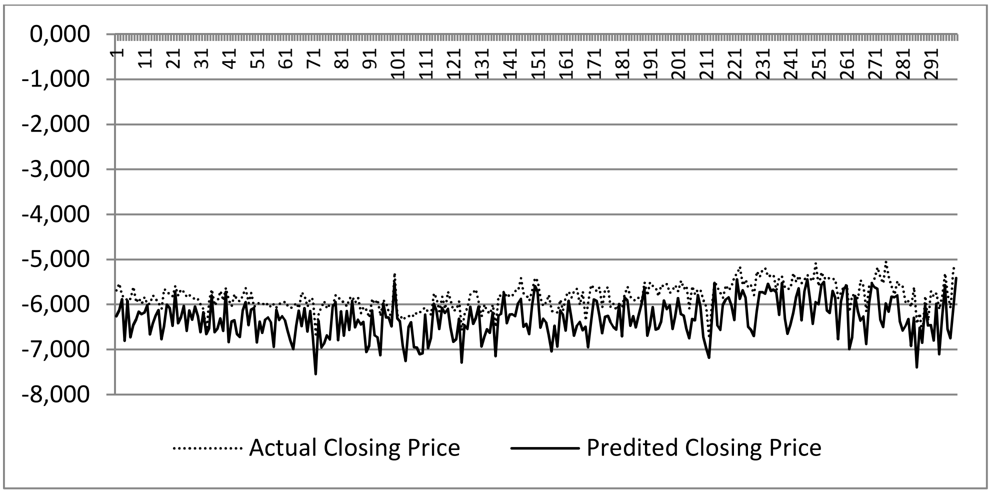
Figures 4 - 6 represent the actual daily ADIB closing price and the predicted values of the Original-NN, Gradient-NN and FastICA-NN Models respectively.

Figure 4. The actual daily ADIB closing price and the predicted values of the Original-NN model



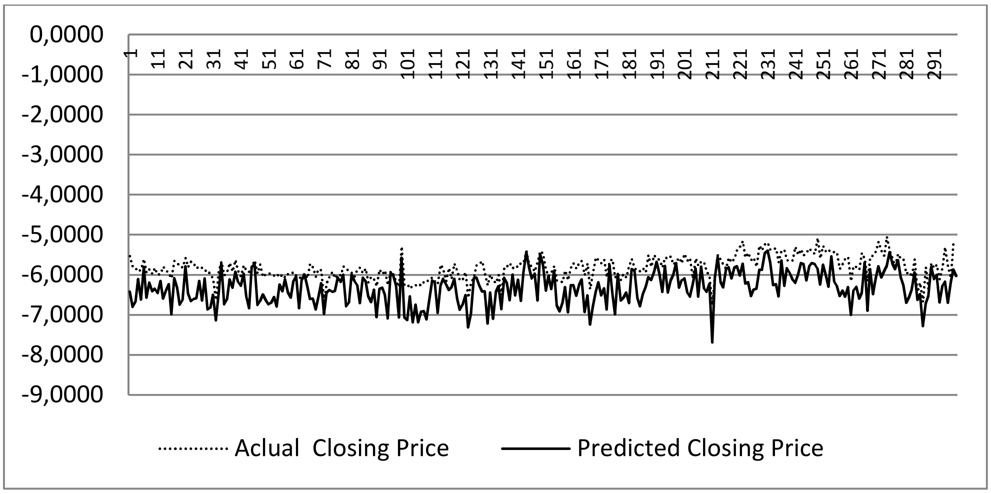
Source: own elaboration

Figure 5. The actual daily ADIB closing price and the predicted values of the Gradient-NN model



Source: own elaboration

Figure 6: The actual daily ADIB closing price and the predicted values of the FastICA-NN model



Source: own elaboration

CONCLUSION

Recently, ICA has been widely applied to financial time series analysis. Estimation the independent components (ICs) is a difficult task. Some highly successful new algorithms with various approaches for solving the ICA model were introduced by several research groups. Stock market forecasting has been one of the biggest challenges to the scientific community. Artificial Neural network (ANN) technique is regarded as more suitable for stock market forecasting than other techniques. Since ICs are statistically independent from each other, we can use them as an input of neural network that can be used to forecasts of the stock market.

In this paper we proved that a fast fixed point algorithm known as FastICA algorithm depending on maximization the nongaussianity using the negentropy approach is better than Gradient algorithm for solving ICA model. The Abu Dhabi Islamic Bank (ADIB) used as illustrative example to evaluate the performance of these two algorithms. In empirical study we use the Independent Component Analysis (ICA) as a preprocessing to forecast the stock market. Experimental results show that FastICA is more robust and faster than Gradient algorithm in stock market analysis.

REFERENCES

Belouchrani A., Abed Meraim K., Cardoso J.F. and Moulines E. (1996) A blind source separation technique using second order statistics, IEEE Trans. Signal Processing, vol. 45, 434–444.

Cardoso J.F., Souloumiac A. (1993) Blind beam forming for non-Gaussian signals, IEEE Proceedings-F, 140(6), 362-370.

Choi S., Cichocki A., Deville Y. (2001) Differential decorrelation for nonstationary source separation, [in:] Proceedings of 3rd International Conference on Independent Component Analysis and Blind Signal Separation, (San Diego, California, USA), 319-322.

Comon P. (1994) Independent component analysis — a new concept?, Signal Processing, 36, 287–314.

Hyvarinen A. (1997) A family of fixed-point algorithms for independent component Analysis, [in:] Proc. ICASSP, Munich, Germany, April 20-24, 3917–3920.

Hyvärinen A. (1999) Fast and Robust Fixed-Point Algorithms for Independent Component Analysis, IEEE Trans. on Neural Networks, 10(3), 626-634.

Hyvarinen A. (1999) Survey of Independent Component Analysis, Neural Computing Surveys, 2, 150, 1-36, 94-128.

Hyvarinen A., Oja E. (1997) A fast fixed-point algorithm for independent component analysis, Neural Computation, 9(7), 1483–1492.

Hyvärinen A., Oja E. (2000) Independent Component Analysis: Algorithms and Applications, Neural Networks, 13(4-5), 411-430.

Hyvärinen A., Karhunen J. and Oja E. (2001) Independent Component Analysis, John Wiley & Sons, New York.

Jones M., Sibson R. (1987) What is projection pursuit? Journal of the Royal Statistical Society, Ser., A150, 1-36.

Jutten C. (2000) Source separation: from dust till dawn, [in:] Proc. 2nd Int. Workshop on Independent Component Analysis and Blind Source Separation (ICA’ 2000), 15-26.

Jutten C., Herault J. (1991) Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture, Signal Processing, 24, 1–10.

Pham D.T., Cardoso J. F. (2000) Blind separation of instantaneous mixtures of nonstationary sources, in Proc. ICA, (Helsinki, Finland), 187–192.

Pham D.T., Garat P. (1997) Blind separation of mixture of independent sources through a quasimaximum likelihood approach, IEEE Trans. Signal Processing, vol. 45, 1712–1725.