In Artificial Neural Networks and Neural Information Processing - ICANN/ICONIP 2003

LNCS 2714, Springer-Verlag, ed. by O. Kaynak, E. Alpaydin, E. Oja, L. Xu, p132-139, 2003

Dimension Reduction Based on Orthogonality —a Decorrelation Method in ICA

Kun Zhang, Lai-Wan Chan

Department of Computer Science and Engineering,
The Chinese University of Hongkong, Hongkong
{kzhang, lwchan}@cse.cuhk.edu.hk

Abstract. In independent component analysis problems, when we use a one-unit objective function to iteratively estimate several independent components, the uncorrelatedness between the independent components prevents them from converging to the same optimum. A simple and popular way of achieving decorrelation between recovered independent components is a deflation scheme based on a Gram-Schmidt-like decorrelation [7]. In this method, after each iteration in estimation of the current independent component, we subtract its 'projections' on previous obtained independent components from it and renormalize the result. Alternatively, we can use the constraints of uncorrelatedness between independent components to reduce the number of unknown parameters of the de-mixing matrix directly. In this paper, we propose to reduce the dimension of the de-mixing matrix to decorrelate different independent components. The advantage of this method is that the dimension reduction of the observations and de-mixing weight vectors makes the computation lower and produces a faster and efficient convergence.

1 Introduction

The objective of this paper is to propose a dimension reduction appoach to achieving decorrelation between independent components. In this section we review the decorrelation method currently used. In Sect. 2 we introduce our method in detail.

Let us denote by $X = (x_1, x_2, ..., x_m)^T$ a zero-mean m-dimensional variable, and $S = (s_1, s_2, ..., s_n)^T$, $n \leq m$, is its linear transform with a constant matrix W:

$$S = WX \tag{1}$$

Given X as observations, based on different assumptions, principal component analysis (PCA) and independent component analysis (ICA) both aim to estimating W and S. The goal of PCA is to find a new variable S under the orthogonal constraint $W^TW = I$ (I is the identity matrix) such that S becomes uncorrelated in components and accounts for as much as possible of the variance of the variable X [10]. While in ICA, the transformed components s_i are not only uncorrelated with each other, but also statistically as independent of each other as possible [3].

Since generally there is no closed-form solution to ICA problems, an ICA algorithm consists of two parts: an objective function (contrast function) and an optimization method used to optimize the objective function [11]. The objective function measures the independence between independent sources with the help of mutual information between them [3], entropy (negentropy) of each independent source [3, 6], or their higher-order cumulants [4, 9], etc. A multi-unit contrast function treats the problem of estimating all the independent components (the whole data model) at the same time. Or motivated by projection pursuit, we can use a one-unit contrast function whose optimization enables estimation of a single independent component [8, 11]. And this procedure can be iterated to find several independent components.

Higher-order cumulants like kurtosis, and approximations of negentropy can provide one-unit contrast functions. The contrast functions used in FastICA [9, 5, 7] are approximations of negentropy based on the maximum entropy principle [6]. These approximations are often more accurate than the cumulant-based approximations [3], and contrast functions based on approximations of negentropy are more robust than the kurtosis [8, 5]. In the simplest case, these approximations are of the form:

$$J_G(y) = [E_y\{G(y)\} - E_v\{G(v)\}]^2 \tag{2}$$

where G is a non-quadratic, sufficiently smooth function, v a standardized Gaussian random variable, y is zero-mean and normalized to unit variance. As for the optimization method, the convergence of adaptive algorithms based on stochastic gradient descent is often slow and depends crucially on the choice of the learning rate sequence. Batch algorithms based on fixed-point iteration can avoid this problem [9, 5, 7]. FastICA, a fixed-point algorithm for ICA, was firstly introduced using kurtosis in [9], and it was generalized for general contrast functions (Eq.2) in [5, 7]. The following is the FastICA algorithm for whitened data:

$$w(k) = E\{Xg(w(k-1)^T X)\} - E\{g'(w(k-1)^T X)\}w(k-1) \tag{3}$$

where w^T is a row of W and w is normalized to unit norm after each iteration, the function g is the derivative of the function G used in Eq.2. In this paper, FastICA will be used to extract an independent component from the observations.

