FINDING CLUSTERS IN INDEPENDENT COMPONENT ANALYSIS

Francis R. Bach

Computer Science Division
University of California
Berkeley, CA 94720, USA
fbach@cs.berkeley.edu

Michael I. Jordan

Computer Science Division
and Department of Statistics
University of California
Berkeley, CA 94720, USA
jordan@cs.berkeley.edu

ABSTRACT

We present a class of algorithms that find clusters in independent component analysis: the data are linearly transformed so that the resulting components can be grouped into clusters, such that components are dependent within clusters and independent between clusters. In order to find such clusters, we look for a transform that fits the estimated sources to a forest-structured graphical model. In the *non-Gaussian, temporally independent* case, the optimal transform is found by minimizing a contrast function based on mutual information that directly extends the contrast function used for classical ICA. We also derive a contrast function in the *Gaussian stationary* case that is based on spectral densities and generalizes the contrast function of Pham [22] to richer classes of dependency.

1. INTRODUCTION

Given a multivariate random variable x in $\mathbb{R}^m$, independent component analysis (ICA) consists in finding a linear transform W such that the resulting components of $s = Wx = (s_1, \ldots, s_m)^\top$ are as independent as possible (see, e.g., [13, 4, 10, 20]). ICA has been applied successfully to many problems where it can be assumed that the data are actually generated as linear mixtures of independent components, such as audio blind source separation or biomedical imagery.

In previous work [3], we relaxed the independence assumption to allow for richer classes of dependencies, namely we search for a linear transform W such that the components of $s = Wx = (s_1, \ldots, s_m)^\top$ can be well modeled by a tree-structured graphical model. We refer to this model as *tree-dependent component analysis* (TCA). In the same semiparametric likelihood framework as presented in [10], the contrast function that is minimized is a linear combination of mutual information terms that directly extends the classical contrast function for ICA, and we showed that many ICA algorithms and techniques for estimation can be extended to this richer class of dependencies.

In this paper, we extend this approach in two directions: first we allow the tree to be a *forest*; that is, a non-spanning tree with potentially any number of connected components. As shown in Section 2, such a graphical model is appropriate for the modeling of clusters, each cluster being one of the connected components. As was the case for TCA [3], the topology of the forest in not fixed in advance; rather, we search for the best possible forest in a manner analogous to the Chow-Liu algorithm [11]. We refer to this model as *forest-dependent component analysis* (FCA).

In addition, we extend the semiparametric approach of [3] to the Gaussian stationary case, making use of the notion of graphical models for time series [16]. Not surprisingly, the contrast function that we obtain is a linear combination of entropy rate terms that directly extends the contrast function presented by Pham [22].

In Section 3, we derive the contrast function for our semiparametric model and review techniques to estimate it in the temporally independent case, while in Section 4, we extend these results to the Gaussian stationary case. In Section 5, we give a precise description of our algorithms, and we present simulation results in Section 6.

2. MODELING CLUSTERS

In order to model clusters in ICA, it is necessary to model both *inter-cluster independence* and *intra-cluster dependence*. Forest-structured graphical models are particularly appropriate because they model exactly inter-cluster independence, while providing a rich but tractable model for intra-cluster dependence, by allowing an arbitrary pattern of tree-structured dependence within a cluster.

More precisely, let $x_1, \ldots, x_m$ denote m random variables. A forest T is a non-spanning undirected tree on the vertices $\{1, \ldots, m\}$, and a probability distribution $p(x)$ is said to *factorize* in T if and only if it can be written as $p(x) \propto \prod_{(u,v)\in T} \varphi_{uv}(x_u, x_v)$, where the potentials φ_{uv} are arbitrary functions (see e.g. [21]). Let $C_1, \ldots, C_k$ be the k connected components of T, as shown in Figure 1.

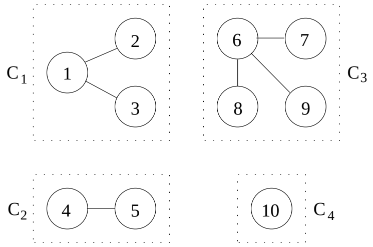

Fig. 1. A forest with 10 nodes and 4 clusters

Under certain positivity and regularity conditions, a distribution $p(x)$ factorizes in T if and only if (a) $x_{C_1}, \ldots, x_{C_k}$ are mutually independent, and (b) two variables in the same cluster are conditionally independent given all other variables in that cluster. The factorization of the distribution of each cluster of variables is flexible enough to model a wide variety of distributions, and still tractable enough to enable the easy computation of the semiparametric likelihood, as shown in the following section.

