Available online at www.sciencedirect.com
SCIENCE DIRECT®

Neurocomputing 64 (2005) 529–536

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Letters

A fast fixed-point algorithm for complexity pursuit

Zhenwei Shi[a,b,*], Huanwen Tang[a], Yiyuan Tang[b,c,d]

[a]Institute of Computational Biology and Bioinformatics, Dalian University of Technology, Dalian 116023, P.R. China
[b]Institute of Neuroinformatics, Dalian University of Technology, Dalian 116023, P.R. China
[c]Laboratory of Visual Information Processing, The Chinese Academy of Sciences, Beijing 100101, P.R. China
[d]Key Lab for Mental Health, The Chinese Academy of Sciences, Beijing 100101, P.R. China

Received 1 November 2004; received in revised form 27 November 2004; accepted 28 November 2004
Communicated by R.W. Newcomb
Available online 19 January 2005

Abstract

Complexity pursuit is a recently developed algorithm using the gradient descent for separating interesting components from time series. It is an extension of projection pursuit to time series data and the method is closely related to blind separation of time-dependent source signals and independent component analysis (ICA). In this paper, a fixed-point algorithm for complexity pursuit is introduced. The fixed-point algorithm inherits the advantages of the well-known FastICA algorithm in ICA, which is very simple, converges fast, and does not need choose any learning step sizes.

Keywords: Independent component analysis; Blind source separation; Complexity pursuit; Projection pursuit; Time series

*Corresponding author. Institute of Computational Biology and Bioinformatics, Dalian University of Technology, Dalian 116023, P.R. China.

E-mail addresses: szw1977@yahoo.com.cn (Z. Shi), yy2100@163.net (Y. Tang).

0925-2312/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2004.11.028

1. Introduction

Independent component analysis (ICA) has been widely applied to blind source separation, blind deconvolution, and feature extraction, and so on. The model of ICA consists of mixing independent random variables, usually linearly [1,2,4,5,9,11,13,18,19,21,22]. In many applications, however, what is mixed is not random variables but time signals, or time series. ICA in its basic form ignores any time structure and uses only the nongaussianity criteria. It is to be noted that under certain restrictions, it is also possible to estimate the independent components using the time-dependency information alone [3,15,16,20]. However, a principled way of combining both of these estimation criteria (nongaussianity and time-correlations) has been introduced by Hyvärinen in the complexity pursuit algorithm [10]. Complexity pursuit is an extension of projection pursuit [6] to time series, that is, signals with time structure. The goal is to find projections of time series that have interesting structure, defined using criteria related to Kolmogoroff complexity [17] or coding length. Time series which have the lowest coding complexity are considered the most interesting. Hyvärinen derived a simple approximation of Kolmogoroff complexity that takes into account both the nongaussianity and the autocorrelations of the time series. He developed a gradient ascent algorithm for its approximative optimization. The method is closely related to blind separation of time-dependent source signals and ICA.

In this paper, motivated by the work of Hyvärinen, we propose a fixed-point algorithm for complexity pursuit. The fixed-point algorithm inherits the advantages of the well-known FastICA algorithm [9,11] in ICA.

2. Complexity pursuit

Assume that the observed data are multivariate time series $x(t)$, that is, a vector of time signals. The basic idea in the complexity pursuit is to find projections $w^T x(t)$ such that the Kolmogoroff complexity of the projection is minimized. We search for projections that can be easily coded in the complexity pursuit. This is a general-purpose measure and is probably connected to information-processing principles used in the brain [10].

First, we derive an approximation of the Kolmogoroff complexity of a scalar signal $y(t)(t = 1, \ldots, T)$ along similar lines as Hyvärinen [10]. For simplicity, the signal is assumed to have zero mean and unit variance.