In general 'independence' between two variables is a much stronger property than 'uncorrelatedness' between them. When we use a one-unit objective function to iteratively calculate several n independent components, in order to prevent different neurons from converging to the same optimum we must decorrelate the outputs. A simple and common way of achieving decorrelation is the deflation scheme based on Gram-Schmidt-like decorrelation [12, 9, 5, 7]. For whitened data, after we have estimated p independent components, or p weight vectors $w_1, w_2, ..., w_p$, we run the one-unit algorithm to estimate w_{p+1}. In this procedure, after each update iteration step, we subtract from updated w_{p+1} its projections on the previous estimated p vectors, $w_{p+1}^T w_j w_j, j = 1, ..., p$, i.e. let

$$w_{p+1} = w_{p+1} - \sum_{j=1}^{p} w_{p+1}^T w_j w_j, \text{ and then renormalize } w_{p+1}:$$

$$w_{p+1} = w_{p+1} / \sqrt{w_{p+1}^T w_{p+1}}$$

2 Dimension Reduction based on orthogonality

In the ICA problem, let n and m be the number of independent components and observations respectively. Generally (but not necessarily), if $n < m$, we first use PCA to extract the n-dimensional 'principal' subspace from the m-dimensional observation space, and then obtain the n independent components in this subspace. So without loss of generality, we extract n independent components given n observations with a positive definite covariance matrix in the following analysis.

In both ICA and PCA, s_i must be mutually uncorrelated, i.e. $E(s_i s_j) = E(w_i^T X X^T w_j^T) = 0$, where $i, j = 1, 2, ..., n, i \neq j$. In PCA, the scaling of each basis vector w_i^T, which is a row of W, is of unit length, i.e. $w_i^T w_i = 1$. In ICA, we can fix the scaling of the independent components to avoid the inherent scaling indeterminacy. Generally we set the variance of s_i to be 1, i.e. $E[s_i^2] = 1$. Now we have $\frac{n(n-1)}{2} + n = \frac{n(n+1)}{2}$ equations for both PCA and ICA problems.

There are n^2 parameters to be determined, which are elements of W. Therefore the PCA or ICA problem can not be solved uniquely with only these restrictions. In PCA, the current PC accounts the maximum variance in current space; and in ICA, IC's should be independent of each other (or they should be as non-Gaussion as possible). These characteristics, together with the $\frac{n(n+1)}{2}$ equations discussed above help to solve the PCA and ICA problems respectively.

The uncorrelatedness between the independent components can help us to obtain multiple independent components with a one-unit objective function. After p independent components have been obtained, we search for the $(p+1)$-th independent component which is uncorrelated with the previous p ones. With the Gram-Schmidt-like decorrelation scheme and whitened data, in each iteration step of estimating w_{p+1} we search for updated w_{p+1} in the original n-dimensional parameter space, and afterwards project the new vector onto the space which is orthogonal to the obtained p weight vectors. Intuitively, since the contrast curve of the objective function may be very complex, this scheme may do harm to the convergence of w_{p+1} to a target in this subspace.

In fact, for whitened data, w_{p+1} lies in the $(n-p)$-dimensional parameter subspace which is orthogonal to the previous p de-mixing weight vectors. Alternatively, we can search w_{p+1} in this space directly, which always guarantees the orthogonality. And in addition, compared to the Gram-Schmidt-like deflation scheme, in this way parameters needed to be estimate become fewer because the parameter dimension used for search becomes lower. Therefore we can lower the computation, and obtain a faster convergence.

