Complexity pursuit:

Separating interesting components from time-series

Aapo Hyvärinen
Neural Networks Research Centre
Helsinki University of Technology
P.O. Box 5400, FIN-02015 HUT, Finland
aapo.hyvarinen@hut.fi
http://www.cis.hut.fi/projects/ica/

Abstract

A generalization of projection pursuit for time series, i.e. signals with time structure, is introduced. The goal is to find projections of time series that have interesting structure. We define the interestingness using criteria related to Kolmogoroff Complexity or coding length: Interesting signals are those that can be coded with a short code length. We derive a simple approximation of coding length that takes into account both the nongaussianity and the autocorrelations of the time series. Also, we derive a simple algorithm for its approximative optimization. The resulting method is closely related to blind separation of nongaussian, time-dependent source signals.

1 Introduction

In multivariate statistics, a central problem is to find 1-D projections of the data vector which reveal interesting aspects of the data. Denoting by $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ the n-dimensional random vector corresponding to the observed data, such projections are defined by a constant vector $\mathbf{w} = (w_1, w_2, ..., w_n)^T$ as the linear combination $\mathbf{w}^T\mathbf{x} = \sum_i w_i x_i$. A classical method for finding such projections is to compute the principal components (Oja, 1982; Jolliffe, 1986) of the data vectors. The first principal component gives a projection that optimally approximates the data vector in the sense of mean square error. The second

principal component gives the optimal approximation of the residual of the approximation given by the first principal component, and so forth.

Recently, however, it has been argued that the projections given by the principal components do not describe the data in a meaningful way in many cases. This is because principal component analysis neglects important aspects of the data such as clustering and higher-order independence. This has lead to development of methods that are *not* based on mean-square error, but rather on higher-order statistics. This means using other information than that contained in the covariance matrix.

An important method using higher-order information is projection pursuit (Friedman and Tukey, 1974). In basic (1-D) projection pursuit, we try to find directions $\mathbf{w}$ such that the projection of the data vector in that direction, $\mathbf{w}^T\mathbf{x}$, has an "interesting" distribution in the sense of displaying some structure. It has been argued by Huber (1985) and by Jones and Sibson (1987) that the Gaussian distribution is the least interesting one, and that the most interesting directions are those that show the least Gaussian distribution. An information-theoretic justification for this is that the Gaussian distribution has maximum entropy among all distributions of unit variance. Entropy can be considered a measure of disorder, i.e. lack of (interesting) structure.

Projection pursuit is closely related to independent component analysis (ICA) (Jutten and Herault, 1991; Comon, 1994). ICA is a statistical model where the observed data is expressed as a linear transformation of latent variables that are nongaussian and mutually independent. We may express the model as

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1}$$

where $\mathbf{x} = (x_1, x_2, ..., x_n)^T$ is the vector of observed random variables, $\mathbf{s} = (s_1, s_2, ..., s_n)^T$ is the vector of the latent variables called the independent components or source signals, and $\mathbf{A}$ is an unknown constant matrix, called the mixing matrix. Exact conditions for the identifiability of the model were given in (Comon, 1994). Note that we assumed, for simplicity, that the dimension of $\mathbf{x}$ equals the dimension of $\mathbf{s}$, but this need not necessarily be the case. If nongaussianity is measured suitably, finding maximally nongaussian directions gives one method of estimating ICA (Delfosse and Loubaton, 1995; Hyvärinen and Oja, 1997; Hyvärinen, 1999a). Thus we obtain independent components as the projection pursuit directions $\mathbf{s}_i = \mathbf{w}_i^T\mathbf{x}$, and the $\mathbf{w}_i$ correspond to estimates of the rows of $\mathbf{A}^{-1}$.

In many cases, projection pursuit and ICA are applied on data sets that are not simply random vectors, but multivariate time series, i.e. signals with time dependencies. However, these two methods in their basic forms completely ignore any time structure and use only the marginal distributions of the projections.

Interestingly, it has been shown that under some restrictions, the time dependency information alone is sufficient to estimate independent components (Tong et al., 1991; Belouchrani et al., 1997; Matsuoka et al., 1995; Molgedey and Schuster, 1994). Results obtained by these methods are likely to improve if one exploits all the available information on the interestingness of the projections. It would be most useful, therefore, to define a more general method that finds interesting projections of time series using both the nongaussianity and the time-structure of the projections.

In this paper, we propose a generalization of projection pursuit that takes into account the time structure of the projections as well. This is based using the coding complexity of the projection. In our "complexity pursuit", we search for projections that can be easily coded. This is a general-purpose measure of structure, closely related to using Kolmogoroff Complexity as a criterion for finding a representation (Pajunen, 1998a), and is probably connected to information-processing principles used in the brain (Atick, 1992; Hochreiter and Schmidhuber, 1999; Pajunen, 1998a). We develop simple approximations of coding complexity, and derive an algorithm for (approximative) minimization of the complexity approximations. This algorithm can be seen as a principled combination of algorithms using the criteria of nongaussianity and autocorrelation as in projection pursuit, ICA and source separation. Moreover, it can be interpreted as maximum likelihood estimation of the ICA model, assuming time-correlated models for the independent components.