We consider predictive coding of the signal. The value $y(t)$ is predicted from the preceding values by some function to be specified:

$$\hat{y}(t) = f(y(t-1), \ldots, y(1)). \tag{1}$$

To code the actual value $y(t)$, the residual

$$\delta y(t) = y(t) - \hat{y}(t) \tag{2}$$

is coded by a scalar quantization method. According to the basic principles of information theory, the length of this code is asymptotically approximated by the sum of the entropies H of the residuals. The coding complexity can be approximated by [10]

$$\hat{K}(y) = \sum_t H(\delta y(t)). \tag{3}$$

Assuming that the residual is stationary and ergodic and that the predictor uses a history of bounded length, and ignoring border effects, we have the simpler version [10]:

$$\hat{K}(y) = TH(\delta y), \tag{4}$$

where δy denotes a random variable with the marginal distribution of the residual. Note that we made the assumption here that the signal is stationary.

To use the approximation in Eq. (4) in practice, we need to fix the structure of the predictor f and find an approximation of the entropy of δy. We use a computationally simple predictor structure, given by a linear autoregressive model:

$$\hat{y}(t) = \sum_{\tau > 0} \alpha_\tau y(t - \tau). \tag{5}$$

To approximate the entropy of δy, we adopt a simpler method here, which is possible by assuming that we have prior knowledge of the distribution of the residual (in particular, in many cases we can assume that the residuals are supergaussian [10]). We assume that we know a good approximation of the (negative) logarithm of the probability density of the residual, denoted by G. Then we obtain the approximation:

$$H(\delta y) \approx E\{G(\delta y)\}. \tag{6}$$

In ICA, if the signals to be reconstructed satisfy certain properties, an exact form of the contrast function is not required in order to achieve the desired estimation results [4,10,12]. We may therefore optimistically assume that the exact form of the function G is not very important here either, as long as it is qualitatively similar enough [10].

To find the 'most interesting' directions w, use the above approximation of complexity for $y(t) = w^T x(t)$. Note that the values of α_τ are function of w only. Thus, we can express the approximation of complexity as a contrast function of w only:

$$\hat{K}(w^T x(t)) = E\{G(w^T(x(t) - \sum_{\tau > 0} \alpha_\tau(w) x(t - \tau)))\}, \tag{7}$$

then we can use an algorithm to find the minima of the approximation of complexity. To begin, we can simplify the algorithm by first whitening the zero mean data $x(t)$, for example, by

$$\tilde{x}(t) = Vx(t) = (E\{x(t)x(t)^T\})^{-1/2} x(t). \tag{8}$$

This is a well-known preprocessing step in ICA [9–11]. This implies that the constraint of unit variance of $w^T x(t)$ can be replaced by the constraint of unit norm of w. Thus, we can estimate the 'most interesting' directions w by minimizing, for whitened data $\tilde{x}(t)$, the following contrast function:

$$\min_{\|w\|^2=1} \hat{K}(w^T\tilde{x}(t)) = E\left\{ G(w^T(\tilde{x}(t) - \sum_{\tau>0} \alpha_\tau(w)\tilde{x}(t-\tau))) \right\}. \tag{9}$$

Remark. It should be noted that the contrast function derived here simplifies the estimation procedure of the original complexity pursuit. Using the contrast function, we can derive the same original gradient descent complexity pursuit algorithm. But we would develop a fixed-point algorithm for complexity pursuit, which is one of the most powerful learning algorithms and has advantages as compared to other gradient-based algorithms often used in neurocomputing.

3. A fixed-point algorithm for complexity pursuit

To perform the optimization in (9), we can use a fixed-point iteration along similar lines as the FastICA algorithm for maximizing nongaussianity [9,11]. The fixed-point algorithm can be found using an approximative Newton method. Denote by

$$z(w) = \tilde{x}(t) - \sum_{\tau>0} \alpha_\tau(w)\tilde{x}(t-\tau). \tag{10}$$