2.1 Algorithm

Let's decompose W into two parts:

$$W = \widetilde{W}^{(n)} P \tag{4}$$

where P is the whitening matrix, so that $E(PXX^TP^T) = I$. Since $\widetilde{W}^{(n)}\widetilde{W}^{(n)T} = \widetilde{W}^{(n)}E[PXX^TP^T]\widetilde{W}^{(n)T} = E(SS^T) = I$, $\widetilde{W}^{(n)}$ is an orthonormal matrix. Let $\tilde{w}_i^{(n)T}$ be a row of $\widetilde{W}^{(n)}$.[1] We know:

$$\sum_{k=1}^{n} \tilde{w}_i^{(n)}(k)\tilde{w}_j^{(n)}(k) = 0, i \neq j$$

There exists at least one q such that $\tilde{w}_1^{(n)}(q)$ is not zero, so we have

$$\tilde{w}_2^{(n)}(q) = -\frac{1}{\tilde{w}_1^{(n)}(q)} \sum_{\substack{k=1,\\ k\neq q}}^{n} \tilde{w}_1^{(n)}(k)\tilde{w}_2^{(n)}(k), \text{ thus,}$$

$$\tilde{w}_2^{(n)} = \begin{pmatrix} \mathbf{I_{q-1}} & \mathbf{0_{(q-1)\times(n-q)}} \\ -\frac{\tilde{w}_1^{(n)}(1)}{\tilde{w}_1^{(n)}(q)} \cdots -\frac{\tilde{w}_1^{(n)}(q-1)}{\tilde{w}_1^{(n)}(q)} & -\frac{\tilde{w}_1^{(n)}(q+1)}{\tilde{w}_1^{(n)}(q)} \cdots -\frac{\tilde{w}_1^{(n)}(n)}{\tilde{w}_1^{(n)}(q)} \\ \mathbf{0_{(n-q)\times(q-1)}} & \mathbf{I_{n-q}} \end{pmatrix}_{n\times(n-1)} \cdot \begin{pmatrix} \tilde{w}_2^{(n)}(1) \\ \vdots \\ \tilde{w}_2^{(n)}(q-1) \\ \tilde{w}_2^{(n)}(q+1) \\ \vdots \\ \tilde{w}_2^{(n)}(n) \end{pmatrix}$$

$$\stackrel{def}{=} Aw_2^{(n-1)} \tag{5}$$

And $s_2 = \tilde{w}_2^{(n)T}PX = w_2^{(n-1)T}A^TPX = w_2^{(n-1)T}X'$, where $X' = A^TPX$. We can see that s_2 can be considered as an independent component of $(n-1)$-dimensional data X'. Let P_1 be the whitening matrix of X', we have $w_2^{(n-1)} = P_1^T\tilde{w}_2^{(n-1)}$, where $\tilde{w}_2^{(n-1)}$ is a de-mixing weight vector of the new data X' after whitening. Obviously the covariance matrix of X' is A^TA, Let $E = (e_1...e_{(n-1)})$ be the orthonormal matrix composed of eigenvectors of A^TA and $D = diag(d_1...d_{(n-1)})$ be the diagonal matrix of its eigenvalues. $P_1 = D^{-1/2}E^T$ is a whitening matrix of X'.

After the estimation of $(n-1)$-dimensional de-mixing weight vector $\tilde{w}_2^{(n-1)}$ given X' as observations with the chosen one-unit contrast function, we can construct $\tilde{w}_2^{(n)}$ and w_2 by Eq.5 and Eq.4, i.e. $\tilde{w}_2^{(n)} = Aw_2^{(n-1)} = AP_1^T\tilde{w}_2^{(n-1)}, w_2 = P^T\tilde{w}_2^{(n-1)} = P^T \cdot AP_1^T\tilde{w}_2^{(n-1)}$.

