Journal of Machine Learning Research 4 (2003) 1205-1233 Submitted 11/02; Published 12/03

Beyond Independent Components: Trees and Clusters

Francis R. Bach FBACH@CS.BERKELEY.EDU
Computer Science Division
University of California
Berkeley, CA 94720, USA

Michael I. Jordan JORDAN@CS.BERKELEY.EDU
Computer Science Division and Department of Statistics
University of California
Berkeley, CA 94720, USA

Editors: Te-Won Lee, Jean-François Cardoso, Erkki Oja and Shun-ichi Amari

Abstract

We present a generalization of independent component analysis (ICA), where instead of looking for a linear transform that makes the data components independent, we look for a transform that makes the data components well fit by a tree-structured graphical model. This *tree-dependent component analysis (TCA)* provides a tractable and flexible approach to weakening the assumption of independence in ICA. In particular, TCA allows the underlying graph to have multiple connected components, and thus the method is able to find "clusters" of components such that components are dependent within a cluster and independent between clusters. Finally, we make use of a notion of graphical models for time series due to Brillinger (1996) to extend these ideas to the temporal setting. In particular, we are able to fit models that incorporate tree-structured dependencies among multiple time series.

Keywords: Independent component analysis, graphical models, blind source separation, time series, semiparametric models

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., Comon, 1994, Bell and Sejnowski, 1995, Hyvärinen et al., 2001b). It has been applied successfully to many problems in which it can be assumed that the data are actually generated as linear mixtures of independent components, such as problems in audio blind source separation or biomedical image processing. It can also be used as a general multivariate density estimation method where, once the optimal transformation W has been found, only univariate density estimation needs to be performed. In this paper, we generalize these ideas: 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. The topology of the tree is not fixed in advance; rather, we search for the best possible tree in a manner analogous to the Chow-Liu algorithm in supervised learning (Chow and Liu, 1968), which indeed serves as an inner loop in our algorithm. We refer to this methodology as *tree-dependent component analysis (TCA)*.

By weakening the assumption made by ICA that the underlying components are independent, TCA can be applied to a wider range of problems in which data are transformed by an unknown linear transformation. For example, in a musical recording, instruments are generally not mutually independent. Modeling their dependencies should be helpful in achieving successful demixing, and the TCA model provides a principled approach to solving this problem.

An important feature of the TCA framework is that the trees are allowed to have more than a single connected component. Thus the methodology applies immediately to the problem of finding "clusters" of dependent components— by defining clusters as connected components of a graphical model, we find a decomposition of the source variables such that components are dependent within clusters and independent between clusters. In contrast to existing clustering algorithms (Hyvärinen and Hoyer, 2000), our approach does not require the number and sizes of components to be fixed in advance. (See Section 9.3 for simulations dedicated to this situation).

As with ICA, the TCA approach can also be used as an efficient method for general multivariate density estimation. Indeed, once the linear transform W and the tree T are found, we need only perform *bivariate* density estimation, skirting the curse of dimensionality while obtaining a flexible model. The models that we obtain using these two stages—first find W and T, then estimate densities—are fully tractable for learning and inference.

We treat TCA as a *semiparametric model* (Bickel et al., 1998), in which the actual marginal and conditional distributions of the tree-dependent components are left unspecified. In the simpler case of ICA, it is known that if the data are assumed independently and identically distributed (*iid*), then maximizing the semiparametric likelihood is equivalent to minimizing the mutual information between the estimated components (Cardoso, 1999). In Section 3, we review the relevant ICA results and we extend this approach to the estimation of W and T in the TCA model with *iid* data, deriving an expression for the semiparametric likelihood which involves a number of pairwise mutual information terms corresponding to the cliques in the tree. As in ICA, to obtain a criterion that can be used to estimate the parameters in the model from data (a "contrast function"), we approximate this population likelihood. In particular, in this paper, we derive three contrast functions that are direct extensions of ICA contrast functions. In Section 5.1, we use kernel density estimation to provide plug-in estimates of the necessary mutual information terms. In Section 5.2, we use entropy estimates based on Gram-Charlier expansions. Finally, in Section 5.3, we show how the "kernel generalized variance" proposed in our earlier work on ICA (Bach and Jordan, 2002) can be extended to approximate the TCA semiparametric likelihood. Finally, once the contrast functions are defined, we are faced with the minimization of a function $F(W,T)$ with respect to the matrix W and the tree T. We first perform a partial minimization of F with respect to T, via the Chow-Liu algorithm. This yields a continuous piecewise differentiable function of W, a function that we minimize by coordinate descent, taking advantage of the special structure of the manifold in which the matrix W lies. The algorithm is presented in Section 6.

Although ICA algorithms which assume that the samples are exchangeable can work in settings where the data have an evident temporal structure, it is preferable to take into account this temporal information. In this situation, the classical requirement for non-Gaussianity of the sources is not necessary and it is possible to base contrast functions on second-order information (Belouchrani et al., 1997, Pham, 2002), essentially modeling the sources as stationary Gaussian processes. In Section 8, we extend the semiparametric TCA approach to this setting, making use of the notion of graphical models for time series (Brillinger, 1996, Dahlhaus, 2000). Not surprisingly, the contrast function that we obtain is a linear combination of entropy rate terms that directly extends the contrast function for mutually independent time series presented by Pham (2002).

The TCA model has interesting properties that differ from the classical ICA model. First, in the Gaussian case, whereas the ICA model reduces to simply finding uncorrelated components (with a lack of identifiability for specific directions in the manifold of uncorrelated components), in the TCA case there are additional solutions beyond uncorrelated components. Second, in the general non-Gaussian case, additional identifiability issues arise. We study these issues in Section 4. In Section 7, we discuss the problem of density estimation once the optimal linear transform W has been obtained. Finally, in Section 9, we illustrate our algorithm with simulations on synthetic examples, some of which do, and some of which do not, follow the TCA model.

2. Graphical Models

In this section we provide enough basic background on tree-structured graphical models so as to make the paper self-contained. For additional discussion of graphical models, see Lauritzen (1996) and Jordan (2002).

The graphical model formalism sets up a relationship between graphs and families of probability distributions. Let $G(\mathcal{V},\mathcal{E})$ be a graph, with nodes $\mathcal{V}$ and edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. To each node $v \in \mathcal{V}$, we associate a random variable X_v. Probability distributions on the set of variables (X_v) are defined in terms of products over functions on "connected subsets" of nodes. There are two main categories of graphical models—*directed graphical models* (which are based on directed acyclic graphs), and *undirected graphical models* (which are based on undirected graphs). In the case of a directed graph, the basic "connected subset" of interest is a node and its parents, while in the undirected case, the basic "connected subset" is a *clique* (a clique is a fully-connected subset of nodes, that is, such that every pair of nodes is connected). These two notions of connectedness are the same in the case of trees, and we can work with either the directed or undirected formalism without loss of generality.[1] We work with undirected trees throughout most of the current paper, although we also make use of directed trees.

Two extreme cases are worth noting. First, a graph with no edges asserts that the random variables are mutually independent—the classical setting for ICA. Second, the complete graph makes no assertions of independence, and the corresponding family of probability distributions is the set of all distributions.