3. CONTRAST FUNCTIONS

In this paper, we wish to model the vector x using the model $x = As$, where A is an invertible mixing matrix and s factorizes in a forest T. We make no assumptions on the local bidimensional marginal distributions which are necessary in order to completely specify the model: A and T are the parametric parts, while the local bidimensional marginal distributions are nuisance parameters that are left unspecified. The semiparametric likelihood is obtained by maximizing first with respect to the nuisance parameters. As was the case for ICA, this is easily done, essentially because for trees, a "Pythagorean" expansion of the Kullback-Leibler divergence holds that generalizes the expansion for independent components (which is encoded by a graphical model with no edges).

We first restrict to the case where $W = A^{-1}$ is fixed. This is essentially the problem solved by the Chow-Liu algorithm [11]; however, we need to show how a prior probability on the topology of the forest can be included.

In the following sections, $p(x_u, x_v)$ and $p(x_u)$ will denote the marginalizations of $p(x)$ on (x_u, x_v) and x_u respectively. The Kullback-Leibler (KL) divergence between two distributions $p(x)$ and $q(x)$ is defined as $D(p \,||\, q) \triangleq E_{p(x)} \log \frac{p(x)}{q(x)}$. We will also work with the pairwise mutual information $I(x_u, x_v)$ between two variables x_u and x_v, defined as $I(x_u, x_v) = D(p(x_u, x_v) \,||\, p(x_u)p(x_v))$, and the m-fold mutual information defined as $I(x_1, \ldots, x_m) = D(p(x) \,||\, p(x_1) \cdots p(x_m))$.

3.1. Chow-Liu algorithm and forests

Given a forest T on the vertices $\{1, \ldots, m\}$, we let $\mathcal{D}^T$ denote the set of probability distributions $q(x)$ that factorize in T. We want to model $p(x)$ using a distribution $q(x)$ in $\mathcal{D}^T$. Trees are a special case of *decomposable models* and thus, for a given tree T, minimizing the KL divergence yields the following "Pythagorean" expansion of the KL divergence [21, 15]:

Theorem 1 *For a given forest T and a target distribution $p(x)$, we have, for all distributions $q \in \mathcal{D}^T$:*

$$D(p \,||\, q) = D(p \,||\, p_T) + D(p_T \,||\, q),$$

where $p_T(x) = \prod_{(u,v) \in T} \frac{p(x_u, x_v)}{p(x_u)p(x_v)} \prod_u p(x_u)$. In addition, $q = p_T$ minimizes $D(p \,||\, q)$ over $q \in \mathcal{D}^T$, and we have:

$$I^T(x) \triangleq \min_{q \in \mathcal{D}^T} D(p \,||\, q) = D(p \,||\, p_T) \quad (1)$$

$$= I(x_1, \ldots, x_m) - \sum_{(u,v) \in T} I(x_u, x_v). \quad (2)$$

We refer to $I^T(x)$ as the T-*mutual information*: it is the minimum possible loss of information when encoding the distribution $p(x)$ with a distribution that factorizes in T. It is equal to zero if and only if p does factorize in T.

In order to find the best forest T, we need to minimize $I^T(x)$ in Eq. (2), with respect to T. Without any restriction on T, since all mutual information terms are nonnegative, the minimum is attained at a spanning tree and thus the minimization is equivalent to a maximum weight spanning tree problem with weights $I(x_u, x_v)$, which can easily be solved in polynomial time by greedy algorithms (see e.g. [14]).

