A COMPARATIVE STUDY FOR ICA MULTIUNIT ALGORITHMS

Doru CONSTANTIN[1], Emilia CLIPICI[2] and Alina-Florentina ȘTEFAN[3]

[1,3]Department of Mathematics-Informatics, University of Pitesti,
Street Targu din Vale, No.1, Pitesti, Romania
[2]Department of Finance, Accounting and Economics, University of Pitesti,
Street Targu din Vale, No.1, Pitesti, Romania

ABSTRACT

We present the comparative study of convergence for multiunit algorithms based on negentropy function for estimating the independent components.

KEYWORDS

Independent Component Analysis (ICA),Blind Source Separation (BSS), Signal Processing, Negentropy function

1. INTRODUCTION

A fundamental problem in neural network research, as well as in many other disciplines, is finding a suitable representation of multivariate data, random vectors. For reasons of computational and conceptual simplicity, the representation is sought as a linear transformation of the original data. In other words, each component of the representation is a linear combination of the original variables. Well known linear transformation methods include principal component analysis, factor analysis, and projection pursuit. Independent component analysis is a recently developed method in which the goal is to find a linear representation of non-Gaussian data so that the components are statistically independent, or as independent as possible [9,7]. Such a representation seems to capture the essential structure of the data in many applications, including feature extraction and signal separation.

2. NEGENTROPY FUNCTION FOR ONE-UNIT ALGORITHMS

The negentropy function is a measure of the nongaussianity and is defined based on the entropy function. The entropy function H of a random vector y with density function $p_y(\eta)$ have the expression:

$$H(y) = -\int p_y(\eta) \log p_y(\eta) \tag{1}$$

Dhinaharan Nagamalai et al. (Eds) : AI, CSTY, SIGI - 2017
pp. 01– 09, 2017. © CS & IT-CSCP 2017 DOI : 10.5121/csit.2017.71801

A fundamental result of information theory is that a gaussian variable has the largest entropy among all random variables of equal variance [3,7]. This means that entropy could be used as a measure of nongaussianity.

To obtain a measure of nongaussianity that is zero for a gaussian variable and always nonnegative, one often uses a normalized version of differential entropy, called negentropy. Negentropy J is defined as follows:

$$J(y) = H(y_{gauss}) - H(y) \tag{2}$$

where y_{gauss} is a gaussian random variable of the same correlation (and covariance) matrix as y .

Negentropy approximations

There are some approximations of the negentropy function used in practical applications. The classic method of approximating negentropy is using higher-order cumulants:

$$J(y) \approx \frac{1}{12} E\{y^3\}^2 + \frac{1}{48} kurt(y)^2 \tag{3}$$

where y is assumed to be of zero mean and unit variance.

Another approximation is based on two nonquadratic functions G^1 and G^2 so that G^1 is odd and G^2 is even, and we obtain:

$$J(y) \approx k_1(E\{G^1(y)\})^2 + k_2(E\{G^2(y)\} - E\{G^2(\nu)\})^2), \tag{4}$$

where k_1 and k_2 are positive constants, ν is a gaussian variable of zero mean and unit variance and y is assumed to have zero mean and unit variance [6,7,9].

In the case where we use only one nonquadratic function G , the approximation becomes:

$$J(y) \approx [E\{G(y)\} - E\{G(\nu)\}]^2 \tag{5}$$

The gradient algorithm

Taking the gradient of the approximation of negentropy in (5)with respect to w and taking the normalization $E\{(w^T z)^2\} = \quad w^2 = 1$ we obtain:

$$\Delta w \propto \gamma E\{zg(w^T z)\} \tag{6}$$

$$w \leftarrow \frac{w}{w} \tag{7}$$

where $\gamma = E\{G(w^T z)\} - E\{G(\nu)\}$ and ν being a standardized gaussian random variable. For function g we may use:

$$g_1(y) = \tanh(a_1 y) \tag{8}$$

$$g_2(y) = y\exp(-\frac{y^2}{2}) \tag{9}$$

$$g_3(y) = y^3 \tag{10}$$

where $1 \le a_1 \le 2$ is a constant.

The algorithm for one independent component estimation

1. Data centering (make its mean zero).

2. Data preprocessing (whitening data) and obtain z.

3. Choose an initial value for w of unit norm and an initial value for γ.

4. Update scheme by

$$\Delta w \propto \gamma z g(w^T z),$$

where the function g is defined in (8), (9), (10).