We also have $\tilde{w}_3^{(n)} = Aw_3^{(n-1)}$, and $w_3^{(n-1)} = P_1^T\tilde{w}_3^{(n-1)}$. Since $w_2^{(n-1)}$ and $w_3^{(n-1)}$ are two different de-mixing weight vectors of X', the $(n-1)$-dimensional vectors $\tilde{w}_2^{(n-1)}$ and $\tilde{w}_3^{(n-1)}$ are orthogonal. And there exists r such that $\tilde{w}_2^{(n-1)}(r) \neq 0$. In a similar way we can get $\tilde{w}_3^{(n-1)} = A_1w_3^{(n-2)}$, where $w_3^{(n-2)}$ is $(n-2)$-

[1] The superscript n indicates the dimension of de-mixing matrix.

dimensional and A_1 is a $(n-1)\times(n-2)$ matrix:

$$A_1 = \begin{pmatrix} \mathbf{I_{r-1}} & \mathbf{0_{(r-1)\times(n-r-1)}} \\ -\frac{\tilde{w}_2^{(n-1)}(1)}{\tilde{w}_2^{(n-1)}(r)} \cdots -\frac{\tilde{w}_2^{(n-1)}(r-1)}{\tilde{w}_2^{(n-1)}(r)} & -\frac{\tilde{w}_2^{(n-1)}(r+1)}{\tilde{w}_2^{(n-1)}(r)} \cdots -\frac{\tilde{w}_2^{(n-1)}(n-1)}{\tilde{w}_2^{(n-1)}(r)} \\ \mathbf{0_{(n-r-1)\times(r-1)}} & \mathbf{I_{n-r-1}} \end{pmatrix}_{(n-1)\times(n-2)} \tag{6}$$

We can see $s_3 = w_3^{(n-1)T}X' = w_3^{(n-2)T}A_1^T P_1 X' = w_3^{(n-2)T}X''$, where $X'' = A_1^T P_1 X' = A_1^T P_1 A^T P X$. s_3 is considered as an independent component as the $(n-2)$-dimensional data X''. Thus the data dimension has been reduced from n to $(n-2)$. Let P_2 be the whitening matrix of X'', which can be constructed easily with the eigenvalues and eigenvectors of $A_1^T A_1$. We have $w_3^{(n-2)} = P_2^T \tilde{w}_3^{(n-2)}$, and $w_3 = P^T \cdot A P_1^T \cdot A_1 P_2^T \tilde{w}_3^{(n-2)}$. In this way after the estimation of $\tilde{w}_3^{(n-2)}$(from which w_3 is constructed) and some preprocessing, estimation of the next independent component can be performed in $(n-3)$-dimensional parameter space. And so on until the last independent component is recovered.

In practical computation, usually the elements of $\tilde{w}^{(i)}$ are hardly 'strictly' equal to zero. When none of them equals to zero, we can choose its last element as the first non-zero element to construct A (or A_1, etc.).

2.2 Implementation

Using our dimension reduction decorrelation method, the decorrelation step is performed only once for each independent component. While if we use the Gram-Schmidt-like deflation scheme, as we have shown before, we must 'deflate' the weight vector after each update iteration for each independent component.

Based on the analysis above, given n observations, the ICA algorithm for extracting n independent components using a one-unit objective function and our decorrelatin method is formulated as (initially let $k = 1, D = I_n$):

for k = 1:n

1. *If $k = 1$, $A \leftarrow I_n$; otherwise according to Eq.6, use the vector u to construct the $(n-k+2)\times(n-k+1)$ matrix A. $X \leftarrow A^T X$.*
2. *Preprocess the data X with whitening. Let P be the $(n-k+1)\times(n-k+1)$ whitening matrix. If k=1, P is obtained by PCA; otherwise let the eigenvalue decomposition (EVD) of $A^T A$ is EDE^T, and $P \leftarrow D^{-1/2}E^T$. $X \leftarrow PX$, $D \leftarrow PA^T D$.*
3. *Optimize the chosen one-unit objective function to estimate an independent component s_k from data X: $s_k \leftarrow u^T X$, where u is a $(n-k+1)$-dimensional de-mixing weight vector of X. $w_k \leftarrow D^T u$.*

end

The computation load (or time) used for each independent component depends on the number of samples, the iteration steps used for convergence of the

chosen contrast function, and the dimension of the observations (or de-mixing weight vectors). If the Gram-Schmidt-like deflation scheme is adopted, without taking computation of decorrelation into account, computation of each update iteration is almost the same in estimating all the independent components. In our method, since the independent components obtained later are extracted from lower dimensional data, computation of each update iteration becomes less.