3.2. Prior on forests

In order to model forests, we include a prior term $w(T) = \log p(T)$ where $p(T)$ is a prior probability on the forest T which penalizes dense forests. In order to be able to minimize $I^T(x) - w(T)$, we restrict the penalty $w(T)$ to be of the form $w(T) = \sum_{(u,v) \in T} w^0_{uv} + f(\#(T))$, where w^0_{uv} is a fixed set of weights, f is a concave function and $\#(T)$ the number of edges in T. We use the algorithm outlined in Figure 2, with weights $w_{uv} = I(x_u, x_v) + w^0_{uv}$. Starting from the empty graph, while it is possible, incrementally pick a safe edge (i.e., one that does not create a cycle) such that the gain is maximal and positive. The following proposition shows that we obtain the global maximum:

Proposition 1 *If $J(T)$ has the form $J(T) = \sum_{(u,v) \in T} w_{uv} + f(\#(T))$ where $\{w_{uv}, u, v \in \{1, \ldots, m\}\}$ is a fixed set of weights, and f is a concave function, then the algorithm outlined in Figure 2 outputs the global maximum of $J(T)$.*

Natural priors are such that $w(T) \propto -\#(T)$ or $w(T) \propto -(\#(T))^\alpha$, $\alpha > 1$ (the earlier edges are penalized less than the later ones).

```
Input: weights $\{w_{uv}, u, v \in \{1, \ldots, m\}\}$
       $t_{max} \geqslant 0$, concave function $f(t)$

Algorithm:
  1. Initialization: $T = \varnothing, t = 0$
                     $E = \{1, \ldots, m\} \times \{1, \ldots, m\}$
  2. While $E \neq \varnothing$ and $t < t_{max}$
    a. Find $w_{u_0 v_0} = \max_{(u,v) \in E} w_{uv}$
    b. If $w_{u_0 v_0} + f(t+1) - f(t) > 0$
         $T \leftarrow T \cup (u_0, v_0), \quad t \leftarrow t + 1$
         $E \leftarrow \{e \in E, T \cup \{e\}$ has no cycles$\}$
       else $E = \varnothing$

Output: maximum weight forest $T$
```

Fig. 2. Greedy algorithm for the maximum weight forest problem, with a maximal number of edges t_{max}.

3.3. Tree-dependent component analysis

We now let the demixing matrix W vary. Let $\mathcal{D}^{W,T}$ denote the set of all such distributions. The KL divergence is invariant by invertible transformation of its arguments, so Theorem 1 can be easily extended [3]:

Theorem 2 *If x has distribution $p(x)$, then the minimum KL divergence between $p(x)$ and a distribution $q(x) \in \mathcal{D}^{W,T}$ is equal to the T-mutual information of $s = Wx$, that is:*

$$\begin{aligned} J(x, W, T) &\triangleq \min_{q \in \mathcal{D}^{W,T}} D(p \,||\, q) = I^T(s) \\ &= I(s_1, \ldots, s_m) - \sum_{(u,v) \in T} I(s_u, s_v). \end{aligned}$$

Therefore, in the semiparametric TCA approach, we wish to minimize $J(W, T) = J(x, W, T) - w(T)$ with respect to W and T. An approach to the estimation of this contrast function from a finite sample is presented in the next section while the optimization procedure is presented in Section 5.

3.4. Estimation of the contrast function

As in ICA, we do not know the density $p(x)$ and the estimation criterion must be replaced by empirical contrast functions. In [3], we describe three estimation methods, each of them extending classical ICA methods to the TCA model. We briefly summarize these methods here, which readily extend to forests.

Because the joint entropy of $s = Wx$ can be written as $H(s) = H(x) + \log|\det W|$, we only need one-dimensional entropies in order to minimize the term $I(s_1, \ldots, s_m)$. We also need two-dimensional entropies to estimate the pairwise mutual informations. In the first method we obtain $m(m-1)/2$ two-dimensional *kernel density estimates* (KDE), which can be done efficiently using the fast Fourier transform [23], and use the density estimates in order to compute the entropy terms. The overall complexity of evaluation of the contrast function is $O(m^2 N)$.

In the second approach, we extend the contrast function of [2], based on the *kernel generalized variance* (KGV), which is a Mercer kernel-based approximation to the mutual information. The evaluation can be performed in $O(mN)$.

Finally, in the third approach, we use *Gram-Charlier expansions* for one-dimensional and two-dimensional entropies, as laid out in [1]. The resulting contrast function involves fourth-order cumulants and is easily computed and minimized. Although cumulant-based contrast functions are usually less robust to outliers and source densities (see e.g. [2]), they provide a fast good initialization for the lengthier optimizations using KDE or KGV.