5. Normalize the vector w by:

$$w \leftarrow \frac{w}{w}.$$

6. If the sign of γ is not known a priori, update

$$\Delta\gamma \propto (G(w^T z) - E\{G(\nu)\}) - \gamma.$$

7. If the algorithm not converged, go back to Step 4.

The fixed-point algorithm for ICA model estimation

From the gradient method in (6) we may establish the following fixed-point iteration:

$$w \leftarrow E\{zg(w^T z)\} \tag{11}$$

After rewriting the (11) relation we have:

$$w = E\{zg(w^T z)\} \Leftrightarrow (1+\alpha)w = E\{zg(w^T z)\} + \alpha w \tag{12}$$

According to the Lagrange conditions $E\{G(w^T z)\}$ under the constraint $E\{w^T z\} = \quad w^2 = 1$ are obtained at points where the gradient of the Lagrangian is zero:

$$E\{zg(w^T z)\} + \beta w = 0 \tag{13}$$

Now let us try to solve this equation by Newton's method, which is equivalent to finding the optima of the Lagrangian by Newton's method. Denoting the function on the left-hand side of (13) with F , we obtain its gradient:

$$\frac{\partial F}{\partial w} = E\{zz^T g^{'}(w^T z)\} + \beta I \tag{14}$$

Apply a reasonable approximation:

$E\{zz^T g^{'}(w^T z)\} \approx E\{zz^T\}E\{g'(w^T z)\} = E\{g^{'}(w^T z)\}I$. Thus we obtain the following approximative Newton iteration:

$$w \leftarrow w - \frac{E\{zg(w^T z)\} + \beta w}{E\{g^{'}(w^T z)\} + \beta} \tag{15}$$

This algorithm can be further simplified by multiplying both sides of (16) with $\beta + E\{g^{'}(w^T z)\}$. This gives the following form:

$$w \leftarrow E\{zg(w^T z)\} - E\{g^{'}(w^T z)\}w \tag{16}$$

This is the basic fixed-point iteration in FastICA.

The FastICA algorithm for estimating one independent component

1. Data centering.
2. Data preprocessing and obtain z .
3. Choose an initial value for vector w of unit norm.
4. Apply the updating rule:

$$w \leftarrow E\{zg(w^T z)\} - E\{g^{'}(w^T z)\}w,$$

where function g is defined in (8), (9), (10).

5. Normalize the vector w :

$$w \leftarrow \frac{w}{w}.$$

6. If the algorithm not converge, come back to 4.

3. MULTI-UNIT ALGORITHMS FOR ICA MODEL ESTIMATIN

It is possible to find more independent components by running an one-unit algorithm many times and using different initial points but with the property like the vectors w_i corresponding to different independent components are orthogonal in the whitened space [6,7,9,13].

3.1. The IC's estimation by deflationary orthogonalization

For deflationary orthogonalization is using the GramSchmidt method. This means that we estimate the independent components one by one and alternate the following steps:

1. Set the desired number of ICs to estimate m and initialization $p = 1$.
2. Initialize w_p.
3. Do an iteration of a one-unit algorithm and obtain w_p.
4. Do orthogonalization transformation:

$$w_p \leftarrow w_p - \sum_{j=1}^{p-1} (w_p^T w_j) w_j \qquad (17)$$

5. Normalize the vector w_p:

$$w \leftarrow \frac{w}{w}.$$

6. if w_p has not converged back to step 3.
7. Set $p \leftarrow p+1$. If p is not greater than m back to step 2.

3.2. The IC's estimation by symmetric orthogonalization

In this case the vectors w_i are estimated in parallel, not estimated one by one. Thus the symmetric orthogonalization methods enable parallel computation of ICs. The general form of this algorithm is:

1. Set the desired number of ICs to estimate m.
2. Initialize $w_i, i = 1,...,m$.
3. Do an iteration of a one-unit algorithm on every w_i in parallel scheme.
4. Do a symmetric orthogonalization of the matrix $W = (w_1,...,w_m)^T$.
5. If w_p not converged back to step 3.