According to the Kuhn–Tucker conditions [14], the optima of $\hat{K}(w^T\tilde{x}(t)) = E\{G(w^Tz(w))\}$ under the constraint $\|w\|^2 = 1$ are obtained at points where

$$E\{z(w)g(w^Tz(w))\} + E\left\{\frac{\partial z(w)}{\partial w} wg(w^Tz(w))\right\} - \beta w = 0, \tag{11}$$

where β is some constant and the function g is a derivative of G. Similar to the analysis of Hyvärinen [10], note that the quantity $\partial z(w)/\partial w\, w$ depends on only the past values of $w^T\tilde{x}(t)$. Therefore, it is independent of the residual $w^Tz(t)$, which has the role of the innovation process here. Thus, the second term in (11) disappears and we have

$$E\{z(w)g(w^Tz(w))\} - \beta w = 0. \tag{12}$$

Now let us try to solve this equation by Newton's method. Denoting the function on the left-hand side of (12) by F, we obtain its Jacobian matrix $JF(w)$ as

$$\begin{aligned} JF(w) = {} & E\left\{\frac{\partial z(w)}{\partial w} g(w^Tz(w))\right\} + E\left\{\frac{\partial z(w)}{\partial w} wg'(w^Tz(w))z(w)^T\right\} \\ & + E\{z(w)z(w)^Tg'(w^Tz(w))\} - \beta I. \end{aligned} \tag{13}$$

Similar to the above analysis again, the first and the second term in (13) disappear as well and we obtain

$$JF(w) = E\{z(w)z(w)^Tg'(w^Tz(w))\} - \beta I. \tag{14}$$

To simplify the inversion of this matrix, we decide to approximate the first term in (14). First, assume that $x(t)$ and $s(t)$ follow the ICA mixing model: $x(t) = As(t)$. After the data $x(t)$ are whitened, we have $\tilde{x}(t) = \tilde{A}s(t)$, where the new mixing matrix $\tilde{A}$ is orthogonal [11]. Then the innovation processes $\tilde{s}(t) = s(t) - \sum_{\tau>0} \alpha_\tau s(t-\tau)$ follow the model as well (assume that the innovation process has zero mean and unit variance) [8], i.e. $z(w) = \tilde{A}\tilde{s}(t)$. We obtain

$$E\{z(w)z(w)^T\} = \tilde{A}E\{\tilde{s}(t)\tilde{s}(t)^T\}\tilde{A}^T = \tilde{A}\tilde{A}^T = I. \tag{15}$$

Since the data are whitened, a reasonable approximation seems to be

$$\begin{aligned} &E\{z(w)z(w)^T g'(w^T z(w))\} \\ &\quad \approx E\{z(w)z(w)^T\}E\{g'(w^T z(w))\} = E\{g'(w^T z(w))\}I. \end{aligned} \tag{16}$$

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

$$w \leftarrow w - \frac{[E\{z(w)g(w^T z(w))\} - \beta w]}{[E\{g'(w^T z(w))\} - \beta]} \tag{17}$$

$$w \leftarrow w/\|w\|. \tag{18}$$

This algorithm can be further simplified by multiplying both sides of (17) by $\beta - E\{g'(w^T z(w))\}$. This gives, after straightforward algebraic simplification

$$w \leftarrow E\{z(w)g(w^T z(w))\} - E\{g'(w^T z(w))\}w \tag{19}$$

$$w \leftarrow w/\|w\|. \tag{20}$$

Thus, the fixed-point iteration in Eq. (9) can be obtained as:

$$\begin{aligned} w \leftarrow {} & E\{(\tilde{x}(t) - \sum_{\tau>0} \alpha_\tau(w)\tilde{x}(t-\tau))g(w^T(\tilde{x}(t) - \sum_{\tau>0} \alpha_\tau(w)\tilde{x}(t-\tau)))\} \\ & - E\{g'(w^T(\tilde{x}(t) - \sum_{\tau>0} \alpha_\tau(w)\tilde{x}(t-\tau)))\}w \end{aligned} \tag{21}$$

$$w \leftarrow w/\|w\|. \tag{22}$$

The function g should be chosen as in ordinary ICA, but according to the probability distribution of the residual [10]. If the residual is supergaussian, $g(u) = \tanh(au)$ is suitable, where $a \geqslant 1$ is a constant [2,9,10]. For subgaussian residuals, one could use $g(u) = u - \tanh(u)$ [7] or $g(u) = u^3$, for example. For almost gaussian residuals, a linear g could be used [10].

