ICA PREPROCESSING FOR TIME SERIES PREDICTION

Simona Mălăroiu, Kimmo Kiviluoto and Erkki Oja

Simona.Malaroiu@hut.fi, Kimmo.Kiviluoto@hut.fi, Erkki.Oja@hut.fi
Helsinki University of Technology,
Laboratory of Computer and Information Science,
P.O.B. 2200, 02015 HUT, Finland

ABSTRACT

We propose a new method to predict time series using the technique of Independent Component Analysis (ICA) as a preprocessing tool. If certain assumptions hold, we show that ICA can be used to transform a set of time series into another set that is easier to predict. These assumptions are not unrealistic for many real-world time series, including financial time series. After prediction, the original time series can be constructed by the inverse ICA transform. We have tested this approach on two sets of data: artificial toy data and financial time series. The results suggest that these can be predicted more accurately using the ICA preprocessing.

1. INTRODUCTION

In this paper we build a prediction model using Independent Component Analysis (ICA). In the introduction part we briefly explain the concept of ICA, we present the prediction algorithm and the general model we use. In Section 2, the generic solution to the ICA problem is outlined and the FastICA algorithm is introduced. Further on, in section 3 and 4, the simulations and the results are shown and some conclusions are drawn.

We start by considering a set of parallel signals or time series $x_i(t)$, with i indexing the individual time series, $i = 1, \dots, n$ and t denoting discrete time. In our case these signals are the toy data (see Fig. 3 or the financial time series. In the basic ICA, a generative model is assumed, by which the original signals $x_i(t)$ are instantaneous linear mixtures of *independent source signals or underlying factors* $s_j(t)$, $j = 1, \dots, m$ with some unknown mixing coefficients $a_{i,j}$:

$$x_i(t) = \sum_j a_{i,j} s_j(t), \tag{1}$$

for each signal $x_i(t)$. We assume the effect of each time-varying underlying factor $s_j(t)$ on the measured time series to be approximately linear.

Utilizing information of either higher-order moments or time structure of the observed time series $x_i(t)$, the ICA algorithms are able to find good estimates for the underlying independent signals $s_j(t)$ and the unknown mixture coefficients $a_{i,j}$.

If we go to vector-matrix formulation, defining a vector-valued time series $\mathbf{x}(t) = [x_1(t), \dots, x_n(t)]$ with elements $x_i(t)$, a vector-valued source signal $\mathbf{s}(t)$ with elements $s_j(t)$, and a matrix $\mathbf{A} = (a_{i,j})$, then we can write (1) as:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t). \tag{2}$$

Matrix $\mathbf{A}$ is called the mixing matrix. The basic idea of ICA is that we do not have to know either the matrix $\mathbf{A}$ or the vector $\mathbf{s}(t)$ at all, but instead can estimate the model and obtain both matrix $\mathbf{A}$ and the underlying factors $\mathbf{s}(t)$, given sequential observations on $\mathbf{x}(t)$ if we make the assumption that the factors are *s*tatistically independent.

The ICA model (1) is realistic in certain sensor array applications in which a number of independent signals s_i arrive at a number of sensors but are weighted and superimposed due to the different locations of the sensors. Also, in the case of financial time series, there may be some underlying factors like seasonal variations or economic events that affect a number of simultaneous time series but can be assumed to be quite independent [2]. In [9], evidence of such factors was found also in retail store sales data. It is notable that ICA has a close relationship to minimizing algorithmic complexity [11]; the obtained signals $s_j(t)$ have lower complexity and clearer structure than their mixtures, and therefore it may be expected that predicting them is easier than predicting their mixtures such as the original time series.

We propose the following algorithm to predict a set of time series:

1. After subtracting the mean of each time series

and removing as much as possible of the second order statistic effects by normalization (after which each time series has zero mean and unit variance), estimate the independent components $s_j(t)$ of the given set of time series, and simultaneously find the mixing coefficients $a_{i,j}$ in (1).

2. For each component, apply suitable filtering to reduce the effects of noise – smoothing for components that contain very low frequencies (trend, slow cyclical variations), and high-pass filtering for components containing high frequencies and/or sudden shocks. Different filtering methods may be used for different components.

3. Predict each smoothed independent component separately, for instance using some method of AR-modeling.

4. Combine the predictions for each independent component by weighing them with the coefficients $a_{i,j}$, thus obtaining the predictions for the original observed time series $x_i(t)$.