4. STATIONARY GAUSSIAN PROCESSES

In this section, we assume first that the sources are doubly infinite sequences of real valued observations $\{s_k(t), t \in \mathbb{Z}\}$, $k = 1, \ldots, m$. We model this multivariate sequence as a zero-mean multivariate Gaussian stationary process (we assume that the mean has been removed). We let $\Gamma(h)$, $h \in \mathbb{Z}$, denote the $m \times m$ matrix autocovariance function, defined as

$$\Gamma(h) = E[s(t)s(t+h)^\top].$$

We assume that $\sum_{-\infty}^{\infty} ||\Gamma(h)|| < \infty$, so that the spectral density matrix $f(\omega)$ is well-defined, as

$$f(\omega) = \frac{1}{2\pi} \sum_{h=-\infty}^{+\infty} \Gamma(h) e^{-ih\omega}.$$

For each ω, $f(\omega)$ is an $m \times m$ Hermitian matrix. In addition the function $\omega \mapsto f(\omega)$ is 2π-periodic.

4.1. Entropy rate of Gaussian processes

The entropy rate of a process s is defined as [15]:

$$H(s) = \lim_{T \to \infty} \frac{1}{T} H(s(t), \ldots, s(t+T)).$$

In the case of Gaussian stationary processes, the entropy rate can be computed using the spectral density matrix (due to an extension of Szegö's theorem to multivariate processes, see e.g. [18]):

$$H(s) = \frac{1}{4\pi} \int_{-\pi}^{\pi} \log\det[4\pi^2 e f(\omega)] d\omega.$$

Note that this is an analog of the expression for the entropy of a Gaussian vector z with covariance matrix Σ, where $H(z) = \frac{1}{2}\log\det[2\pi e \Sigma]$.

By the usual linear combination of entropy rates, the mutual information between process can be defined. Also, we can express the entropy rate of the process $x = Vs$, where V is a $d \times m$ matrix, using the spectral density of s

$$H(Vs) = \frac{1}{4\pi} \int_{-\pi}^{\pi} \log\det[4\pi^2 e V f(\omega) V^\top] d\omega.$$

4.2. Graphical model for time series

The graphical model framework can be extended to multivariate time series [7, 16]. The dependencies that are considered are between whole time series, that is between the entire sets $\{s_i(t), t \in \mathbb{Z}\}$, for $i = 1, \dots m$. If the process is jointly Gaussian stationary, then most of the graphical model results for Gaussian variables can be extended. In particular, maximum likelihood estimation of spectral density matrices in decomposable models decouples and is equivalent to equating local spectral density matrices. As we show in the next section, this enables Theorem 1 and Theorem 2 to be extended to the time series case.

4.3. Contrast function

Let x be a multivariate time series $\{x_k(t), t \in \mathbb{Z}\}$, $k = 1, \dots, m$. We wish to model the variable x using the model $x = As$, where A is an invertible mixing matrix and s is a Gaussian stationary time series that factorizes in a forest T. Letting $W = A^{-1}$, we let $\mathcal{D}_{stat}^{W,T}$ denote the set of all such distributions. We state without proof the direct extension of Theorem 2 to time series (W_u denotes the u-th row of W):

Theorem 3 *If x has a distribution with spectral density matrix $f(\omega)$, then the minimum KL divergence between $p(x)$ and a distribution $q(x) \in \mathcal{D}_{stat}^{W,T}$ is equal to the T-mutual information of $s = Wx$, that is:*

$$J(f, T, W) \triangleq I^T(f, W) = I(f, W) - \sum_{(u,v)\in T} I_{uv}(f, W), \tag{3}$$

where

$$I(f, W) \triangleq -\frac{1}{4\pi}\int_{-\pi}^{\pi} \log \frac{\det W f(\omega) W^\top}{W_1 f(\omega) W_1^\top \cdots W_m f(\omega) W_m^\top} d\omega$$

is the m-fold mutual information between $s_1, \dots, s_m$ and

$$I_{uv}(f, W) \triangleq \\ -\frac{1}{4\pi}\int_{-\pi}^{\pi} \log \left\{ 1 - \frac{\left(W_u f(\omega) W_v^\top\right)^2}{W_u f(\omega) W_u^\top \cdot W_v f(\omega) W_v^\top} \right\} d\omega$$

is the pairwise mutual information between s_u and s_v.