Thus, after the data $x(t)$ are whitened, the complexity pursuit algorithm is then as follows. At every step, first estimate the autoregressive constants $\alpha_\tau(w)$ in Eq. (5) for the time series given by $w^T\tilde{x}(t), t = 1, \ldots, T$. Then do the fixed-point iteration in (21) and the normalization in (22) (such quantities are for the current estimate of w in each iteration step).

To estimate several projections, one can simply use a deflation scheme (Gram–Schmidt orthogonalization scheme) [9,11].

A simple special case of the method is obtained when the autoregressive model has just one predicting term [10]:

$$\hat{y}(t) = \alpha_1 y(t-1). \tag{23}$$

The lag need not be equal to 1, but this is the basic case. The parameter α_1 in the algorithm can then be estimated simply by a least-squares method as [10]:

$$\hat{\alpha}_1 = w^T E\{\tilde{x}(t)\tilde{x}(t-1)^T\}w. \tag{24}$$

4. Simulations

We created six signals using an AR(1) model. Signals 1, 2, 3 and 4 were created with supergaussian innovations and signals 5 and 6 with gaussian innovations; all innovations had unit variance. Signals 1, 3 and 5 had identical autoregressive coefficients (0.25) and therefore identical autocovariances; signals 2, 4 and 6 had identical coefficients (0.5) as well. The signals were mixed by a 6 × 6 random mixing matrix (denote by A) as in ICA. Sample size T was 20 000, and the error index defined as [1]:

$$E = \sum_{i=1}^{M}\left(\sum_{j=1}^{M}\frac{|p_{ij}|}{\max_k |p_{ik}|} - 1\right) + \sum_{j=1}^{M}\left(\sum_{i=1}^{M}\frac{|p_{ij}|}{\max\limits_k |p_{kj}|} - 1\right), \tag{25}$$

where p_{ij} is the ijth element of $M \times M$ matrix $P = WA$ (W is the separating matrix in ICA). Ordinary ICA methods and methods based on autocovariances would fail with these data [10]. Thus, for the goal of comparison, we tested two algorithms (complexity pursuit (GradCP) [10] and complexity pursuit using the fixed-point iteration in this paper (FastCP)) in the simulations. The nonlinearity was chosen as $g(u) = \tanh(u)$ in the two algorithms and the step size in the GradCP was taken equal to 1. The two algorithms were run for every fixed iteration N and the iteration N was varied from 5 to 60. At every trial, the two algorithms were run 100 times with different mixing matrices and the error was estimated as the average of the errors. The results are depicted in Fig. 1.

5. Conclusions

In this paper, we have presented a fixed-point algorithm for complexity pursuit. The fixed-point algorithm simplifies the estimation procedure of the original complexity pursuit and improves its convergence properties. Furthermore, in contrast to other gradient-based algorithms, the fixed-point algorithm does not need to choose any learning step sizes. This means that the algorithm is easy to use. Interestingly, assuming that the signals have no time dependencies, our method reduces to the well-known FastICA algorithm in ICA [9,11].

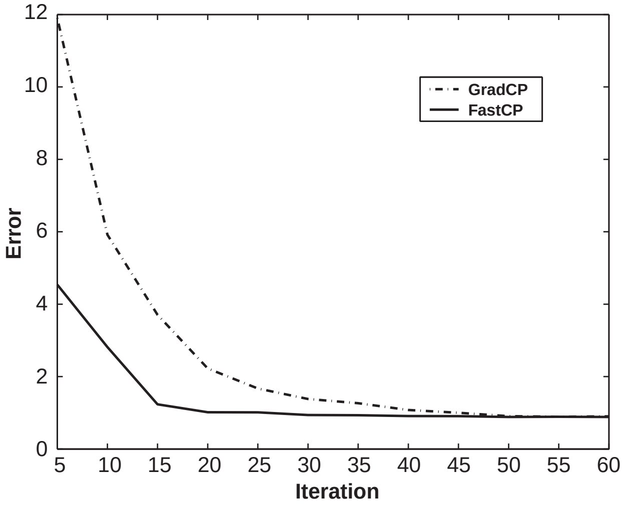

Fig. 1. Convergence of the two complexity pursuit algorithms for artificially generated data. Horizontal axis: iteration. Vertical axis: error measure as given in the text. Dot–dashed line: GradCP; Solid line: FastCP.