The prediction model is represented in Fig. 1. It consists of an unmixing stage where the sources are obtained by applying a separating linear transformation $\mathbf{W}$ to the input time series $\mathbf{x}(t)$ (see Section 2 for details for computing $\mathbf{W}$),

$$\mathbf{s}(t) = \mathbf{W}\mathbf{x}(t) \tag{3}$$

The elements of $\mathbf{s}(t)$ are the underlying factors found by applying ICA.

The next stages consist of a non-linear smoothing function $\mathbf{f}$ and an AR prediction function $\mathbf{g}$. The smoothing is formally done by applying $\mathbf{f}$ on the source vectors $\mathbf{s}(t)$,

$$\mathbf{s}^s(t) = \mathbf{f}[\mathbf{s}(\cdot)] \tag{4}$$

Taking into consideration the temporal structure in a q-order AR model, the prediction equation is:

$$\mathbf{s}^p(t+1) = \mathbf{g}[\mathbf{s}^s(t), \mathbf{s}^s(t-1), \ldots, \mathbf{s}^s(t-q)] \tag{5}$$

The forecasted values $\mathbf{x}^p(t)$ result from the mixing stage by applying a linear transform $\mathbf{A}$ to the predicted sources $\mathbf{s}^p(t)$.

$$\mathbf{x}^p(t) = \mathbf{A}\mathbf{s}^p(t) \tag{6}$$

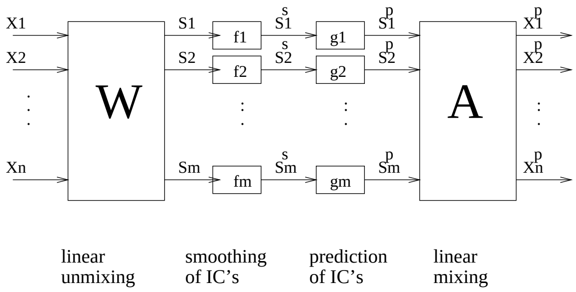


Figure 1: An illustration of the prediction model. The unmixing stage consists of a linear unmixing matrix $\mathbf{W}$ yielding the sources $s_j(t)$, $j = 1, \ldots, m$. Applying the nonlinear transfer functions f_j, $j = 1, \ldots, m$ the smoothed components are obtained: $s_j^s(t)$, $j = 1, \ldots, m$. The forecasted time series result from a linear mixing of the predicted smoothed sources $s_j^p(t)$, $j = 1, \ldots, m$.

2. INDEPENDENT COMPONENT ANALYSIS AND ICA ALGORITHMS

In the basic approach to solve the ICA problem [1, 3, 7, 8, 10], the temporal structure of the time series is in fact omitted and $\mathbf{s}(t)$ and $\mathbf{x}(t)$ in Eq. (3) are regarded as realizations of random vectors $\mathbf{s}$ and $\mathbf{x}$. We thus seek the solution in the form:

$$\hat{\mathbf{s}} \equiv \mathbf{y} = \mathbf{W}\mathbf{x}. \tag{7}$$

The goal is now to find a matrix $\mathbf{W}$ that makes the elements of $\mathbf{y}$ statistically independent. We call such a matrix a separating matrix. A recent review of various information theoretic contrast functions for solving $\mathbf{W}$, like mutual information, negentropy, maximum entropy, and infomax, as well as the maximum likelihood approach, is given by Cardoso [3], who also discusses the numerical problems in minimizing / maximizing such contrast functions.

The problem of solving the separating matrix $\mathbf{W}$ is somewhat simplified if we consider only one of the source signals at a time. From eq. (7) it follows

$$\hat{s}_i \equiv y_i = \mathbf{w}_i^T \mathbf{x} \tag{8}$$

with $\mathbf{w}_i^T$ the i-th row of $\mathbf{W}$. The last author has earlier suggested and analyzed neural type one-unit learning rules [6] that give as solutions one row $\mathbf{w}_i^T$ of the separating matrix. A condition of local convergence to a correct solution was given. The condition is very robust and shows that a wide range of nonlinear functions in these learning rules are possible.

The problem is further simplified by performing a preliminary sphering or prewhitening of the data $\mathbf{x}$:

the observed vector $\mathbf{x}$ is first linearly transformed to another vector whose elements are mutually uncorrelated and all have unit variances. This transformation is always possible and can be accomplished by classical Principal Component Analysis. At the same time, the dimensionality of the data should be reduced so that the dimension of the transformed data vector equals m, the number of independent components. This also has the effect of reducing noise. It can be shown that after this preprocessing, $\mathbf{W}$ will be an orthogonal matrix.