Tree-structured graphical models lie intermediate between these extremes. They provide a reasonably rich family of probabilistic dependencies, while restricting the family of distributions in meaningful ways (in particular, the inference and estimation problems associated with trees are tractable computationally). See Willsky (2002) for a recent review covering some of numerous applications of tree-structured graphical models to problems in signal processing, estimation theory, machine learning and beyond.

2.1 Undirected Trees

We begin with graph-theoretic definitions and then turn to the probabilistic definitions. A *tree* $T(\mathcal{V},\mathcal{E})$ is an undirected graph in which there is at most a single path between any pair of nodes. Note that we explicitly allow the possibility that there is *no* path between a particular pair of nodes;

1. More formally, for any family of probability distributions represented by a directed tree, there is a corresponding family of probability distributions represented by an undirected tree, and vice versa. It is also worth noting that trees are a special case of a broader class of models known as *decomposable models* that are also representable by either directed or undirected graphs. Decomposable models share many of the important properties of trees.

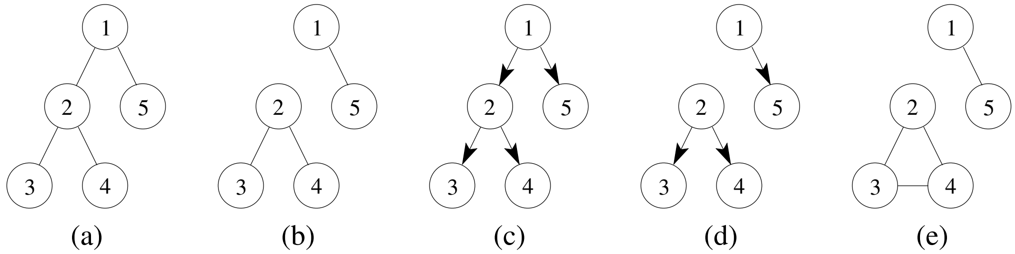

Figure 1: Graphical models on five nodes: (a) undirected spanning tree, (b) undirected non-spanning tree, (c) directed spanning tree, (d) directed non-spanning tree, (e) undirected clusters.

that is, the tree may consist of multiple connected components (such a graph is sometimes referred to as a "forest"). A *spanning tree* is a tree that includes all of the nodes in the graph (see Figure 1 for examples).

A probability distribution on an undirected graphical model is generally defined as a product of functions defined on the cliques of the graph. The cliques in an undirected tree are pairs of nodes and single nodes; thus, to parameterize a probability distribution on a tree, we define functions $\psi_{uv}(x_u,x_v)$ and $\psi_u(x_u)$, for $(u,v) \in \mathcal{E}$, and $u \in \mathcal{V}$, respectively. These *potential functions* are nonnegative, but otherwise arbitrary. The joint probability is defined as a product of potential functions:

$$p(x) = \frac{1}{Z} \prod_{(u,v)\in\mathcal{E}} \psi_{uv}(x_u,x_v) \prod_{v\in\mathcal{V}} \psi_u(x_u), \tag{1}$$

where Z is a normalization factor. Ranging over all possible choices of potential functions, we obtain a family of probability distributions associated with the given tree. We say that the members of this family *factorize according to T*.

Given a graphical model, we want to be able to compute marginal and conditional probabilities of interest—the *probabilistic inference problem*. An algorithm known as the *junction tree algorithm* provides a general framework for solving the probabilistic inference problem for arbitrary graphs, but scales exponentially in the size of the maximal clique. In the case of trees, the junction tree algorithm reduces to a simpler algorithm known as the *sum-product algorithm* or *belief propagation algorithm*. This algorithm yields marginal probabilities $p(x_u)$ and $p(x_u,x_v)$, for all $(u,v) \in \mathcal{E}$ and for all $u \in \mathcal{V}$, in time proportional to the number of edges in the tree (Pearl, 2000). The existence of this algorithm is one of the major justifications for restricting ourselves to trees.

The functions $\psi_{uv}(x_u,x_v)$ and $\psi_u(x_u)$ are arbitrary nonnegative functions, and need have no direct relationship to the marginal probabilities $p(x_u,x_v)$ and $p(x_u)$. It turns out, however, that it is also possible to express the joint probability on an undirected tree in terms of these marginals. Thus, for any distribution defined via Equation (1), it can be shown that we can also write the distribution in the following form:[2]

$$p(x) = \prod_{(u,v)\in\mathcal{E}} \frac{p(x_u,x_v)}{p(x_u)p(x_v)} \prod_{u\in\mathcal{V}} p(x_u). \tag{2}$$

Note that this is a special case of Equation (1) in which the normalization factor is equal to one.

2.2 Directed Trees

A *directed tree* is a directed graph in which every node has at most one parent. We can obtain a directed tree from an undirected tree by choosing a *founder* node in each connected component and orienting all edges in that component to point away from the founder (in Figure 1, the founders were chosen to be the nodes numbered 1 and 2).

Probability distributions on directed graphs are defined in general in terms of products of conditional and marginal probabilities, one for each node. In the case of directed tree, letting $\mathcal{F}$ denote the set of founders, and letting $\mathcal{U} = \mathcal{V}\backslash\mathcal{F}$ denote the remaining nodes, we obtain the following definition of the joint probability:

$$p(x) = \prod_{f\in\mathcal{F}} p(x_f) \prod_{u\in\mathcal{U}} p(x_u \mid x_{\pi_u}), \tag{3}$$

where π_u denotes the parent of node u.

Note that Equation (3) is a special case of Equation (1), with normalizing constant Z equal to one. This supports our earlier contention that there are no differences in representational power between undirected and directed trees.

As we will see in Section 7, however, the fact that the normalizing constant Z is equal to one in the directed case is convenient for estimating the densities associated with a tree (once the matrix W and the tree T have been determined). A log likelihood based on Equation (3) decouples into a sum of logarithms of individual marginal and conditional probabilities, allowing us to decompose the density estimation problem into separate estimation problems.

2.3 Conditional Independence

Apart from being parsimonious probabilistic models, directed and undirected graphical models have an equivalent characterization based on conditional independence and graph separation. In the case of trees, this is particularly simple: under mild assumptions on the joint probability density of the random variables, the variables $(x_1,\ldots,x_m)$ factorize according to the tree T if and only if, for any three subsets A, B, C of $\mathcal{V}$, such that C separates A from B in the graph, then the set of variables $x_A = \{x_i, i\in A\}$, $x_B = \{x_i, i \in B\}$ and $x_C = \{x_i, i\in B\}$ are such that x_A is independent from x_B given x_C (in the model (a) in Figure 1, we have for example $\{x_3,x_4\}$ independent from x_5 given x_2). In particular, if there are two separate connected components, variables in different connected components are independent from one another.

2.4 Non-Spanning Trees and Clusters

An exact (undirected) graphical model for clusters of variables would have no edges between nodes that belong to different clusters and would be fully-connected within a cluster (as illustrated in

2. An intuitive appreciation of Equation (2) can be obtained by considering the special case of a chain, in which Equation (2) reduces to the familiar Markovian product of conditionals. The proof is essentially a by-product of the proof of correctness of the junction tree algorithm; for details, see Lauritzen (1996).

Figure 1 for the clusters $\{1,5\}$ and $\{2,3,4\}$). Using a non-spanning tree models 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. In particular, learning the best possible non-spanning tree that fits a given distribution provides a way to learn the number and size of the clusters for that particular distribution.