Acknowledgements

We are grateful to all the anonymous reviewers who provided insightful and helpful comments. The work was supported by NSFC(60472017,90103033), MOE(KP0302) and MOST(ICPDF2003).

References

[1] S.-I. Amari, A. Cichocki, H. Yang, A new learning algorithm for blind source separation, in: Advances in Neural Information Processing System, vol. 8, MIT Press, Cambridge, MA, 1996, pp. 757–763.

[2] A. Bell, T. Sejnowski, An information–maximization approach to blind separation and blind deconvolution, Neural Computation 7 (6) (1995) 1129–1159.

[3] A. Belouchrani, K.A. Meraim, J.-F. Cardoso, E. Moulines, A blind source separation technique based on second order statistics, IEEE Trans. Signal Process. 45 (2) (1997) 434–444.

[4] J.-F. Cardoso, B.H. Laheld, Equivariant adaptive source separation, IEEE Trans. Signal Process. 44 (12) (1996) 3017–3030.

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

[6] J.H. Friedman, J.W. Tukey, A projection pursuit algorithm for exploratory data analysis, IEEE Trans. Comput. C-23 (9) (1974) 881–890.

[7] M. Girolami, An alternative perspective on adaptive independent component analysis algorithms, Neural Computation 10 (8) (1998) 2103–2114.

[8] A. Hyvärinen, Independent component analysis for time-dependent stochastic processes, in: Proceedings of the International Conference on Artificial Neural Networks (ICANN’98), Skövde, Sweden, 1998, pp. 135–140.

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

[10] A. Hyvärinen, Complexity pursuit: separating interesting components from time-series, Neural Computation 13 (4) (2001) 883–898.

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

[12] A. Hyvärinen, E. Oja, Independent component analysis by general nonlinear Hebbian-like learning rules, Signal Process. 64 (3) (1998) 301–313.

[13] C. Jutten, J. Herault, Blind separation of sources, part I: an adaptive algorithm based on neuromimetic architecture, Signal Process. 24 (1991) 1–10.

[14] D.G. Luenberger, Optimization by Vector Space Methods, Wiley, New York, 1969.

[15] K. Matsuoka, M. Ohya, M. Kawamoto, A neural net for blind separation of nonstationary signals, Neural Networks 8 (3) (1995) 411–419.

[16] L. Molgedey, H.G. Schuster, Separation of a mixture of independent signals using time delayed correlations, Phys. Rev. Lett. 72 (23) (1994) 3634–3637.

[17] P. Pajunen, Blind source separation using algorithmic information theory, Neurocomputing 22 (1998) 35–48.

[18] Z. Shi, H. Tang, Y. Tang, A new fixed-point algorithm for independent component analysis, Neurocomputing 56 (2004) 467–473.

[19] Z. Shi, H. Tang, W. Liu, Y. Tang, Blind source separation of more sources than mixtures using generalized exponential mixture models, Neurocomputing 61 (2004) 461–469.

[20] L. Tong, R.W. Liu, V. Soon, Y.-F. Huang, Indeterminacy and identifiability of blind identification, IEEE Trans. Circuits Systems 38 (5) (1991) 499–509.

[21] M. Zhong, H. Tang, H. Chen, Y. Tang, An EM algorithm for learning sparse and overcomplete representations, Neurocomputing 57 (2004) 469–476.

[22] M. Zhong, H. Tang, Y. Tang, Expectation–maximization approaches to independent component analysis, Neurocomputing 61 (2004) 503–512.