3 Experiments and Discussion

ICA has been applied in finance to construct factor models [1, 2]. We use ICA to extract 22 independent sources with the returns of 22 stocks as observations and compare the performances of our decorrelation method and the Gram-Schmidt-like deflation scheme. There are 2072 samples for each stock. In all experiments, the contrast function is as Eq.2 with $G(u) = \frac{1}{4}u^4$. FastICA (Eq.3) is used to do the optimization. MATLAB is used to do the simulation.

First in order to compare the convergence of these two methods, their termination conditions are set to be the same to guarantee the same quality of the independent components obtained by them. In the Gram-Schmidt-like deflation scheme, the termination condition is $||w(k)-w(k-1)|| < \varepsilon$ or $||w(k)+w(k-1)|| < \varepsilon$. In our dimension reduction method, since $w = D^T u$, the termination condition for $u(k)$ is $||D^T \cdot (u(k) - u(k-1))|| < \varepsilon$ or $||D^T \cdot (u(k) + u(k-1))|| < \varepsilon$. We randomly choose the initial condition for the two methods and repeat them 100 times. The average number of iterations and time needed for convergence of each IC are shown in Fig. 1. Using these two methods, each independent component takes almost the same number of iteration steps for convergence. But our method takes less time, especially for the independent components processed later. However, when the same initialization condition is used, there do exist some cases (about 1%) where our method needs fewer iteration steps, or even our method converges normally (about 50 iteration steps needed) while the Gram-Schmidt-like deflation method does not converge in 1000 steps.

In another experiment we compare the time taken by each iteration step using these two methods. We neglect the termination condition and fix the number of iteration steps used for extracting each independent component as 50, and compare the time taken by these two methods, see Fig. 2. With our dimension reduction method, time taken by each independent component decreases quickly when its sequence number increases. This is encouraging when the number of independent components is large.

4 Conclusion

When a one-unit contrast function is used to estimate the whole ICA transformation, a decorrelation method is needed to prevent the contrast function from converging to the same optimum for different independent components.