This paper is organized as follows. The basic principle of complexity pursuit is introduced in Sec. 2. Suitable approximations of complexity are developed in Sec. 3. An algorithm for minimizing the complexity approximation is given in Sec. 4. Relation to other methods is discussed in Sec. 5, simulation results are given in Sec. 6, and, finally, conclusions are drawn in Sec. 7.

2 Complexity and time series

Projection pursuit and ICA, in their classic forms, consider the observed data $\mathbf{x}$ to be a multivariate random vector. In many cases, however, the observed data is a multivariate time series $\mathbf{x}(t)$, i.e. a vector of time signals. A time signal has much more structure than a random variable. This structure is completely neglected in projection pursuit. In some ICA methods, it is taken into account, but such ICA methods typically neglect the marginal distribution of the data. A unifying theoretical framework for using both kinds of information is given by Kolmogoroff Complexity (Pajunen, 1998a).

Kolmogoroff Complexity is based on the interpretation of coding length as structure. Suppose that we want to code a signal $s(t), t = 1, ..., T$. For simplicity, let us assume for the moment that the signal is

binary, so that every value $s(t)$ is 0 or 1. In general, it is not possible to code this signal with less than T bits, so that every bit in the code gives the value of $s(t)$ for one t. However, most natural signals have *redundancy*, i.e. parts of the signal can be efficiently predicted from other parts. Such a signal can be coded, or compressed, so that the code length is shorter than the original code length. It is well-known that audio or image signals, for example, can be coded so that the code length is decreased considerably. This is because such natural signals are highly structured. For example, image signals do not consist of random pixels, but of such higher-order regularities as edges and contours.

We could thus measure the amount of structure of the signal $s(t)$ by the amount of compression that is possible in coding the signal. For signals of fixed length T, the structure could be measured by *the length of the shortest possible code for the signal*. Note that the signal could be compressed by many different kinds of coding schemes (the coding theory literature is full of them), but we are here considering the shortest code possible, thus maximizing the compression over all possible coding schemes. This is a non-rigorous definition of Kolmogoroff Complexity; for a more rigorous definition of the concept, see (Pajunen, 1998a; Pajunen, 1998b). Kolmogoroff Complexity is usually defined for binary signals, but it could be applied on continuous-valued signals as well after a suitable quantization.

Thus we arrive at the following theoretical *definition of Complexity Pursuit*. The goal is to find projections $\mathbf{w}^T\mathbf{x}(t)$ such that the Kolmogoroff Complexity of the projection is minimized. We must also fix the scale of the projection, since otherwise taking $\mathbf{w} = 0$ would give a projection that is trivially structured. Thus we constrain the variance of the projection to be unity, as usual in projection pursuit.

Kolmogoroff Complexity is a theoretical measure, since its computation involves finding the best coding scheme for the signal. The number of possible coding schemes is infinite, so this optimization in intractable. Therefore, in practice approximations must be used.

If the signals have no time structure, their Kolmogoroff Complexities are given (approximately) by their entropies (Pajunen, 1998a). In this case, we rediscover ordinary projection pursuit. Furthermore, in (Pajunen, 1998a; Pajunen, 1999) it was shown how to approximate Kolmogoroff Complexity by criteria using the autocorrelations of the signals, in which case we rediscover a method closely related to the source separation methods in (Tong et al., 1991; Belouchrani et al., 1997; Molgedey and Schuster, 1994). In the next section, we introduce a more general framework for approximating Kolmogoroff Complexity.

3 Approximation of complexity

In this section, we derive an approximation of the Kolmogoroff Complexity of a scalar signal $y(t), t = 1, ..., T$. For simplicity, the signal is assumed to have zero mean and unit variance.

3.1 General formulation by predictive coding

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

$$\hat{y}(t) = f(y(t-1), y(t-2), ...y(1)). \tag{2}$$

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

$$\delta y(t) = y(t) - \hat{y}(t) \tag{3}$$

is coded by a scalar quantization method. The point is that in many cases, it is easier to code the residual than the original value $y(t)$, if the predictor f is suitably chosen so that it gives a reasonable prediction of $y(t)$. This coding stategy is used for $y(t)$ at all time points t. Thus the whole signal is coded by coding the residuals (and the initial value(s) of $y(t)$ for t near 1). The residuals are coded independently from each other, neglecting any dependencies.

According to the basic principles of information theory (Cover and Thomas, 1991), the length of this code is asymptotically approximated by the sum of the entropies H of the residuals. We use this as an approximation of the coding complexity:

$$\hat{K}(y) = \sum_t H(\delta y(t)) \tag{4}$$

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

$$\hat{K}(y) = TH(\delta y) \tag{5}$$

where δy denotes a random variable with the marginal distribution of the residual. Note that we made here the assumption that the signal is stationary. In the case of a non-stationary signal, more sophisticated models for the signal need to be developed, in which the function f is changing in time (Matsuoka et al., 1995).