Thus, the goal of FCA is to minimize $I^T(f, W)$ in Eq. (3) with respect to W and T; in our simulations, we refer to this contrast function as the STAT contrast function.

4.4. Estimation of the spectral density matrix

In the presence of a finite sample $\{x(t), t = 0, \dots, N-1\}$, we use the *smoothed periodogram* [8] in order to estimate the spectral density matrix at points $\omega_j = 2\pi j/N$, $\omega_j \in [-\pi, \pi]$. At those frequencies, the periodogram is defined as

$$I_N(\omega_j) = \frac{1}{N}\left(\sum_{t=1}^N x_t e^{-it\omega_j}\right)\left(\overline{\sum_{t=1}^N x_t e^{-it\omega_j}}\right)^\top$$

(we assume the means have been previously removed from the data), and can readily be computed using m fast Fourier transforms (FFT). We use the following estimated spectral density matrices

$$\hat{f}(\omega_k) = \frac{1}{N}\sum_{j=0}^{N-1} W(j) I_N(\omega_{j+k}), \tag{4}$$

where $W(j)$ is a smoothing window that is required to be symmetric and sum to one. In our simulations, we took the window $W(j) = (2p+1)^{-1}$ for $|j| \leqslant p$, and 0 otherwise. Note that if the number of samples N tends to infinity with $p(N)$ tending to infinity such that $p(N)/N \to 0$, then Eq. (4) provides a consistent estimate of the spectral density matrix [8].

Our contrast function involves integrals of the form

$$\int_{-\pi}^{\pi} B(f(\omega)) d\omega.$$

They can be estimated by Riemannian sums using estimated values of f at $\omega_j = 2\pi j/N$, $\omega_j \in [-\pi, \pi]$, from Eq. (4). However, in large samples, we subsample (using a linear filter) the estimated density matrix to a grid of size $d = 50$, and use the following approximation:

$$\int_{-\pi}^{\pi} B(f(\omega)) d\omega \approx \frac{2\pi}{d}\sum_{i=1}^{d} B(f(\omega_i)).$$

5. OPTIMIZATION

5.1. Identifiability

For the ICA model, it is well known that in the temporally independent case the model is identifiable up to permutation and scaling, if and only if there is at most one Gaussian source, and that in the Gaussian stationary case, the model is identifiable up to permutation and scaling if and only if the sources have linearly independent autocovariance functions [20]. It is possible to show that—for a given forest C—a necessary condition for identifiability up to permutation and scaling of the demixing matrix W is that the components are non-Gaussian in the temporally independent case, and have linearly independent autocovariance functions in the Gaussian stationary case. However this condition is not sufficient because there are additional invariances beyond permutation and scaling. Indeed, for a given set of clusters, as pointed out in [9], only those subspaces spanned by the corresponding rows of W are identifiable, namely if the subspace has p dimensions, they can be premultiplied by any linear transform on p dimensions.

We deal with the scaling invariance by imposing a unit norm constraint on the rows of W, while we ignore during the optimization (but not during the evaluation stage, in Section 6) the invariance by permutations or by global linear transform of clusters.

5.2. Algorithm

Our model is estimated by minimizing the contrast function $F(W, T) = J(x, W, T) - w(T)$, where $w(T)$ is the log-

```
Input: data $\{x\} = \{x^1, \dots, x^N\}, \forall n,\ x^n \in \mathbb{R}^m$

Algorithm:
  1. Initialization: $T = \varnothing$, $W$ random
  2. For $i = 0$ to $m - 1$
    a. While $F(W, T)$ is decreasing
      1. Find best tree $T$ with $i$ edges, for fixed $W$
      2. Compute gradient of $F$ with respect to $W$
      3. Perform line search in $W$, in direction of
         the gradient
    b. $W_i = W$, $T_i = T$, $J_i = J$
  3. Find $i^* = \arg\max_i J_i$

Output: demixing matrix $W = W_{i^*}$, forest $T = T_{i^*}$
```

Fig. 3. The TCA algorithm

prior on the tree T and $J(x, W, T)$ is the contrast function that depends on the forest T, the demixing matrix W and the data x, defined in Theorem 2 or Theorem 3, using the estimation methods presented in Section 3.4 and Section 4.4. The algorithm is presented in Figure 3.