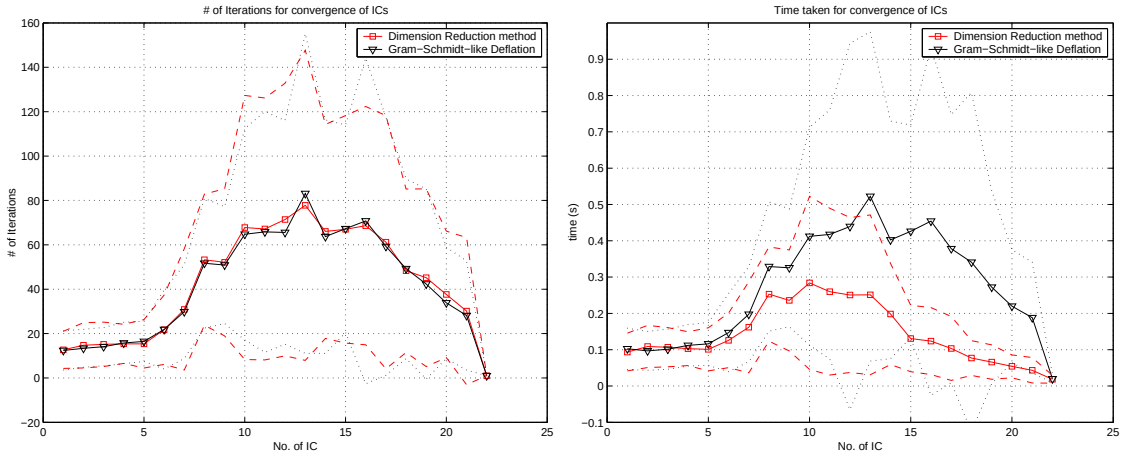

Fig. 1. Average number of iteration steps and time needed for convergence of each independent component Using the deflation scheme and the dimension reduction method respectively. Left and Right are the number of iterations and time used for convergence of each independent component respectively. The dashed and dotted lines indicate standard deviations of the two methods.

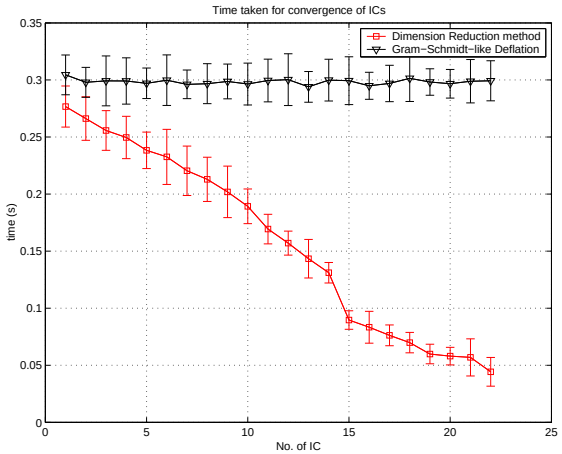

Fig. 2. Average time taken by our decorrelation method and the deflation scheme with the number of iterations used for estimation of each independent component fixed as 50, and each method has been repeated 100 times. Error bar: standard deviation.

Based on the orthogonality of the de-mixing matrix of whitened data, we propose a decorrelation method which lowers the dimension of the observations and de-mixing weight vectors when estimating subsequent independent components. Obviously the Gram-Schmidt-like deflation scheme is easier for comprehension and implementation. However, the dimension reduction method provides better convergence and is more efficient comparing with the popular deflation scheme.

Acknowledgement

The work in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administration Region, China. We would thank Prof. Hyvärinen and his colleagues for providing free download of FastICA package for MATLAB. Also we are grateful to the reviewers for their helpful comments.

References

1. Siu-Ming Cha and Lai-Wan Chan. Applying independent component analysis to factor model in finance. *Intelligent Data Engineering and Automated Learning-IDEAL 2000, Springer*, pages 538–544, 2000.
2. Lai-Wan Chan and Siu-Ming Cha. Selection of independent factor model in finance. In *proceedings of 3rd International Conference on Independent Component Analysis and blind Signal Separation*, San Diego, California, USA, December 2001.
3. P. Comon. Independent component analysis – a new concept? *Signal Processing*, 36:287–314, 1994.
4. N. Delfosse and P. Loubaton. Adaptive blind separation of independent sources: a deflation approach. *Signal Processing*, 45:59–83, 1995.
5. Aapo Hyvärinen. A family of fixed-point algorithms for independent component analysis. *ICASSP*, pages 3917–3920, 1997.
6. Aapo Hyvärinen. New approximations of differential entropy for independent component analysis and projection pursuit. In *Advances in Neural Information Processing Systems 10*, pages 273–279. MIT Press, 1998.
7. Aapo Hyvärinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999.
8. Aapo Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, 2:94–128, 1999.
9. Aapo Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9(7):1483–1492, 1997.
10. I. J. Jolliffe. *Principal Component Analysis*. Springer series in Statistics. Springer Verlag, 2nd edition, 2002.
11. J. Karhunen, E. Oja, L. Wang, R. Vigario, and J. Joutsensalo. A class of neural networks for independent component analysis. *IEEE Trans. on Neural Networks*, 8(3):486–504, 1997.
12. D. Luenberger. *Optimization by Vector Space Methods*. Wiley, 1969.