3.2 Using linear models for prediction

To use the approximation in (5) in practice, we need to fix the structure of the predictor f, and find an approximation of the entropy of δy.

We use here a computationally simple predictor structure, given by a linear autoregressive model:

$$\hat{y}(t) = \sum_{\tau>0} \alpha_\tau y(t-\tau) \tag{6}$$

To optimize the performance of the predictor, the parameters α_τ should ideally be estimated so that the entropy of the residual δy is minimized. This is equivalent to estimating the autoregressive model by the method maximum likelihood, taking into account the true distribution of the residual. Alternatively, to simplify the computations, the α_τ could be estimated by a least-squares method, i.e. minimizing the variance of the residual. Note that in practice, the sum in (6) is taken over a finite, possibly very small set of lag indices τ.

3.3 Approximation of the entropy of residual

To approximate the entropy of δy, many different methods could be devised. In particular, it is a good idea to standardize the variable to unit variance to decouple the effects of scale and nongaussianity. Denoting by σ_δ^2 the variance of the residual, we have by the well-known scaling property of entropy:

$$H(\delta y) = H(\frac{\delta y}{\sigma_\delta}) + \log \sigma_\delta. \tag{7}$$

The estimation of the entropy of the standardized version $H(\delta y/\sigma_\delta)$ could be done, for example, using the general entropy approximation introduced in (Hyvärinen, 1998b). We adopt here a simpler method, however, which is possible by assuming that we have prior knowledge on the distribution of the residual. We assume that we know a good approximation of the (negative) logarithm of the probability density of the residual, denoted by G. Then we can plug this into the definition of entropy, and obtain the approximation

$$H(\delta y) \approx E\{G(\frac{\delta y}{\sigma_\delta})\} + \log \sigma_\delta. \tag{8}$$

In ICA, it is well-known that the exact form of the non-quadratic function used to probe higher-order statistics is not very important (Cardoso and Laheld, 1996; Hyvärinen and Oja, 1998; Cardoso, 2000). Likewise, in (Hyvärinen, 1998b) it was shown that entropy can be well approximated using a fixed non-quadratic function. 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. The simulations in Sec. 6 give some support for this assumption.

In particular, in many cases we can assume that the residuals are supergaussian, which seems to be the preponderant case in natural data (Hyvärinen, 1999b; Vigário et al., 2000). Then we can use the negative log-density of a generic supergaussian random variable, say

$$G(\delta y) = \frac{1}{2}\log 2 + \sqrt{2}|y|. \tag{9}$$

The additive constant in (9) is immaterial and can be omitted.

4 Finding minimum complexity directions

Using the approximation given in the preceding section, we can formulate a practical method for complexity pursuit. To find the "most interesting" directions $\mathbf{w}^T\mathbf{x}(t)$, use the approximation of complexity introduced in the preceding section and find its minima.

4.1 Formulating the criterion

Let us consider what kind of practical criterion we obtain using the above approximation of complexity for $y(t) = \mathbf{w}^T\mathbf{x}(t)$. First note that the values of α_τ and σ_δ are functions of $\mathbf{w}$ only. To emphasize this, we write $\alpha_\tau(\mathbf{w})$ and $\sigma_\delta(\mathbf{w})$. Thus we can express the approximation of complexity as a function of $\mathbf{w}$ only:

$$\hat{K}(\mathbf{w}^T\mathbf{x}(t)) = E\{G(\frac{1}{\sigma_\delta(\mathbf{w})}\mathbf{w}^T(\mathbf{x}(t) - \sum_{\tau>0}\alpha_\tau(\mathbf{w})\mathbf{x}(t-\tau)))\} + \log\sigma_\delta(\mathbf{w}). \tag{10}$$

As mentioned above, we must fix the scale of $\mathbf{w}^T\mathbf{x}(t)$ to gain direct access to the higher-order structure. Thus we constrain

$$E\{(\mathbf{w}^T\mathbf{x}(t))^2\} = 1 \tag{11}$$

The terms in the approximation in (10) have intuitive interpretations. The first one measures the contribution of the nongaussianity to the entropy of the residual of the linear predictor. The argument of G is normalized to unit variance, so this non-quadratic function measures the nongaussianity in the same way as the one-unit contrast function in (Hyvärinen and Oja, 1998; Hyvärinen, 1999a). Minimizing this term alone amounts to finding direction in which the residual is as nongaussian as possible.

The second term measures the contribution of the variance of the residual to its entropy. Minimization of this term alone amounts to finding a projection that has maximum autocorrelations, i.e. maximum time-dependencies. This principle is closely related to those used in blind source separation methods in (Tong et al., 1991; Molgedey and Schuster, 1994; Belouchrani et al., 1997), as will be seen in the Discussion.