In order to perform the optimization of $F(W, T)$, we alternate minimization with respect to T, using a greedy algorithm for the maximum weight spanning tree problem as presented in Section 3.2, and minimization with respect to W, using steepest descent with line search. Note that since we constrain each row of W to have unit norm, our search space for W is a product of m spheres. On each sphere we perform line search along the geodesic, as detailed in [17]. To compute the gradient with respect to W, we use either first-order differences (KGV, KDE, CUM) or exact computations (STAT), but in principle, exact computations can be carried through for all four methods.

The procedure converges to a local minimum with respect to W. In order to deal with multiple local minima for the pair (W, T), an efficient heuristic is to start from an ICA solution (zero edges) and incrementally increase the number of allowed edges from zero to $m - 1$, as we describe in Figure 3.

6. SIMULATIONS

In all of our simulations, data were generated from q independent clusters $C_1, \dots, C_q$, and then rotated by a random but known matrix B. We measure the demixing performance by comparing W to $V = B^{-1}$.

6.1. Performance metric

In the case of ICA, the only invariances are invariances by permutation or scaling, which can be taken care of by a simple metric. Indeed, what needs to be measured is how much $A = WV^{-1}$ differs from a diagonal matrix, up to permutation. In our case, however, we need to measure how much A differs from a block diagonal matrix, up to permutation.

We first build the $m \times m$ *cost matrix* B as follows: for any $i \in \{1, \dots, m\}$ and $j \in C_k$, we have

$$B_{ji} = 1 - (\textstyle\sum_{p \in C_k} |A_{pi}|)/(\sum_{p=1}^m |A_{pi}|),$$

which is the cost of assigning component i to the cluster C_k. For each permutation σ over m elements, we define the cost of the assignment of components to clusters defined by σ to be $e(\sigma) = \sum_i B_{\sigma(i)i}$. Finally, the performance metric is defined as the minimum of $e(\sigma)$ over all permutations:

$$e = \max_\sigma e(\sigma) = \max_\sigma \sum_i B_{\sigma(i)i},$$

which can be computed in polynomial time by the Hungarian method [6]. e is always between 0 and m and is equal to 0 if and only if V is equivalent to W.

6.2. Comparisons

For temporally independent sources, we compare our algorithm—with the three different contrast functions CUM, KDE, KGV—to two ICA algorithms JADE [10], and FastICA [20].

For Gaussian stationary sources, we compare our algorithm to three ICA algorithms for time series, SOBI [5], TDSEP [24] and an algorithm that minimizes with respect to W the contrast function used by Pham [22]. This contrast function corresponds exactly to our contrast function J^{STAT} when no edges are allowed in the graph; that is, $J^{STAT}(f, W, \varnothing)$.

6.3. Temporally independent sources

We generated patterns of components as follows: given m variables and q components, we generated component sizes from a multinomial distribution. For each component we generated N iid samples from a mixture of two Gaussian with means μ_+ and μ_- and covariance matrices Σ_+ and Σ_-, where $\mu_\pm = \pm a u$ (u is a random unit norm vector and a is sampled from a Gamma distribution) and $\Sigma_\pm = U_\pm D_\pm U_\pm^\top$ (the diagonal elements of $D_\pm$ are sampled from a Dirichlet distribution and the orthogonal matrices are both random). Then the data were rotated by a random orthogonal matrix.

We performed simulations with various numbers of sources, from $m = 4$ to $m = 8$. We report results obtained from 20 replications in Table 1. The FCA methods recover the components better than the “plain” ICA algorithms. Note that if the distribution of a component can itself be modeled by a local ICA model, then such a component should be retrieved perfectly by an ICA algorithm. Since the sets of random distributions that we use contain such distributions, the average performance of all the algorithms (both ICA and FCA) is relatively good. These remarks also apply to the simulations with stationary Gaussian processes.