3. Semiparametric Maximum Likelihood

In this section we derive the objective function that will be minimized to determine the demixing matrix W and the tree T in the TCA model with *iid* samples. Let $x = (x_1, \ldots, x_m)^\top$ be an m-component random vector with joint "target" distribution $p(x)$. Our primary goal is to minimize the Kullback-Leibler (KL) divergence $D(p||q) = E_{p(x)} \log \frac{p(x)}{q(x)}$ between $p(x)$ and our model $q(x)$ of this vector. Typically, $p(x)$ will be the empirical distribution associated with a training set and minimizing the KL divergence is well known to be equivalent to maximizing the likelihood of the data. In a semiparametric model, the parameters of interest—the matrix W and the tree T in our case—do not completely specify the distribution $q(x)$; the additional (infinite-dimensional) set of parameters that would be needed to complete the specification are left unspecified. More precisely, we define our objective function for T and W to be a "profile likelihood"—the minimum of the KL divergence $D(p||q)$ with respect to arbitrary distributions of the source components s_i (Murphy and van der Vaart, 2000). As we will show, it turns out that this criterion can be expressed in term of mutual information terms relating components that are neighbors in the tree T. We first review the classical ICA setting where the components s_i are assumed independent. Then, we describe the case where W is fixed to identity and T can vary—this is simply the tree model presented by Chow and Liu (1968). We finally show how the two models can be combined and generalized to the full TCA model where both W and T can vary.

In the following sections we generally label the nodes with integers; thus, we let $\mathcal{V} = \{1,2,\ldots,m\}$. We will 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 $I(x_1,\ldots,x_m)$, defined as $I(x_1,\ldots,x_m) = D(p(x)||p(x_1)\cdots p(x_m))$. Finally, all mutual informations, entropies and expectations are relative to the distributions $p(x)$ or $p(s)$ unless otherwise noted.

3.1 ICA Model

The classical ICA model takes the form $x = As$ where A is an invertible mixing matrix and s has independent components. If the variable x is Gaussian, then ICA is not identifiable, that is, the optimal matrix $W = A^{-1}$ is only defined up to an orthogonal matrix. Thus, non-Gaussianity of components is a crucial assumption for a full ICA solution to be well-defined, and we also make this assumption throughout the current paper.[3] However, the Gaussian case is still of interest because it allows us to reduce the size of the search space for W (see Section 4.2 and Appendix A for details). In addition, the Gaussian contrast function is a basis for the kernel generalized variance empirical contrast function presented in Section 5.3 and the Gaussian stationary time series contrast function in Section 8.

3. The identifiability of the ICA model has been discussed by Comon (1994). Briefly, the matrix A is identifiable, up to permutation and scaling of its columns, if and only if at most one of the component distributions $p(s_i)$ is Gaussian. In Section 4.1, we study some additional invariances verified by the TCA model.

Given a random vector x with distribution $p(x)$ (not necessarily having independent components), the distribution $q(x)$ with independent components that is closest to $p(x)$ in KL divergence is the product $q(x) = p(x_1)\cdots p(x_m)$, and the minimum KL divergence is thus $D(p(x)||p(x_1)\cdots p(x_n))$, which is exactly the mutual information $I(x_1,\ldots,x_m)$.

We now turn to the situation where A can vary. Letting $W = A^{-1}$, we let $\mathcal{D}^W$ denote the set of all distributions $q(x)$ such that $s = Wx$ has independent components. Since the KL divergence is invariant under an invertible transformation, the best approximation to $p(x)$ by a distribution in $\mathcal{D}^W$ is obtained as the product of the marginals of $s = Wx$, which yields:

$$\min_{q\in\mathcal{D}^W} D(p||q) = I(s_1,\ldots,s_m). \tag{4}$$

Thus, in the semiparametric ICA approach, we wish to minimize the mutual information of the estimated components $s = Wx$. We will generalize Equation (4) to the TCA model in Section 3.4.

In practice, we do not know the density $p(x)$ and thus the estimation criteria must be replaced by functionals of the sample data, functionals that are referred to as "empirical contrast functions." Classical ICA contrast functions involve either approximations to the mutual information or alternative measures of dependence involving higher-order moments (Cardoso, 1999, Hyvärinen et al., 2001b, Pham, 2001b).

3.2 Chow-Liu Algorithm and T-Mutual Information

Given an undirected tree T on the vertices $\mathcal{V} = \{1,\ldots,m\}$, we let $\mathcal{D}^T$ denote the set of probability distributions $q(x)$ that factorize according to 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 is straightforward and yields the following "Pythagorean" expansion of the KL divergence (Jirousek, 1991):

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

$$D(p||q) = D(p||p_T) + D(p_T||q), \tag{5}$$

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

$$\begin{aligned} I^T(x) &= \min_{q\in\mathcal{D}^T} D(p||q) = D(p||p_T) \\ &= I(x_1,\ldots,x_m) - \sum_{(u,v)\in\mathcal{E}} I(x_u,x_v). \end{aligned} \tag{6}$$

Proof Let q be a distribution in $\mathcal{D}^T$, that is such that $q(x) = \prod_{(u,v)\in\mathcal{E}} \frac{q(x_u,x_v)}{q(x_u)q(x_v)} \prod_{u\in\mathcal{V}} q(x_u)$. Since the density of $q(x)$ is a product of functions of pairs of variables linked by an edge and of functions of single variables, and since $p(x)$ and $p_T(x)$ have the same marginal distributions on the cliques of the tree, we have $E_{p(x)} \log q(x) = E_{p_T(x)} \log q(x)$. Applied to p_T (which factorizes according to T), we get $E_{p(x)} \log p_T(x) = E_{p_T(x)} \log p_T(x)$. We thus have:

$$\begin{aligned}
D(p(x)||q(x)) &= E_{p(x)} \log p(x) - E_{p(x)} \log q(x) \\
&= E_{p(x)} \log p(x) - E_{p_T(x)} \log q(x) \\
&= E_{p(x)} \log p(x) - E_{p_T(x)} \log p_T(x) + E_{p_T(x)} \log p_T(x) - E_{p_T(x)} \log q(x) \\
&= E_{p(x)} \log p(x) - E_{p(x)} \log p_T(x) + E_{p_T(x)} \log p_T(x) - E_{p_T(x)} \log q(x) \\
&= D(p(x)||p_T(x)) + D(p_T(x)||q(x)),
\end{aligned}$$

which proves the Pythagorean identity; this in turn implies that the distribution in $\mathcal{D}^T$ with minimum KL divergence is indeed p_T. We can now compute $D(p||p_T)$ as follows (entropy and mutual information terms are computed using the distribution $p(x)$):

$$\begin{aligned}
D(p(x)||p_T(x)) &= E_{p(x)} \log p(x) - E_{p(x)} \log p_T(x) \\
&= -H(x) - \sum_{(u,v)\in\mathcal{E}} E_{p(x)} \log \frac{p(x_u,x_v)}{p(x_u)p(x_v)} - \sum_{u\in\mathcal{V}} E_{p(x)} \log p(x_u) \\
&= -H(x) - \sum_{(u,v)\in\mathcal{E}} I(x_u,x_v) + \sum_{u\in\mathcal{V}} H(x_u) \\
&= I(x_1,\ldots,x_m) - \sum_{(u,v)\in\mathcal{E}} I(x_u,x_v)
\end{aligned}$$

■

We refer to $I^T(x)$ in Equation (6) 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 according to T. Such a quantity can be defined for any directed or undirected graphical model and can be computed in closed form for all directed graphical models and all decomposable undirected graphical models (Dawid and Lauritzen, 1993, Friedman and Goldszmidt, 1998).