Thus, our criterion uses simultaneously the two most widely used criteria for ICA and related methods: nongaussianity and autocorrelations. Moreover, Kolmogoroff Complexity leads to another modification of ordinary projection pursuit: It is the nongaussianity of the *residuals* of predictive coding that is measured, instead of the nongaussianity of the original signals.

4.2 Deriving the algorithm

4.2.1 The gradient

To find the minima of the approximation of complexity, we can use a simple gradient descent. The gradient of $\hat{K}$ in (10) with respect to $\mathbf{w}$ can be obtained straight-forwardly as

$$\nabla_{\mathbf{w}} \hat{K}(\mathbf{w}^T \mathbf{x}(t)) = \frac{1}{\sigma_\delta(\mathbf{w})} E\{(\mathbf{x}(t) - \sum_{\tau>0} \alpha_\tau(\mathbf{w})\mathbf{x}(t-\tau)) g(\frac{1}{\sigma_\delta(\mathbf{w})} \mathbf{w}^T(\mathbf{x}(t) - \sum_{\tau>0} \alpha_\tau(\mathbf{w})\mathbf{x}(t-\tau)))\}$$
$$+ \beta \nabla_{\mathbf{w}} \sigma_\delta(\mathbf{w}) - \frac{1}{\sigma_\delta(\mathbf{w})} E\{[\sum_{\tau>0} (\nabla_{\mathbf{w}} \alpha_\tau(\mathbf{w})) \mathbf{w}^T \mathbf{x}(t-\tau)] g(\frac{1}{\sigma_\delta(\mathbf{w})} \mathbf{w}^T(\mathbf{x}(t) - \sum_{\tau>0} \alpha_\tau(\mathbf{w})\mathbf{x}(t-\tau)))\} \quad (12)$$

where g is the derivative of G, and β is defined as

$$\beta = \frac{1}{\sigma_\delta(\mathbf{w})} [1 - \frac{1}{\sigma_\delta(\mathbf{w})} E\{\mathbf{w}^T(\mathbf{x}(t) - \sum_{\tau>0} \alpha_\tau(\mathbf{w})\mathbf{x}(t-\tau)) g(\frac{1}{\sigma_\delta(\mathbf{w})} \mathbf{w}^T(\mathbf{x}(t) - \sum_{\tau>0} \alpha_\tau(\mathbf{w})\mathbf{x}(t-\tau)))\}] \quad (13)$$

To simplify the resulting algorithm we use the following approximation of the gradient, which is quite accurate. Assume that $\mathbf{w}^T \mathbf{x}(t)$ is really generated by the autoregressive model that is used to predict it, and that G is the negative log-density of the residual. Consider the following lemma:

Lemma 1 *For any random variable x with density p_x, we have*

$$E\{x \frac{p'(x)}{p(x)}\} = -1 \quad (14)$$

Proof of lemma: by partial integration, we obtain

$$E\{x \frac{p'(x)}{p(x)}\} = \int x \frac{p'(x)}{p(x)} p(x) dx = \int x\, p'(x) dx = 0 - \int 1 \times p(x) dx = -1 \quad (15)$$

Applying this lemma for the residual (normalized to unit variance), we have

$$E\{\frac{1}{\sigma_\delta(\mathbf{w})}\mathbf{w}^T(\mathbf{x}(t)-\sum_{\tau>0}\alpha_\tau(\mathbf{w})\mathbf{x}(t-\tau))g(\frac{1}{\sigma_\delta(\mathbf{w})}\mathbf{w}^T(\mathbf{x}(t)-\sum_{\tau>0}\alpha_\tau(\mathbf{w})\mathbf{x}(t-\tau)))\}=1, \tag{16}$$

which implies that $\beta=0$. Moreover, the quantity $\sum_{\tau>0}(\nabla_{\mathbf{w}}\alpha_\tau(\mathbf{w}))\mathbf{w}^T\mathbf{x}(t-\tau)$ depends only on the past values of $\mathbf{w}^T\mathbf{x}(t)$. Therefore, it is independent from the residual $\mathbf{w}^T(\mathbf{x}(t)-\sum_{\tau>0}\alpha_\tau(\mathbf{w})\mathbf{x}(t-\tau))$ that has the role of the innovation process here. Thus the third term in the gradient vanishes as well.

Thus we have the following approximation of the gradient

$$\nabla_{\mathbf{w}}\hat{K}(\mathbf{w}^T\mathbf{x}(t)) \;\approx\; \frac{1}{\sigma_\delta(\mathbf{w})}E\{(\mathbf{x}(t)-\sum_{\tau>0}\alpha_\tau(\mathbf{w})\mathbf{x}(t-\tau))g(\frac{1}{\sigma_\delta(\mathbf{w})}\mathbf{w}^T(\mathbf{x}(t)-\sum_{\tau>0}\alpha_\tau(\mathbf{w})\mathbf{x}(t-\tau)))\} \tag{17}$$

To further simplify the resulting algorithm, we can use the fact that the factor $1/\sigma_\delta(\mathbf{w})$ multiplying the gradient is less important, since it does not change the direction of the gradient. Thus it can be omitted. Furthermore, the same factor multiplying the argument of g could be omitted in many cases without changing the point where the gradient vanished. This is because for homogenous g, for example $g(u)\propto \mathrm{sign}(u)$ or $g(u)\propto u^3$, this constant does not change the direction of the gradient, either, and could be omitted as well. In practice, we often use g that is a good approximation of a homogenous function (e.g. the tanh function as given below).