Table 1. (Top) Results for temporally independent sources. (Bottom) Results for Gaussian stationary sources.

m	Jade	FastICA	FCA-Cum	FCA-Kde	FCA-Kgv
4	0.6	0.65	0.25	0.15	0.14
6	1.3	1.2	0.7	0.51	0.5
8	2.4	2.5	1.1	0.9	0.9

m	Sobi	Tdsep	Pham	FCA
4	0.11	0.12	0.06	0.02
6	0.27	0.25	0.10	0.06
8	0.8	0.8	0.28	0.20
12	1.1	1.2	0.39	0.25
16	1.5	1.4	1.21	1.02

6.4. Stationary Gaussian processes

We generate patterns of components as before, but the data are generated from causal autoregressive models, with random coefficients. We performed simulations with various numbers of sources, from $m = 4$ to $m = 12$. We report results obtained from 20 replications in Table 1, where, as in the temporally independent case, our algorithm outperforms the extant ICA algorithms.

7. CONCLUSION

We have presented algorithms that find clusters in an independent component analysis, by explicitly modeling the sources with a forest-structured graphical model. The forest T and the demixing matrix W are determined by minimizing contrast functions within a semiparametric estimation framework, for temporally independent non-Gaussian data or for stationary Gaussian processes. Searching for the optimal forest enables us to determine the number and sizes of components, which was not feasible in previously proposed approaches to finding clusters [9, 19].

Although we have limited ourselves to a generalization of ICA that allows forest-structured dependency among the sources, it is clearly of interest to make use of the general graphical model toolbox and consider broader classes of dependency.

Acknowledgements

We would like to acknowledge support from NSF grant IIS-9988642, ONR MURI N00014-00-1-0637. The MATLAB codes for the ICA algorithms that we used for comparison were downloaded from the webpages of their respective authors, except for SOBI, for which we used the implementation in ICALAB [12].

8. REFERENCES

[1] S. Akaho, Y. Kiuchi, and S. Umeyama. Multimodal independent component analysis. In *Proc. IJCNN*, 1999.

[2] F. R. Bach and M. I. Jordan. Kernel independent component analysis. *J. of Machine Learning Research*, 3:1–48, 2002.

[3] F. R. Bach and M. I. Jordan. Tree-dependent component analysis. In *Proc. UAI*, 2002.

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

[5] A. Belouchrani, K. Abed Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique based on second order statistics. *IEEE Sig. Proc.*, 45(2):434–44, 1997.

[6] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientic, 1997.

[7] D. Brillinger. Remarks concerning graphical models for time series and point processes. *Rev. de Econ.*, 16:1–23, 1996.

[8] P. J. Brockwell and R. A. Davis. *Time Series: Theory and Methods*. Springer-Verlag, 1991.

[9] J.-F. Cardoso. Multidimensional independent component analysis. In *Proc. ICASSP*, 1998.

[10] J.-F. Cardoso. High-order contrasts for independent component analysis. *Neural Computation*, 11(1):157–192, 1999.

[11] C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Inf. Theory*, 14:462–467, 1968.

[12] A. Cichocki, S. Amari, and K. Siwek. ICALAB toolboxes. `http://www.bsp.brain.riken.go.jp/ICALAB`.

[13] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.

[14] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1989.

[15] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[16] R. Dahlhaus. Graphical interaction models for multivariate time series. *Metrika*, 51:157–172, 2000.

[17] A. Edelman, T. A. Arias, and S. T. Smith. The geometry of algorithms with orthogonality constraints. *SIAM J. Mat. Anal. Appl.*, 20(2):303–353, 1999.

[18] E. Hannan. *Multiple Time Series*. Wiley, 1970.

[19] A. Hyvärinen and P. Hoyer. Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705–1720, 2000.

[20] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. John Wiley & Sons, 2001.

[21] S. L. Lauritzen. *Graphical Models*. Clarendon Press, 1996.

[22] D. T. Pham. Mutual information approach to blind separation of stationary sources. *IEEE Inf. Theory*, 48(7):1935–1946, 2002.

[23] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, 1985.

[24] A. Ziehe and K.-R. Müller. TDSEP - an efficient algorithm for blind separation using time structure. In *Proc. ICANN*, 1998.