In order to find the best tree T, we need to minimize $I^T(x)$ in Equation (6) 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 (nonnegative) weights $I(x_u,x_v)$, which can be solved in polynomial time by greedy algorithms (see next section and Cormen et al., 1989).

3.3 Prior Distributions on Trees

In order to allow non-spanning trees in our model, 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 find a global minimum of $I^T(x) - w(T)$ using greedy algorithms, we restrict the penalty $w(T)$ to be of the form $w(T) = \sum_{(u,v)\in\mathcal{E}} w^0_{uv} + f(\#(T))$, where w^0_{uv} is a fixed set of weights, f is a concave function, and $\#(T)$ is 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:

Input: weights $\{w_{uv}, u, v \in \mathcal{V}\}$, concave function $f(t)$

Algorithm:

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

Output: maximum weight forest $T(\mathcal{V}, \mathcal{E})$

Figure 2: Greedy algorithm for the maximum weight forest problem.

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

The magnitude of the weights $w(T)$ provides a way to control the degree of sparsity of the resulting tree: if the weights w^0_{uv} are negative and have large magnitude, then additional edges are heavily penalized and the tree will tend to have a small number of edges. We can set the weights by casting the problem of finding the best forest as a model selection problem for graphical models and use classical model selection criteria, such as the Akaike information criterion (AIC) or the Bayesian information criterion (BIC) (Heckerman et al., 1995, Hastie et al., 2001, Bach and Jordan, 2003). In simulations, we used a constant negative penalty for each edge.

3.4 TCA Model

In TCA, we wish to model the variable x using the model $x = As$, where A is an invertible mixing matrix and s factorizes in a tree T. Letting $W = A^{-1}$, we 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:

Theorem 3 *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:*

$$\begin{aligned} J(x, W, T) &= \min_{q \in \mathcal{D}^{W,T}} D(p||q) = I^T(s) && (7) \\ &= I(s_1, \ldots, s_m) - \sum_{(u,v) \in \mathcal{E}} I(s_u, s_v). && (8) \end{aligned}$$

Therefore, in the semiparametric TCA approach, we wish to minimize $J(x, W, T)$ with respect to W and T.

As in ICA, we do not know the density $p(x)$ and the estimation criteria must be replaced by empirical contrast functions. In the TCA setting, it is important that we maintain a link with mutual information: indeed the interplay between the 2-fold and m-fold mutual information terms is crucial, making it possible to avoid overcounting or undercounting the pairwise dependencies. The contrast functions that we propose thus have such a link—our first two contrast function approximates the mutual information terms directly, and our third proposed contrast function has an indirect link to mutual information. Before describing these three contrast functions, we turn to the description of the main properties of the TCA model.

4. Properties of the TCA Model

In this section we describe some properties of the TCA model, relating them to properties of the simpler ICA model. In particular we focus on identifiability issues and on the Gaussian case.

4.1 Identifiability Issues

In the ICA model, it is well known that the matrix W can only be determined up to permutation or scaling of its rows. In the TCA model, we have the following set of indeterminacies. Note that this set of indeterminacies is only necessary and may not be sufficient in general.

- **Permutation of components.** W can be premultiplied by a permutation matrix without changing the value of $J(x,W,T)$, as long as the tree T is also permuted analogously. This implies that in principle we don't have to consider all possible trees, but just equivalence classes under vertex permutation. Our empirical contrast functions are invariant with respect to that invariance. Therefore, it can be safely, although slightly inefficiently, ignored.

- **Scaling of the components.** W can be premultiplied by any invertible diagonal matrix. Thus we can restrict our search to components that have unit variance.

- **Rotation of connected components of size two.** If the tree T is non-spanning and has more than one connected component, then for each component C of size two, W can be premultiplied by any linear transform that leaves all other components invariant and the component C globally invariant. When identifying clusters instead of forests, then, as shown by Cardoso (1998), the demixing matrix is identifiable only up to the p subspaces spanned by the set of rows corresponding to each of the p components. However, this does not hold in general in the tree-structured case which imposes further constraints for connected components of size larger than three. This invariance is a special case of the leaf mixing invariance that we now present.

- **Mixing of a leaf node with its parent.** For a given tree structure T, and a leaf node c, adding a multiple of the value of its parent p to the value of the leaf will not change the goodness of fit of the tree T (see Figure 3). Indeed, a leaf node is only present in the likelihood through the conditional probability $p(s_c|s_p)$, and thus we can equivalently model $p(s_c|s_p)$ or $p(\lambda s_c+\mu s_p|s_p)$ for any μ and any non-zero λ. The T-mutual information $I^T(s)$ is thus invariant under such transformations.

  While the first three identifiability issues are easily handled by simple conventions, this latter indeterminacy is not easily removed via a simple convention and cannot be ignored because the empirical contrast functions that we develop do not all respect the mixing invariance. We could deal with the issue by "normalizing" the relation between a leaf and its parent, for

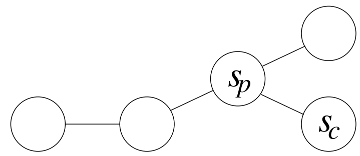

Figure 3: Leaf mixing invariance. See text for details.

example by requiring marginal decorrelation. However, this normalization depends on the tree, so it is not appropriate when comparing trees. Instead, we simply add a penalty term to our contrast functions, penalizing the correlation between components that are linked by an edge of the tree T (see Section 6 for details).

4.2 The Gaussian Case

In the ICA model, if the variable x is Gaussian, the solutions are the matrices W that make the covariance matrix of $s = Wx$ equal to identity. Thus, they are defined up to an orthogonal matrix R.

In the TCA model, with a fixed tree T, there is more than one covariance matrix that leads to a tree-structured graphical model with graph T for the underlying Gaussian random vector. For a Gaussian random vector, conditional independences can be read out from zeros in the inverse of the covariance matrix (e.g. Lauritzen, 1996). Applying this result to trees, we get

Proposition 4 *If $x = (x_1, \ldots, x_m)$ is Gaussian with covariance matrix Σ, it factorizes in the tree $T(\mathcal{V}, \mathcal{E})$ if and only if for all $(u,v) \notin \mathcal{E}$, we have $(\Sigma^{-1})_{uv} = 0$.*

Let $\mathcal{C}^T$ denote the set of all covariance matrices that respect these constraints. Note that it is possible to give a constructive description of this set, simply by writing down the factorization in any directed tree associated with the undirected tree T, using linear Gaussian conditional probability distributions (Luettgen et al., 1994). The number of degrees of freedom in such a construction is $2m-1$ if no constraint is imposed on the variance of the components, and $m-1$ if the components are constrained to have unit variance.