4.2.2 The algorithm

Thus we can use a simple (approximative) gradient descent for updating $\mathbf{w}$. To begin with, we can simplify the algorithm by first whitening the zero-mean data $\mathbf{x}(t)$, for example by:

$$\mathbf{z}(t)=\mathbf{V}\mathbf{x}(t)=(E\{\mathbf{x}(t)\mathbf{x}(t)^T\})^{-1/2}\mathbf{x}(t). \tag{18}$$

Now, the constraint of unit variance of $\mathbf{w}^T\mathbf{x}(t)$ can be replaced by the constraint of unit norm of $\mathbf{w}$. This is a standard procedure in ICA and projection pursuit (Comon, 1994; Friedman, 1987).

Denote by $\mathbf{z}(t)$ the whitened data, and by μ a learning rate. The algorithm is then as follows. At every step, first estimate the autoregressive constants $\alpha_\tau(\mathbf{w})$ in (6) for the time series given by $\mathbf{w}^T\mathbf{z}(t), t=1,...,T$. Then do the the gradient descent and normalization:

$$\mathbf{w}\leftarrow\mathbf{w}-\mu E\{(\mathbf{z}(t)-\sum_{\tau>0}\alpha_\tau(\mathbf{w})\mathbf{z}(t-\tau))g(\mathbf{w}^T(\mathbf{z}(t)-\sum_{\tau>0}\alpha_\tau(\mathbf{w})\mathbf{z}(t-\tau)))\} \tag{19}$$

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

The function g should be chosen as in ordinary ICA, but according to the probability distribution of the *residual* instead of the actual component $\mathbf{w}^T\mathbf{z}(t)$. If the residual is supergaussian, $g(u) = \text{sign}(u)$ is suitable. This could also be approximated by a smoother function $g(u) = \tanh(au)$ where $a \geq 1$ is a suitable constant (Bell and Sejnowski, 1995; Hyvärinen, 1999a). For subgaussian residuals, one could use $g(u) = u - \tanh(u)$ (Girolami, 1998), or $g(u) = u^3$, for example. For almost gaussian residuals, a linear g could be used. Obviously, any scaling constants multiplying g can be dropped, although they are needed, in principle, to insure that $-G$ is actually a log-density of unit variance.

To estimate several projections, one can simply use a deflation scheme (Delfosse and Loubaton, 1995; Hyvärinen and Oja, 1997). This means that after estimating m components, the new one is found by projecting $\mathbf{w}$, after every step of the algorithm, on the subspace orthogonal to the one spanned by the already estimated components. A simple Gram-Schmidt orthogonalization scheme accomplishes this (Hyvärinen and Oja, 1997; Hyvärinen, 1999a). Symmetric orthogonalization may be used as well, in which case the algorithm is more akin to ICA than projection pursuit (see next section).

4.2.3 Simple special case

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

$$\hat{y}(t) = \alpha_1 y(t-1). \tag{21}$$

The lag need not be equal to 1, but this is the basic case. This method may be very useful in practice since it takes into account the most basic form of autocovariance in the same way as the algorithms in (Tong et al., 1991; Molgedey and Schuster, 1994); additional terms may not provide much extra information in many applications. The parameter α_1 in the algorithm can then be estimated very simply by a least-squares method as

$$\hat{\alpha}_1 = \mathbf{w}^T E\{\mathbf{z}(t)\mathbf{z}(t-1)^T\}\mathbf{w}. \tag{22}$$

5 Discussion

5.1 Connection to independent component analysis

We have proposed an algorithm for finding interesting 1-D projections of multivariate time series, as measured by an approximation of Kolmogoroff Complexity. Finding interesting projections of random vectors, as measured by nongaussianity, is closely connected to ICA estimation, and therefore one might expect that our method is closely connected to ICA as well.

In fact, well-known ICA methods can be found as special cases of our method. First, assume that the signal has no (linear) time dependencies, which implies that the residual equals the signal itself. Then our algorithm reduces to

$$\mathbf{w} \leftarrow \mathbf{w} - \mu E\{\mathbf{z}(t) g(\mathbf{w}^T \mathbf{z}(t))\} \tag{23}$$

$$\mathbf{w} \leftarrow \mathbf{w}/\|\mathbf{w}\|. \tag{24}$$

This is in fact the algorithm proposed in (Hyvärinen and Oja, 1998) for finding one independent component. In particular, if G is chosen as the negative log-density of the residual (here: component), the results in (Hyvärinen and Oja, 1998) show that the algorithm converges to one of the independent components. Most well-known algorithms for estimating ICA for nongaussian components are closely related to this deflationary method (Amari et al., 1996; Bell and Sejnowski, 1995; Cichocki and Unbehauen, 1996; Cardoso and Laheld, 1996; Hyvärinen and Oja, 1997; Hyvärinen, 1999a; Karhunen et al., 1997; Oja, 1997).