As an example of contrast functions, consider the simple case of maximizing the kurtosis $E\{y_i^4\}-3[E\{y_i^2\}]^2$ of the estimated signals y_i. Because we assumed that the estimated signals have unit variance, this reduces to maximizing the fourth moment $E\{y_i^4\}$. Its gradient with respect to $\mathbf{w}_i$ [see Eq. (8)] is $4E\{(\mathbf{w}_i^T\mathbf{x})^3\mathbf{x}\}$. Using this gradient and a normalizing term, an efficient fixed point iteration algorithm can be obtained [5]:

1. Take a random initial vector $\mathbf{w}(0)$ of norm 1. Let $k=1$.

2. Let $\mathbf{w}(k) = E\{\mathbf{x}(\mathbf{w}(k-1)^T\mathbf{x})^3\} - 3\mathbf{w}(k-1)$. The expectation can be estimated using a large sample of $\mathbf{x}$ vectors (say, 1,000 points).

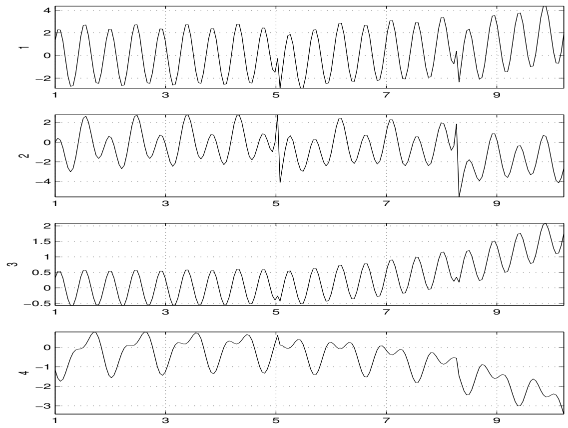
3. Divide $\mathbf{w}(k)$ by its norm.

4. If $\mid \mathbf{w}(k)^T\mathbf{w}(k-1) \mid$ is not close enough to 1, let $k=k+1$ and go back to step 2. Otherwise, output the vector $\mathbf{w}(k)$.

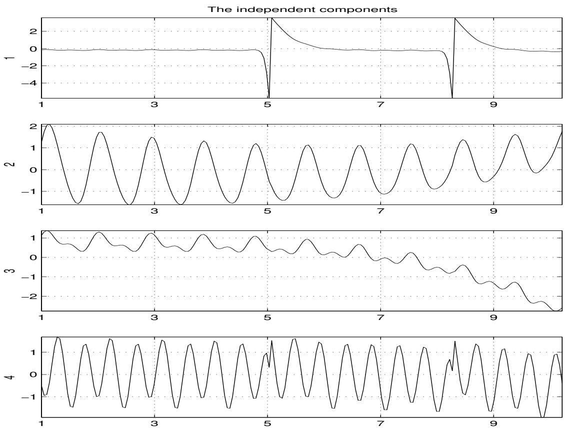
The final vector $\mathbf{w}(k)$ given by the algorithm equals the transpose of one of the rows of the (orthogonal) separating matrix $\mathbf{W}$.

To estimate m independent components, we run this algorithm m times. To ensure that we estimate each time a different independent component, we use the deflation algorithm that adds a simple orthogonalizing projection inside the loop. Recall that the rows of the separating matrix $\mathbf{W}$ are orthonormal because of the sphering. Thus we can estimate the independent components one by one by projecting the current solution $\mathbf{w}(k)$ on the space orthogonal to the rows of the separating matrix $\mathbf{W}$ previously found. Also a symmetrical orthogonalization is possible.

This algorithm, with the preliminary whitening and several extensions, is implemented in Matlab in the FastICA package available through the WWW [4]. A remarkable property of the FastICA algorithm is that a very small number of iterations, usually 5-10, seems to be enough to obtain the maximal accuracy allowed by the sample data. This is due to the cubic convergence of the algorithm shown in [5].