Finally, we can compute $I^T(\Sigma) = I^T(x)$ for a Gaussian variable x with covariance matrix Σ:

$$I^T(\Sigma) = I^G(\Sigma) - \sum_{(u,v)\in\mathcal{E}} I^G_{uv}(\Sigma), \tag{9}$$

where $I^G(\Sigma) = -\frac{1}{2}\log\frac{\det\Sigma}{\Sigma_{11}\cdots\Sigma_{mm}}$ is the m-fold mutual information between $x_1, \ldots, x_m$ and $I^G_{uv}(\Sigma) = -\frac{1}{2}\log\frac{\Sigma_{uu}\Sigma_{vv}-\Sigma_{uv}^2}{\Sigma_{uu}\Sigma_{vv}}$ is the pairwise mutual information between x_u and x_v. We then have the appealing property that for any positive definite matrix Σ, $\Sigma \in \mathcal{C}^T$ if and only if $I^T(\Sigma) = 0$. Once the set $\mathcal{C}^T$ is well defined, we can easily solve TCA in the Gaussian case, as the following theorem makes precise:

Theorem 5 *If $x_1, \ldots, x_m$ are jointly Gaussian with covariance matrix Σ, the variable $s = Wx$ factorizes in the tree T if and only if there exists an orthogonal matrix R and $C \in \mathcal{C}^T$ such that $W = C^{1/2}R\Sigma^{-1/2}$.*

The study of the Gaussian case is useful for two reasons. First, we will use Equation (9) to define the KGV contrast function in Section 5.3 and the contrast function for time series in Section 8.

Second, the Gaussian solution can be exploited to yield a principled reduction of the search space for W. Recall that in ICA it is common to "whiten" (i.e. decorrelate and normalize) the data in a preprocessing step; once this is done the matrix W can be constrained to be an orthogonal matrix. In the TCA model, we cannot always require decorrelation of the components; indeed, two components linked by an edge might be heavily correlated. However, in some specific situations (Hyvärinen et al., 2001a), or when looking for clusters, it is appropriate to look for uncorrelated components and whitening can then be used to limit the search space for the matrix W.

If it is not possible to whiten the data, it seems reasonable for a given tree T to constrain W to be such that it is a solution to the Gaussian relaxation of the problem. This cannot be achieved entirely, because if a distribution factorizes in T, a Gaussian variable with the same first and second order moments does not necessarily factorize according to T (marginal independencies are preserved, but conditional independencies are not); nevertheless, such a reduction in complexity in the early stages of the search is very helpful to the scalability of our methods to large m, as discussed in Appendix A. We now turn to the definitions of our three empirical contrast functions for TCA.

5. Estimation of the Contrast Function

We have developed three empirical contrast functions. Each of them is derived from a corresponding ICA contrast function, extended to the TCA model.[4]

5.1 Estimating Entropies Using Kernel Density Estimation

Our first approach to approximating the objective function $J(x,W,T)$ is a direct approach, based on approximating the component marginal entropies $H(s_u)$ and joint entropies $H(s_u,s_v)$, $H(s)$ and $H(x)$ via kernel density estimation (KDE; Silverman, 1985, Pham, 1995). The first term in $J(x,W,T)$ can be written as $I(s_1,\ldots,s_m) = \sum_u H(s_u) - H(s)$, which can be expanded into $\sum_u H(s_u) - H(x) - \log|\det W|$. Since the joint entropy $H(x)$ is constant we do not need to compute it and thus to estimate $I(s_1,\ldots,s_m)$ we need only estimate one-dimensional entropies $H(s_i)$. We also require estimates of the pairwise mutual information terms in the definition of $J(x,W,T)$, which we obtain using two-dimensional entropy estimates. Thus, letting $\hat{H}_u$ and $\hat{H}_{uv}$ denote estimates of the singleton and pairwise entropies, respectively, we define the following empirical contrast function:

$$J^{KDE} = \sum_u \hat{H}_u - \sum_{(u,v)\in\mathcal{E}} (\hat{H}_u + \hat{H}_v - \hat{H}_{uv}) - \log|\det W|. \tag{10}$$

Given a set of N training samples $\{x_i\}$ in $\mathbb{R}^d$ and a kernel in $\mathbb{R}^d$, that is, a nonnegative function $K:\mathbb{R}^d \to \mathbb{R}$ that integrates to one, the kernel density estimate with bandwidth h is defined as $\hat{f}(x) = \frac{1}{Nh^d}\sum_{i=1}^N K\left(\frac{x-x_i}{h}\right)$ (Silverman, 1985). In this paper we use Gaussian kernels $K(x) = \frac{1}{(2\pi)^{d/2}} e^{-||x||^2/2}$ with $d = 1,2$.

The entropy of x is estimated by evaluating $\hat{f}(x)$ at points on a regular mesh that spans the support of the data, and then performing numerical integration. Binning, together with the fast Fourier transform, can be used to speed up the evaluations, resulting in a complexity which is linear in N and depends on the number M of grid points that are required. Although automatic methods

4. Other ICA empirical contrast functions that are based on likelihood or entropy (Pham, 2003, Hastie and Tibshirani, 2003) could be extended.

exist for selecting the bandwidth h (Silverman, 1985), in our experiments we used a constant $h = 0.25$. We also fixed $M = 64$.

Although each density estimate can be obtained reasonably cheaply, we have to perform $O(m^2)$ of these when minimizing over the tree T. This can become expensive for large m, although, as we show in Appendix A, this cost can be reduced by performing optimization on subtrees. In any case, our next two contrast function are aimed at handling problems with large m.

5.2 Gram-Charlier Expansions

A contrast function based on cumulants is easily derived from Equation (10), using Gram-Charlier expansions to estimate one-dimensional and two-dimensional entropies, as discussed by Amari et al. (1996) and Akaho et al. (1999). Since this contrast function only involves up to fourth order cumulants, it is numerically efficient and can be used to rapidly find an approximate solution which can serve as an initialization for the slower but more accurate contrast functions based on the other estimation methods (KGV or KDE).

More precisely, to compute the entropy of a centered univariate random variable a, we define $r = a/\sigma$ where $\sigma^2 = \mathrm{var}(a)$. We have the following approximation:

$$H(a) = H(r) + \log\sigma \approx \frac{1}{2}\log(2\pi e\sigma^2) - \frac{1}{48}(Er^4-3)^2 - \frac{1}{12}(Er^3)^2.$$

In order to compute the joint entropy of two centered univariate variables a and b, we let r and s denote the corresponding "whitened" variables, i.e. such that $(r,s)^\top = C^{-1/2}(a,b)^\top$, where C is the 2×2 covariance matrix of a and b. The entropy $H(a,b)$ can be computed from the entropy $H(r,s)$ as $H(a,b) = H(r,s) + \frac{1}{2}\log\det C$; for $H(r,s)$, we have the following approximation (Akaho et al., 1999):

$$\begin{aligned} H(r,s) \approx \log 2\pi e &- \frac{1}{12}\left[(Er^3)^2 + (Es^3)^2 + 3(Er^2s)^2 + 3(Ers^2)^2\right] \\ &- \frac{1}{48}\left[(Es^4-3)^2 + (Er^4-3)^2 + 6(Er^2s^2-1)^2 + 4(Er^3s)^2 + 4(Ers^3)^2\right]. \end{aligned}$$

Thus we can compute estimates $\hat{H}_{uv}$ and $\hat{H}_u$ as in previous section and define a contrast function J^{CUM}.