As another special case, assume that the data is gaussian. Then the function g can be taken linear, and the method reduces to something that is closely related to minimization of lagged cross-correlations as in (Belouchrani et al., 1997). To see this more clearly, assume that we use the AR(1) predictor. Denoting by

$$\mathbf{C}_{-1} = \frac{1}{2}[E\{\mathbf{z}(t)\mathbf{z}(t-1)^T\} + E\{\mathbf{z}(t-1)\mathbf{z}(t)^T\}] \tag{25}$$

a symmetric version of the lagged covariance matrix, we have

$$E\{(\mathbf{z}(t) - \alpha_1 \mathbf{z}(t-1)) g(\mathbf{w}^T(\mathbf{z}(t) - \alpha_1 \mathbf{z}(t-1)))\} = [(1-\alpha_1^2)\mathbf{I} - 2\alpha \mathbf{C}_{-1}]\mathbf{w} \tag{26}$$

and the algorithm takes the form

$$\mathbf{w} \leftarrow \mathbf{w} - \mu[(1-\alpha_1^2)\mathbf{I} - 2\alpha \mathbf{C}_{-1}]\mathbf{w} \tag{27}$$

$$\mathbf{w} \leftarrow \mathbf{w}/\|\mathbf{w}\| \tag{28}$$

The algorithm can be considered as a power method for computing the dominant eigenvector of $\mathbf{C}_{-1}$. In (Tong et al., 1991) it was proposed that the independent components could be estimated by finding the eigen-value decomposition of $\mathbf{C}_{-1} = \mathbf{E}\mathbf{D}\mathbf{E}^T$. The eigenvectors $\mathbf{e}_i$ thus found give the independent components as $\mathbf{e}_i^T\mathbf{z}(t)$. This was motivated by the fact that such a decomposition gives signals that are uncorrelated in the usual way, as well as uncorrelated with the lag, i.e. $E\{(\mathbf{e}_i^T\mathbf{z}(t))(\mathbf{e}_j^T\mathbf{z}(t-1))\} = 0$ for $i \neq j$. In a deflationary scheme, where we estimate one component using our algorithm, then remove it from the data, and iterate such extraction steps, our algorithm estimates the same eigenvectors of $\mathbf{C}_{-1}$, and gives the same decomposition. It is interesting to note that minimizing delayed cross-correlations is thus seen to correspond to finding signals with maximum autocorrelations, not unlike in PCA where directions of maximum variance are found by a decorrelating eigen-value decomposition.

The connection to ICA estimation can be made even more explicit by considering an ICA model as in (1) where the independent components are modelled using a autoregressive model:

$$s_i(t) = \sum_{\tau>0} \alpha_\tau^i s_i(t-\tau) + \delta \tag{29}$$

where δ is a nongaussian random variable. Denote by $\mathbf{W} = (\mathbf{w}_1, ..., \mathbf{w}_n)^T$ the inverse of $\mathbf{A}$. The likelihood of such a model can be formulated as

$$\log L(\mathbf{w}_i, \alpha_\tau; i = 1, ..., N, \tau > 0) = \sum_{t=1}^{T}\sum_{i=1}^{n} \log p_i(\mathbf{w}_i^T\mathbf{x}(t) - \sum_{\tau>0} \alpha_\tau^i \mathbf{w}_i^T\mathbf{x}(t-\tau)) + T\log|\det \mathbf{W}| \tag{30}$$

Now, assume that the estimation of the autoregressive coefficients is decoupled from the estimation of the $\mathbf{w}_i$. In other words, the α_τ^i are estimated for fixed $\mathbf{w}_i$, and then the $\mathbf{w}_i$ are estimated for fixed α_τ^i, and so on. Assume further that the data is whitened and that $\mathbf{W}$ is constrained to be orthogonal, as is usual in ICA (Cardoso and Laheld, 1996; Hyvärinen and Oja, 1997; Hyvärinen, 1999a). Then the term $\log|\det \mathbf{W}|$ is constant, and what is left is essentially an approximation of the sum of the negative entropies of the residuals (the p_i should here be adapted so that they correspond to the log-densities of the residuals). Maximization of the likelihood is thus essentially equivalent to minimizing the sum of the approximations of complexity of the n projection $\mathbf{w}^T\mathbf{x}(t)$. Thus our algorithm can be seen as a one-unit version of this ICA estimation problem.

5.2 Related algorithms

Another algorithm that combines nongaussianity and time-correlations was proposed for ICA estimation in (Müller et al., 1999). This was constructed heuristically by combining two estimation criteria, one measuring nongaussianity and another one measuring time-correlations. Using Kolmogoroff Complexity, we find here a principled way of combining these criteria. In particular, we find the optimal weight that we should give to the part measuring nongaussianity with respect to the part measuring time-correlations.