(a) The mixtures



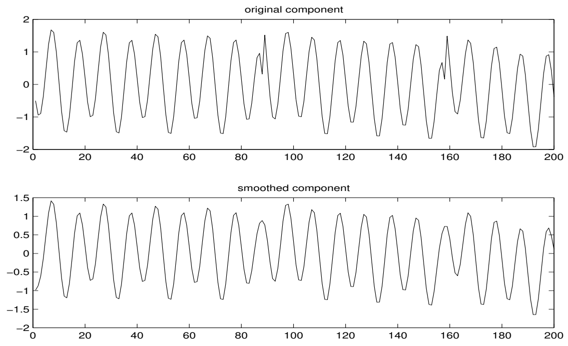
(b) The four independent components

Figure 2: The mixtures and the four independent components

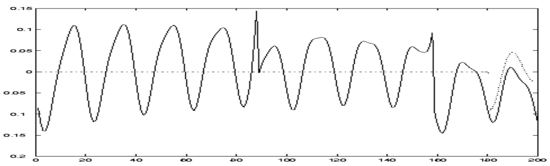
3. SIMULATIONS AND RESULTS

In our simulations we used the FastICA package [4] implementation of the above algorithm. The number of IC's is variable, we considered 4 independent components in the toy data experiment and 5 independent components for the experiment with financial time series.

In figures 2, 3 the algorithm is applied on a toy data set that consists of a mixture of four different signals: a trend like signal, a spiky signal and two seasonal type signals. Fig. 2a shows the original time series (mixtures) and Fig. 2b the independent components found by the FastICA algorithm. The IC's are very close to the true source signals.



(a) Original (above) and smoothed (below) independent component



(b) Prediction of the mixture (dotted line)

Figure 3: Smoothing and prediction

The smoothing method we used was the approximation of the independent components with 3rd order polynomials (spline interpolation) using various smoothing tolerances. The smoothing has been designed to allow visual determination of optimal smoothing tolerance by viewing smoothed curves on the graphic display. The procedure is the following: smooth with an initial tolerance, visualize both the smoothed and original time series and smooth again with the tolerance most visually appropriate so that the characteristics of the time series are preserved.

When dealing with noisy data it makes sense to use a more slowly varying curve which does not interpolate the data points but damps out the noise component. By increasing the smoothness of the curve the variance of predicted values is reduced. However, in the same time this can introduce bias in the predicted values where the true underlying relationship is changing rapidly. The smoothing could be also done in a more principled manner: a first step in this direction could be the minimization of the final prediction error using cross-validation. Fig. 3a shows the fourth IC and its smoothed version.

After smoothing, an AR model is fit to the smoothed data, increasing in time the length of the data with as many points as the number of predicted moments. The order of the model is chosen to minimize the mean square error. The order is different from component to component. In Fig. 3b the continuous line represents the independent component and for the last 20 points the prediction using ICA is plotted using the dotted line.

All these transformations lead us to processed independent components for the future interval of time. Fitting the mixing model (1) to these forecast independent components $\mathbf{s}^p(t)$, the predictions $\mathbf{x}^p(t)$ for the initial time series are obtained. On this data, the ICA prediction clearly outperforms the direct prediction, the error obtained with the ICA prediction being significantly smaller than the error obtained with AR prediction on mixtures. In Fig. 4 the lower curve represents the original toy data i.e. one of the mixtures, the middle one shows the predicted values using our algorithm and the upper curve shows the AR prediction applied directly on the mixture.

We also considered the case where the original signals were AR-processes. In this case, the results obtained using prediction with ICA were similar to the ones obtained using AR prediction directly on the original signals.

In our last experiment, we applied our algorithm on a set of 10 relevant foreign exchange rate time series. The results were promising, as the ICA prediction performed better than direct prediction. Fig. 5 shows an example of prediction using our method. The upper figure represents one of the original mixtures and the lower one the forecast obtained using ICA prediction for an interval of 50 values. In table 1 there is a comparison of errors obtained applying classical AR prediction and our method. The last column shows the magnitude of the errors when no smoothing is applied to the currencies.

4. DISCUSSION

We have presented a prediction model that adopts the ideas from independent component analysis and applies them in forecasting of time series.