5.3 Kernel Generalized Variance

Our third contrast function is based on the *kernel generalized variance*, an approximation to mutual information introduced by Bach and Jordan (2002). We begin by reviewing the kernel generalized variance, emphasizing a simple intuitive interpretation. For precise definitions and properties, see Bach and Jordan (2002).

5.3.1 APPROXIMATION OF MUTUAL INFORMATION

Let $x = (x_1,\ldots,x_m)^\top$ be a random vector with covariance matrix Σ. The m-fold mutual information for an associated Gaussian random vector with same mean and covariance matrix as x is equal to $I^G(\Sigma) = -\frac{1}{2}\log\left(\frac{\det\Sigma}{\Sigma_{11}\cdots\Sigma_{mm}}\right)$. The ratio $\frac{\det\Sigma}{\Sigma_{11}\cdots\Sigma_{mm}}$ is usually referred to as the *generalized variance*.

If x is not Gaussian, I^G is not in general a good approximation to the mutual information between $x_1,\ldots,x_m$. But if we first map each component x_i from $\mathbb{R}$ to a higher-dimensional space $\mathcal{F}$, and then treat the mapped variables as Gaussian in this space, it turns out that we obtain a useful approximation of the mutual information. More precisely, we map each component x_i to $\Phi(x_i) \in \mathcal{F}$ via a map Φ, and define the covariance matrix $\mathcal{K}$ of $\Phi(x) = (\Phi(x_1), \ldots, \Phi(x_m))^T \in \mathcal{F}^m$ by blocks: $\mathcal{K}_{ij}$ is the covariance between $\Phi(x_i)$ and $\Phi(x_j)$. The size of each of these blocks is the dimension of $\mathcal{F}$. For simplicity we can think of $\mathcal{F}$ as a finite-dimensional space, but the definition can be generalized to any infinite-dimensional Hilbert space. Let $\Phi^G(x) \in \mathcal{F}^m$ be a Gaussian random vector with the same mean and covariance as $\Phi(x)$. The mutual information $I^K(\mathcal{K})$ between $\Phi_1^G(x), \ldots, \Phi_m^G(x)$ is equal to

$$I^K(\mathcal{K}) = -\frac{1}{2} \log \frac{\det \mathcal{K}}{\det \mathcal{K}_{11} \cdots \det \mathcal{K}_{mm}}, \tag{11}$$

where the ratio $\frac{\det \mathcal{K}}{\det \mathcal{K}_{11} \cdots \det \mathcal{K}_{mm}}$ is called the *kernel generalized variance (KGV)*.

We now list the main properties of I^K, which we refer to as the *KGV-mutual information*.

- **Mercer kernels and Gram matrices**. A *Mercer kernel* on $\mathbb{R}$ is a function $k(x,y)$ from $\mathbb{R}^2$ to $\mathbb{R}$ such that for any set of points $\{x^1, \ldots, x^N\}$ in $\mathbb{R}$, the $N \times N$ matrix K, defined by $K_{ij} = k(x_i, x_j)$, is positive semidefinite. The matrix K is usually referred to as the *Gram matrix* of the points $\{x^i\}$. Given a Mercer kernel $k(x,y)$, it possible to find a space $\mathcal{F}$ and a map Φ from $\mathbb{R}$ to $\mathcal{F}$, such that $k(x,y)$ is the dot product in $\mathcal{F}$ between $\Phi(x)$ and $\Phi(y)$ (see, e.g., Schölkopf and Smola, 2001). The space $\mathcal{F}$ is usually referred to as the *feature space* and the map Φ as the *feature map*. This allows us, given sample data, to define an estimator of the KGV via the Gram matrices K_i of each component x_i. Indeed, using the "kernel trick," we can find a basis of $\mathcal{F}$ where $\mathcal{K}_{ij} = K_i K_j$. Thus, for the remainder of the paper, we assume that Φ and $\mathcal{F}$ are associated with a Mercer kernel $k(x,y)$.

- **Linear time computation**. If $k(x,y)$ is the Gaussian kernel $k(x,y) = \exp(-(x-y)^2/2\sigma^2)$, then the estimator based on Gram matrices can be computed in linear time in the number N of samples. In this situation, $\mathcal{F}$ is an infinite-dimensional space of smooth functions. This low complexity is obtained through low-rank approximation of the Gram matrices using incomplete Cholesky decomposition (Bach and Jordan, 2002). We need to perform m such decompositions, where each decomposition is $O(N)$. The worst-case running time complexity is $O(mN + m^3)$, but under a wide range of situations, the Cholesky decompositions are the practical bottlenecks of the evaluation of I^K, so that the empirical complexity is $O(mN)$.

- **Relation to actual mutual information**. For $m = 2$, when the kernel width σ tends to zero, the KGV mutual information tends to a quantity that is an expansion of the actual mutual information around independence (Bach and Jordan, 2002). In addition, for any m, I^K is a valid contrast function for ICA, in the sense that it is equal to zero if and only if the variables $x_1, \ldots, x_m$ are pairwise independent.

- **Regularization**. For numerical and statistical reasons, the KGV has to be regularized, which amounts to convolving the Gaussian variable $\Phi^G(x)$ by another Gaussian having a covariance matrix proportional to the identity matrix κI. This implies that in the approximation of the KGV, we have $\mathcal{K}_{ij} = K_i K_j$ for $i \neq j$, and $\mathcal{K}_{ij} = (K_i + N\kappa I)^2$ for $i = j$, where κ is a constant regularization parameter.

5.3.2 A KGV CONTRAST FUNCTION FOR TCA

Mimicking the definition of the T-mutual information in Equation (6) and the Gaussian version in Equation (9), we define the KGV contrast function $J^K(x,T)$ for TCA as $J^K(x,T) = I^K(x) - \sum_{(u,v)\in\mathcal{E}} I^K_{uv}(x)$, that is:

$$J^K(x,T) = -\frac{1}{2}\log\frac{\det\mathcal{K}}{\det\mathcal{K}_{11}\cdots\det\mathcal{K}_{mm}} + \frac{1}{2}\sum_{(u,v)\in\mathcal{E}}\log\frac{\det\mathcal{K}_{uv,uv}}{\det\mathcal{K}_{uu}\det\mathcal{K}_{vv}}.$$

An important feature of J^K is that it is the T-mutual information of $\Phi_1^G(x),\ldots,\Phi_m^G(x)$, which are linked to $x_1,\ldots,x_m$ by the feature maps and the "Gaussianization." It is thus always nonnegative. Note that going from a random vector y to its associated Gaussian y^G is a mapping from distribution to distribution, and not a mapping from each realization of y to a realization of y^G. Unfortunately, this mapping preserves marginal independencies but not conditional independencies, as pointed out in Section 4.2. Consequently, $J^K(x,T)$ does not characterize factorization according to T; that is, $J^K(x,T)$ might be strictly positive even when x does factorize according to T. Nonetheless, based on our earlier experience with KGV in the case of ICA (Bach and Jordan, 2002), we expect $J^K(x,T)$ to provide a reasonable approximation to $I^T(x)$. Intuitively, we fit the best tree for the Gaussians in the feature space and hope that it will also be a good tree in the input space. Recent work by Fukumizu et al. (2003) has shown how to use KGV to exactly characterize conditional independence.