Moreover, our method uses the nongaussianity of the residuals, instead of the nongaussianity of the components. This is in line with the arguments advanced in (Hyvärinen, 1998a) where it was proposed that ICA should be applied on the innovation process instead of the original signals. The innovation process coincides in the present framework with the residual of an optimal (possibly nonlinear) predictor. In (Hyvärinen, 1998a) it was argued that the innovation process is likely to be more nongaussian and to have components that are mode independent than the original data, and thus ICA algorithms should give better estimation results when applied on the innovation process. Thus, measuring the nongaussianity of the residuals should improve the separation results.

A final point of difference between our method and that proposed in (Müller et al., 1999) is that our algorithm allows for deflationary estimation of components in the spirit of projection pursuit.

Finally, it is worth mentioning another ICA algorithm that combines time-structure with nongaussianity (Attias, 2000). This algorithm models the time structure in a very different way, using a hidden Markov model with discrete states. A potential drawback of this approach is that it leads to quite complicated computations. In contrast, using autocorrelations allows computationally much simpler methods.

6 Simulation results

We created four signals using an AR(1) model, with 5000 time points. The signals #1 and #2 were created with supergaussian innovations, and the signals #3 and #4 with gaussian innovations; all innovations had unit variance. The signals #1 and #3 had identical autoregressive coefficients (0.25), and therefore identical autocovariances; the signals #2 and #4 had identical coefficients (0.5) as well.

To be able to validate the results, we performed source separation experiments. The signals were mixed as in ICA, using random mixing matrices. The step size μ was taken equal to 1. The nonlinearity was chosen as $g(u) = \tanh(u)$.

Ordinary ICA or blind source separation methods based on nongaussianity would not be able to separate the two gaussian signals from each other. On the other hand, methods based on autocovariances would not be able to separate signals with identical autocovariances. Thus practically all ICA or source separation algorithm would fail with this data. The only exception, to our knowledge, is the algorithm in (Müller et al., 1999), as discussed in Section 5.2.

Figure 1 shows the convergence of our algorithm, using the same nonlinearity (tanh) for all components. The graph shows results for symmetric orthogonalization. For deflationary orthogonalization, we obtained similar results. In both cases, the algorithm correctly estimated the independent components, in around 10-15 iterations. Note that a single generic nonlinearity that corresponds to supergaussian residuals was able to separate both gaussian and supergaussian signals, which indicates that the method is robust with respect to the choice of nonlinearity in the much same way as ICA.

7 Conclusion

We introduced an extension of projection pursuit, in which the time structure of a time-series (time-dependent signal) is taken into account, instead of using the marginal distribution only. This is based on finding directions in which the coding complexity of the signal is minimized. This also provides a principled method for (possibly deflationary) estimation of independent components that are time-dependent. The coding complexity can be approximated by a combination of the nongaussianity and the variance of the residual of a linear autoregressive model, leading to a computationally simple algorithm.

References

Amari, S.-I., Cichocki, A., and Yang, H. (1996). A new learning algorithm for blind source separation. In *Advances in Neural Information Processing Systems 8*, pages 757–763. MIT Press, Cambridge, MA.

Atick, J. (1992). Entropy minimization: A design principle for sensory perception? *Int. Journal of Neural Systems*, 3:81–90. Supp. 1992.

Attias, H. (2000). Independent factor analysis with temporally structured sources. In *Advances in Neural Information Processing Systems*, volume 12. MIT Press.