The symmetric orthogonalization of W can be accomplished by:

$$W \leftarrow (WW^T)^{-1/2} W \qquad (18)$$

The inverse square root $(WW^T)^{-1/2}$ is obtained from the eigenvalue decomposition of $WW^T = Ediag(d_1,...,d_m)E^T$:

$$(WW^T)^{-1/2} = Ediag(d_1^{-1/2},...,d_m^{-1/2})E^T \qquad (19)$$

A simpler alternative is the following iterative algorithm:

1. Calculate $W \leftarrow W/\quad W$.
2. Calculate $W \leftarrow 3/2W - 1/2WW^T W$.
3. If the matrix WW^T is not close enough to identity matrix then go to step 2.

4. EXPERIMENTAL RESULTS FOR CONVERGENCE OF THE MULTI-UNIT ALGORITHMS

By using the FastICA algorithm we can determine the components independent and was considered the estimate of the independent components problem of a mixture of signals. The original signals are obtained from the mixing matrix signals. For estimate de ICA model we have two multi-unit algorithms: the algorithm based on the deflationary orthogonalization and the algorithm based on the symmetric orthogonalization. In the experimentally applications we choose the following nonlinear functions for function g used in the algorithms:

1. default function $g(u) = u^3$.
2. function tanh $g(u) = tanh(u)$.
3. function gauss $g(u) = u * exp(-u^2/2)$.
4. function $g(u) = u^2$.

To compare convergence for the two types of approaches, by deflating and symmetrically transformation, using the four functions mentioned above, was considered for example the following mixing matrix form:

$$A = \begin{pmatrix} 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 \\ 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 \\ 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 \\ 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 \\ 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 \\ 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 \\ 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 \\ 1 & 2 & 3 & 1 & 2 & 3 & 1 & 2 \end{pmatrix} \qquad (20)$$

The application establish the seven independent components approximation of the original signals and the convergence is shown in the next table by average of the iterations number:

Table 1. The mean number of steps for convergence.

No. item	Function	Symmetric	Deflationary
1.	$g(u) = u^3$	83 steps	12-8-8-5-5-5-2
2.	$g(u) = \tanh(u)$	18 steps	16-14-14-10-5-4-2
3.	$g(u) = u * \exp(-u^2 / 2)$	16 steps	12-8-16-21-17- -
4.	$g(u) = u^2$	17 steps	14-13-16-26- - -

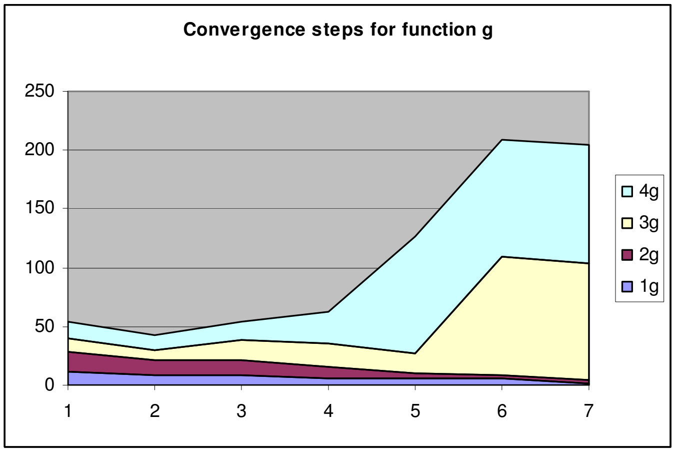

Figure 1. Convergence results for versions of function g

From above table that presents the number of steps of convergence multi-unit algorithms with symmetric and deflationary orthogonalization note that for the algorithm based on the symmetric orthogonalization the function of type 3, 4 and 1 produce a suitable results of convergence expressed through number of steps, and for the algorithm based on the deflationary orthogonalization the function of type 1 and 2 produce a suitable results of convergence. In case of IC's estimation by deflationary orthogonalization algorithm we note a high complexity to estimate the last two or three independent components for $g(u) = u * \exp(-u^2 / 2)$ and $g(u) = u^2$.

5. CONCLUSIONS

For estimating the independent components was used the negentropy function like a contrast function. By using the negentropy we may derive the updating rule for ICA estimation and obtain the general gradient one-unit algorithm, the fastica algorithm and the multi-unit algorithms based on the symmetric and deflationary orthogonalization. For the multi-unit algorithms based on the negentropy function and the symmetric and deflationary orthogonalization were established the experimental results that illustrating the performance of original signals recognition in terms of convergence.

REFERENCES

[1] C.M. Bishop, Neural Network for Pattern Recognition, Clarendon Press, 1995.