Numerically, $J^K(x,T)$ behaves particularly nicely, since all of the quantities needed are Gram matrices, and are obtained from the m incomplete Cholesky decompositions. Thus we avoid the $O(m^2)$ complexity. In our empirical experiments, we used the settings $\sigma = 1$ and $\kappa = 0.01$ for the free parameters in the KGV. The contrast function that we minimize with respect to T and W is then $J^{KGV}(x,W,T) = J^K(Wx,T)$.

6. The TCA Algorithm

We now give a full description of the TCA algorithm. Any of the three contrast functions that we have defined can be used in the algorithm. We generically denote the contrast function as $J(x,W,T)$ in the following description of the algorithm.

6.1 Formulation of the Optimization Problem

First, as noted in Section 4.1, we minimize $J(x,W,T)$ on the space of matrices such that Wx has unit variance components. That is, if Σ denotes the covariance matrix of x, we constrain the rows of $W\Sigma^{1/2}$ to have unit norm. Therefore, the search space $\mathcal{M}$ is isomorphic to a product of m spheres in m dimensions. In order to take into account the "leaf mixing" invariance (see Section 4.1), we also add a penalty term, $J^C(x,W,T) = -\frac{1}{2}\sum_{(u,v)\in\mathcal{E}}\log\left(1-\operatorname{corr}^2((Wx)_u,(Wx)_v)\right)$ that penalizes marginal correlation along edges of the tree T. Finally, a prior term $w(T)$ may be added, as defined in Section 3.3.

We thus aim to solve the following optimization problem over W and T:

$$\begin{array}{ll}\text{minimize} & F(W,T) = J(x,W,T) + \lambda_C J^C(x,W,T) - w(T)\\ \text{subject to} & (W\Sigma W^\top)_{ii} = 1,\ \forall i\in\{1,\ldots,m\},\end{array}$$

where λ_C determines how much we penalize the marginal correlations. In all of our experiments we used $\lambda_C = 0.05$. When the whitening constraints can be imposed (see Section 4.2), then the formulation simplifies as follows, with a search space $\mathcal{M}$ isomorphic to the group of orthogonal matrices:

$$\begin{array}{ll} \text{minimize} & F_w(W,T) = J(x,W,T) - w(T) \\ \text{subject to} & W\Sigma W^\top = I, \end{array}$$

6.2 Minimization Algorithm

The optimization problem that we need to solve involves one continuous variable W and one discrete variable T. Minimization with respect to any of the two variables while the other is fixed can be done efficiently: minimizing with respect to T is equivalent to a maximum weight forest problem, which can easily be solved by the greedy algorithm presented in Section 3.3, while minimizing with respect to W can be done by gradient descent—for the three empirical contrast functions defined in Section 5, there are efficient techniques to compute the gradient (Silverman, 1985, Akaho et al., 1999, Bach and Jordan, 2002). However, since one of the two variables is discrete, an alternating minimization procedure might not exhibit good convergence properties. In order to use continuous optimization techniques, we consider the function $G(W)$ obtained by minimizing with respect to the tree T; that is, $G(W) = \min_T F(W,T)$. This function is continuous and piecewise differentiable. The problem is now that of minimizing this function with respect to W on a manifold $\mathcal{M}$, which in our situation, is a product of spheres in the general case or the orthogonal group when the whitening constraint is imposed. We can use the structure of such manifolds to design efficient coordinate descent algorithms; indeed it is possible to minimize functions defined on those manifolds with an iterative procedure: at each iteration, a pair of indices (i,j) is chosen, then all the rows of W are held fixed except the i-th and j-th row which are allowed to vary inside the subspace spanned by the i-th and j-th rows obtained from the previous iteration. For both types of manifold, all elements can be generated by a finite sequence of such local moves.

In the case of the orthogonal group, this is the classical technique of Jacobi rotations (Cardoso, 1999) where for each pair (i,j) we have to solve a one-dimensional problem. In the case of the product of spheres, this is a two-dimensional problem. For both problems, the evaluation of the contrast functions when all rows but two are held fixed has linear complexity in m, thus making the local searches efficient. An outline of the algorithm is presented in Figure 4. At the obtained stationary point W, if there is only one tree that attains the minimum $\min_T F(W,T)$, then this is a local minimum of $G(W)$. We present in Appendix A a set of techniques to make the algorithm scalable to large numbers of variables.

7. TCA as a Density Estimation Model

TCA can be used as a density estimation model. Indeed, once the optimal (or an approximation thereof) W and T are found, we just need to perform density estimation in a graphical model, independently of the method that was used to find W and T. The model with respect to which we carry out this density estimation is a tree, and thus we can work either within the directed graphical model framework or the undirected graphical model framework. We prefer the former because the lack of a normalizing constant implies that the density estimation problem decouples.

We thus have to estimate a density of the form $p(s) = \prod_{f \in \mathcal{F}} p(s_f) \prod_{u \in \mathcal{V} \setminus \mathcal{F}} p(s_u | s_{\pi_u})$, where $\mathcal{F}$ is a set of founders for the directed tree T and π_u is the parent of node u in the directed tree T. The overall estimation problem reduces to finding m distinct density estimates: one-dimensional estimates

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

Algorithm:

1. Initialization: W random
2. While $G(W) = \min_T F(W,T)$ is decreasing

   for $i = 2$ to m, for $j = 1$ to $i-1$, $W \leftarrow \arg \min_{V \in L_{ij}(W)} \left\{ \min_T F(V,T) \right\}$

   where $L_{ij}(W)$ is the set of matrices $V \in \mathcal{M}$ such that

   (a) $\forall k \notin \{i,j\}, V_k = W_k$

   (b) $\operatorname{span}(V_i, V_j) = \operatorname{span}(W_i, W_j)$
3. Compute $T = \arg\min F(W,T)$

Output: demixing matrix W, tree T

Figure 4: The TCA algorithm: $F(W,T)$ is the contrast function and $\mathcal{M}$ is the search space for W, as defined in Section 6.1 (we use the notation A_k for the k-th row of a matrix A).

for nodes with no parents (the founders), and conditional density estimates for the remaining nodes with one parent. In this paper we use a Gaussian mixture model for the densities at the founders, and conditional Gaussian mixture models, also known as "mixtures of experts models" (Jacobs et al., 1991), for the remaining conditional probabilities. All of these mixture models can be estimated via the expectation-maximization (EM) algorithm. In order to determine the number of mixing components for each model, we use the minimum description length criterion (Rissanen, 1978).

Our two-stage approach is to be contrasted with an alternative one-stage approach in which one would define a model using W, T and Gaussian-mixture conditional distributions, and perform maximum likelihood using EM—such an approach could be viewed as an extension of the independent factor analysis model (Attias, 1999) to the tree setting. By separating density estimation from the search for W and T, however, we are able to exploit the reduction of our parameter estimation problem to bivariate density estimation. Bivariate density estimation is a well-studied problem, and it is possible to exploit any of a number of parametric or nonparametric techniques. These techniques are computationally efficient, and good methods are available for controlling smoothness. We also are able to exploit the KGV technique within the two-phase approach, an alternative that does not rely explicitly on density estimation.