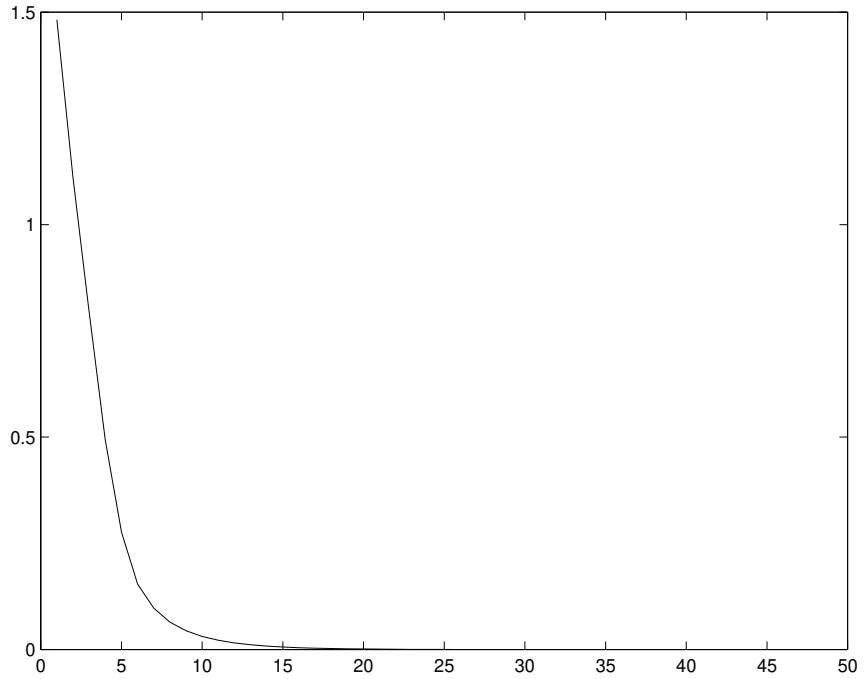

Figure 1: Convergence of complexity pursuit for artificially generated data. The error index shown is the squared distance of the separating matrix times the whitened mixing matrix (**WVA**) from the nearest (signed) permutation matrix. The median was taken over 10 runs with different random matrices and initial conditions.

Bell, A. and Sejnowski, T. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159.

Belouchrani, A., Meraim, K. A., Cardoso, J.-F., and Moulines, E. (1997). A blind source separation technique based on second order statistics. *IEEE Trans. on Signal Processing*, 45(2):434–44.

Cardoso, J. F. (2000). Entropic contrasts for source separation: Geometry and stability. In Haykin, S., editor, *Adaptive Unsupervised Learning, Vol. 1*. IEEE Press.

Cardoso, J.-F. and Laheld, B. H. (1996). Equivariant adaptive source separation. *IEEE Trans. on Signal Processing*, 44(12):3017–3030.

Cichocki, A. and Unbehauen, R. (1996). Robust neural networks with on-line learning for blind identification and blind separation of sources. *IEEE Trans. on Circuits and Systems*, 43(11):894–906.

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

Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory*. John Wiley & Sons.

Delfosse, N. and Loubaton, P. (1995). Adaptive blind separation of independent sources: a deflation approach. *Signal Processing*, 45:59–83.

Friedman, J. (1987). Exploratory projection pursuit. *J. of the American Statistical Association*, 82(397):249–266.

Friedman, J. H. and Tukey, J. W. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Trans. of Computers*, c-23(9):881–890.

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

Hochreiter, S. and Schmidhuber, J. (1999). Feature extraction through LOCOCODE. *Neural Computation*, 11(3):679–714.

Huber, P. (1985). Projection pursuit. *The Annals of Statistics*, 13(2):435–475.

Hyvärinen, A. (1998a). Independent component analysis for time-dependent stochastic processes. In *Proc. Int. Conf. on Artificial Neural Networks (ICANN'98)*, pages 135–140, Skövde, Sweden.

Hyvärinen, A. (1998b). New approximations of differential entropy for independent component analysis and projection pursuit. In *Advances in Neural Information Processing Systems*, volume 10, pages 273–279. MIT Press.

Hyvärinen, A. (1999a). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Trans. on Neural Networks*, 10(3):626–634.

Hyvärinen, A. (1999b). Sparse code shrinkage: Denoising of nongaussian data by maximum likelihood estimation. *Neural Computation*, 11(7):1739–1768.

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

Hyvärinen, A. and Oja, E. (1998). Independent component analysis by general nonlinear Hebbian-like learning rules. *Signal Processing*, 64(3):301–313.

Jolliffe, I. (1986). *Principal Component Analysis*. Springer-Verlag.

Jones, M. and Sibson, R. (1987). What is projection pursuit ? *J. of the Royal Statistical Society, ser. A*, 150:1–36.

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

Karhunen, J., Oja, E., Wang, L., Vigário, R., and Joutsensalo, J. (1997). A class of neural networks for independent component analysis. *IEEE Trans. on Neural Networks*, 8(3):486–504.

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

Molgedey, L. and Schuster, H. G. (1994). Separation of a mixture of independent signals using time delayed correlations. *Phys. Rev. Lett.*, 72:3634–3636.

Müller, K.-R., Philips, P., and Ziehe, A. (1999). $JADE_{TD}$: Combining higher-order statistics and temporal information for blind source separation (with noise). In *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, pages 87–92, Aussois, France.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *J. of Mathematical Biology*, 15:267–273.

Oja, E. (1997). The nonlinear PCA learning rule in independent component analysis. *Neurocomputing*, 17(1):25–46.

Pajunen, P. (1998a). Blind source separation using algorithmic information theory. *Neurocomputing*, 22:35–48.

Pajunen, P. (1998b). *Extensions of Linear Independent Component Analysis: Neural and Information-theoretic Methods*. PhD thesis, Helsinki University of Technology.

Pajunen, P. (1999). Blind source separation of natural signals based on approximate complexity minimization. In *Proc. Int. Workshop on Independent Component Analysis and Signal Separation (ICA'99)*, pages 267–270, Aussois, France.

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

Vigário, R., Särelä, J., Jousmäki, V., Hämäläinen, M., and Oja, E. (2000). Independent component approach to the analysis of EEG and MEG recordings. *IEEE Trans. Biomedical Engineering*, 47(5):589–593.