[2] A. Cichocki, R. Unbehauen, Neural Networks for Signal Processing and Optimization, Wiley, 1994.

[3] D., Constantin, L., State, A Comparative Analysis on a Class of Numerical Methods for Estimating the ICA Model, Proc. International Conference on Computers, Communications & Control, 2008.

[4] D., Constantin, L., State, A Version of the FastICA Algorithm Based on the Secant Method combined with Simple Iterations Method, ICISP 2008, Image and Signal Processing, LNCS, Springer, 2008.

[5] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley, 1991.

[6] R. Gonzalez, P. Wintz, Digital Image Processing, Addison-Wisley, 1987.

[7] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, John Wiley-Sons, 2001.

[8] T.W. Lee, Independent Component Analysis - Theory and Applications, Kluwer, 1998.

[9] J.V. Stone, Independent Component Analysis: A Tutorial Introduction, Mit Press, 2004.

[10] K.I. Diamantaras, S.Y. Kung, Principal Component Neural Networks: Theory and Applications, Wiley, 1996.

[11] Daniela Dănciulescu, Mihaela Colhon, Gheorghe Grigoraş, A System of Knowledge Representation for Right Linear Grammars Generation, Broad Research in Artificial Intelligence and Neuroscience, vol 8(1), ISSN 2068-0473, E-ISSN 2067-3957, pp. 42-51, 2017.

[12] Cosmin Sabo, Petrică C. Pop, Honoriu Vălean, Daniela Dănciulescu, An Innovative Approach to Manage Heterogeneous Information Using Relational Database Systems, Proceedings of International Conference on Intelligent Systems Design and Applications, ISDA 2016, pp 1-10, 2016.

[13] Viorel Negru, Gheorghe Grigoraş, Daniela Dănciulescu, Natural Language Agreement in the Generation Mechanism based on Stratified Graphs, Proceedings of the 7th Balkan Conference in Informatics (BCI 2015), Craiova, Romania, 2015.

[14] Daniela Dănciulescu, Formal Languages Generation in Systems of Knowledge Representation based on Stratified Graphs, INFORMATICA 2015, vol. 26, no. 3, pp. 407-417, ISSN 0868-4952, 2015.

[15] Daniela Dănciulescu, Mihaela Colhon, Systems of knowledge representation based on stratified graphs. Application to Natural Language Generation, Carpathian Journal of Mathematics, 32(1), pp. 49-62, 2014.

[16] Daniela Dănciulescu, Mihaela Colhon, Splitting the structured paths in stratified graphs. Application in Natural Language Generation, Analele ştiinţifice ale Universităţii Ovidius Constanţa, Seria Matematică, vol. 22, no. 2, pp. 59-69, ISSN: 1224-1784, 2014.

[17] Daniela Dănciulescu, Nicolae Tăndăreanu, Splitting the structured paths in stratified graphs, Modeling and Development of Intelligent Systems, Proceedings of the 3th Int Conference on Modeling and Development of Intelligent Systems, ISSN 2067-3965, 2014.

[18] Daniela Dănciulescu, Systems Of Knowledge Representation Based On Stratified Graphs And Their Inference Process, 9th International Conference of Applied Mathematics, 2013.

[19] Nicolae Tăndăreanu, Irina Tudor (Preda), Daniela Dănciulescu, Applications of stratified graphs in optimal planning, Proceedings of the 12th Conference on Artficial Intelligence and Digital Communications (AIDC), pp.7-24, 2012.

[20] W. Gardner, Introduction to Random Processes with Applications to Signal and Systems, Macmillan, 1986.

[21] M. Girolami, Self-Organising Neural Networks - Independent Component Analysis and Blind Source Separation, Springer Verlag, 1999.

[22] D.A., Popescu, N., Bold, O., Domsa, Generating assessment tests with restrictions using genetic algorithms, 12th IEEE International Conf. on Control & Automation (IEEE ICCA 2016), 1-3 June, (2016) Kathmandu, Nepall.

[23] I.T. Jolliffe, Principal Component Analysis, Springer-Verlag New York, 2002.

[24] Oppenheim, R. Schafer, Discrete-Time Signal Processing, Prentince Hall, 1989.

[25] S. Russell, P. Norvig, Artificial Intelligence. A Modern Approach, Prentice Hall, 1995.