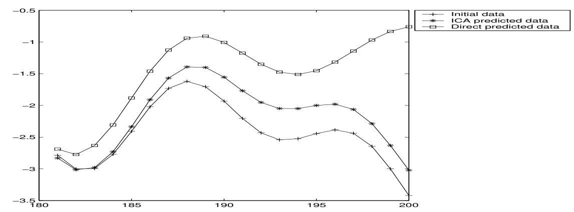


Figure 4: Comparison between the prediction methods

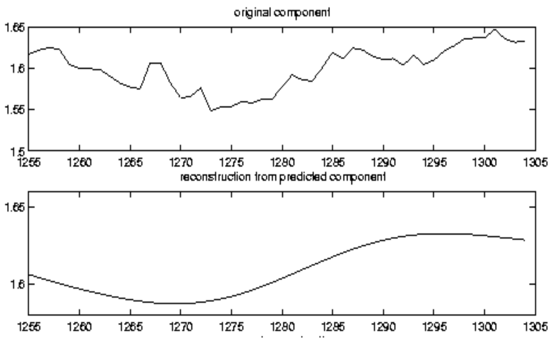


Figure 5: Prediction of real-world financial data: the upper figure represents the original mixtures and the lower one the forecast obtained using ICA prediction for an interval of 50 values.

Table 1: The errors obtained with our method and the classical AR method. Ten currencies were considered and five independent components. With classical AR, several different smoothing tolerances on the original time series were tested. With ICA, each individual IC was smoothed by the tolerance that visually seemed most suitable.

AR smoothing tolerance	AR NMSE	ICA NMSE
2.0	9.7	
0.5	9.1	
0.1	4.7	
0.08	3.9	2.3
0.06	3.4	
0.05	3.1	
0.00	4.2	

The main contribution of this paper is the prediction algorithm itself. It performed well both on toy data and financial time series. We start by supposing that there are some independent factors that affect the time evolution of economic time series. This is one assumption that must hold so that our algorithm works on financial time series. The economic indicators, interest rates and psychological factors can be the underlying factors of exchange rates, as they are closely tied to the evolution of the currencies. Given the time series, by forecasting the underlying factors, which in our case are the independent components, a better prediction of the time series can be obtained. The algorithm predicts very well the turning points.

ICA and AR prediction are linear techniques, although the first one has a non-linear learning rule; however, the smoothing makes the proposed algorithm non-linear. The smoothing enhances the signal-to-noise ratio, allowing a more accurate prediction of the source signals and thus also of the original time series.

For time series which are generated by a linear model – AR-processes, for instance – our algorithm has similar performances with the classical linear time series analysis techniques. Even when such time series are (linearly) mixed, they remain linear, and they can be predicted using standard techniques.

It should be noted that some of the sources may also be useful in analyzing the impact of different external phenomena on the foreign exchange rates [2].

The model we have proposed is flexible and allows a varying degree of smoothing and a different model order for each source. An obvious way to further improve the model would be to consider ICA contrast functions that explicitly try to enhance the predictability of the source signals.

Acknowledgments The first author would like to thank Răzvan Cristescu for helpful comments and discussions.

5. REFERENCES

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

[2] A. Back and A. Weigend. First application of Independent Component Analysis to extracting structure from stock returns. *Int. J. Neural Systems 8*, 473-484, 1997.

[3] J.-F. Cardoso. Blind signal separation: statistical principles. *Proc. IEEE 86*, 2009 - 2025, 1998.

[4] The FastICA public domain package available at `http://www.cis.hut.fi/projects/ica/fastica/`.

[5] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for Independent Component Analysis. *Neural Computation 9*, 1483 - 1492, 1997.

[6] A. Hyvärinen and E. Oja, E. Independent Component Analysis by general non-linear Hebbian-like learning rules. *Signal Processing 64*, 301 - 313, 1998.

[7] C. Jutten and J. Herault. Independent component analysis (INCA) versus independent component analysis. In *Signal Processing IV: Theories and Applications* (J. Lacoume et al, eds.), Amsterdam: Elsevier, 643 - 646, 1988.

[8] 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*, 486 - 504, 1997.

[9] K. Kiviluoto and E. Oja. Independent component analysis for parallel financial time series. In *Proc. ICONIP'98*, Kitakyushu, Japan, Oct. 1998, 895–898.

[10] E. Oja. The nonlinear PCA learning rule in independent component analysis. *Neurocomputing 17*, 25 - 45, 1997.

[11] P. Pajunen. Extensions of Linear Independent Component Analysis: Neural and Information-Theoretic Methods. Ph.D. Thesis, Helsinki University of Technology, 1998.

[12] A. S. Weigend, N.A. Gershenfeld. Time Series Prediction. Proceedings of NATO Advanced Research Workshop on Comparative Time Series Analysis. Santa Fe, New Mexico, May 14-17, 1992.