Finally, and perhaps most significantly, our approach does not lead to intractable inference problems that require sampling or variational methods, as would be necessary within a tree-based generalization of the independent factor analysis approach.[5]

8. Stationary Gaussian Processes

Pham (2002) has shown that for ICA, the semiparametric approach can be extended to stationary Gaussian processes. In this section, we extend his results to the TCA model. We assume first that the sources are doubly-infinite sequences of real valued observations $\{s_k(t), t \in \mathbb{Z}\}$, $k = 1, \ldots, m$.

5. The orthogonalization approach which relies on whitening and which leads to a significant speed up of learning (Welling and Weber, 2001) cannot be used here, as pointed out in Section 4.2.

We model this multivariate sequence as a zero-mean multivariate Gaussian stationary process (we assume that the mean is zero or 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:

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

For each ω, $f(\omega)$ is an $m \times m$ Hermitian positive semidefinite matrix. In addition the function $\omega \mapsto f(\omega)$ is 2π-periodic. In the following we always assume that for every ω, $f(\omega)$ is invertible (i.e., positive definite).

8.1 Entropy Rate of Gaussian Processes

The entropy rate of a process s is defined as (Cover and Thomas, 1991)

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

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. Hannan, 1970):

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

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 processes 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:

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

8.2 Graphical Model for Time Series

The graphical model framework can be extended to multivariate time series (Brillinger, 1996, Dahlhaus, 2000). The dependencies that are considered are between whole time series; that is, between the entire sets $\{s_i(t), t \in \mathbb{Z}\}$, for $i = 1, \ldots m$. If the process is jointly Gaussian and stationary, then most of the graphical model results for Gaussian variables can be extended. In particular, s_a is conditionally independent from s_b given all others $s_u, u \neq a, b$, if and only if $\forall \omega, (f(\omega)^{-1})_{ab} = 0$. Also, maximum likelihood estimation of spectral density matrices in decomposable models decouples and is equivalent to equating local spectral density matrix functions. As we show in the next section, this enables Theorem 1 and Theorem 3 to be extended to the time series case.

8.3 Contrast Function

Let x be a multivariate time series $\{x_k(t), t \in \mathbb{Z}\}$, $k = 1, \ldots, m$. 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 3 to time series (W_u denotes the u-th row of W):

Theorem 6 *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:*

$$J^{STAT}(f, T, W) = I^T(f, W) = I(f, W) - \sum_{(u,v) \in \mathcal{E}} I_{uv}(f, W), \tag{12}$$

where

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

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

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

is the pairwise mutual information between s_u and s_v.

Thus, the goal of TCA is to minimize $J^{STAT}(f, T, W) = I^T(f, W)$ in Equation (12) with respect to W and T; in our simulations, we refer to this contrast function as the STAT contrast function.

8.4 Estimation of the Spectral Density Matrix

In the presence of a finite sample $\{x(t), t = 0, \ldots, N-1\}$, we use the *smoothed periodogram* (Brockwell and Davis, 1991) 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[6]

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

and can readily be computed using m fast Fourier transforms (FFT). We use the following estimated spectral density matrices:

$$\hat{f}(\omega_k) = \frac{1}{N} \sum_{j=0}^{N-1} W(j) I_N(\omega_{j+k}), \tag{13}$$

where $W(j)$ is a smoothing window that is required to be symmetric and sum to one. In our simulations, we used a Gaussian window $W(j) \propto e^{-j^2/p^2}$. If the number of samples N tends to infinity with $p(N)$ tending to infinity such that $p(N)/N \to 0$, then Equation (13) provides a consistent estimate of the spectral density matrix, and there are methods to select an optimal smoothing parameter p automatically (Ombao et al., 2001).

6. We assume the means have been previously removed from the data.

Finally, our contrast function involves integrals of the form

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

Following Pham (2001a), we estimate them by Riemannian sums using estimated values of f at $\omega_j = 2\pi j/N$, $\omega_j \in [-\pi,\pi]$, from Equation (13). Because of the smoothness of the spectral density, the Riemannian sums can be subsampled; in simulations we use 64 samples.

9. Simulation Results

We have conducted an extensive set of experiments using synthetic data.[7] In a first set of experiments, we focus on the performance of the first stage of the algorithm, i.e., the optimization with respect to W and T, when the data actually follow the TCA model. In a second set of experiments, we focus on the density estimation performance, in situations where the TCA model assumptions actually hold, and in situations where they do not. Finally, in a third set of experiments, we focus on the task of recovering clusters in ICA, both in the non-Gaussian temporally independent case and the stationary Gaussian case.

9.1 Recovering the Tree and the Linear Transform

In this set of experiments, for various numbers of variables m, we generated data from distributions with known density: we selected a spanning tree T at random, and conditional distributions were selected among a given set of mixtures of experts. Then 1000 samples were generated and rotated using a known random square matrix A, which corresponds to a demixing matrix $W = A^{-1}$.

To evaluate the results $\hat{W}$ and $\hat{T}$ of the TCA algorithm—with the three contrast function J^{CUM}, J^{KGV} and J^{KDE}, based on cumulants, the kernel generalized variance, and kernel density estimation, respectively—we need to use error measures that are invariant with respect to the known invariances of the model, as discussed in Section 4.1.

For the demixing matrix W, we use a measure commonly used for ICA (Amari et al., 1996), that is invariant by permutation and scaling of rows: we form $B = \hat{W}W^{-1}$ and compute $d = \frac{100}{m(m-1)}\sum_{i=1}^{m}\left\{\frac{\sum_{j=1}^{m}|B_{ij}|}{\max_j |B_{ij}|} - 1\right\}$. This measure is always between 0 and 100 and equal to zero if and only if there is a perfect match between W and $\hat{W}$. Intuitively, it measures the average proportion of components that are not recovered. However, because of the "leaf mixing" invariance, before computing d, we transform W and $\hat{W}$ to equivalent demixing matrices which respect the normalization we choose—marginal decorrelation between the leaf node and its parent. We let e_W denote the final error measure.

In order to compare the trees T and $\hat{T}$, we note that the computation of e_W also outputs, for each estimated component, the true component that is closest. Using these assignments, we compute the number of edges in T that are not recovered in $\hat{T}$, and define the error e_T as this number multiplied by $\frac{100}{m-1}$ (if two components are assigned to the same true component, which can only occur when the demixing is especially bad, then some edges of $\hat{T}$ become trivial when estimated components are assigned to true component, simply making e_T larger).

We report results (averaged over 20 replications) in Table 1. We also ran two ICA algorithms, JADE (Cardoso, 1999) and Fast-ICA (Hyvärinen et al., 2001b). Our algorithms manage to recover

7. A MATLAB implementation can be downloaded at `http://www.cs.berkeley.edu/~fbach/`.

	e_W					e_T		
	ICA		TCA			TCA		
m	F-ICA	JADE	CUM	KGV	KDE	CUM	KGV	KDE
4	34.1	20.9	7.3	2.9	1.8	0	0	0
6	33.2	21.6	9.3	3.9	2.0	8.5	1.5	1.5
8	30.5	17.4	10.8	6.2	2.3	14.5	8.2	2.9
12	25.2	16.4	10.7	6.6	2.3	31.8	9.5	5.5
16	24.1	15.8	12.0	7.0	1.5	34.1	12.4	2.5

Table 1: Recovering the tree T and the matrix W, for increasing numbers m of components. See text for details on the definitions of the performance measures e_W and e_T.

m	τ	GAU	IND	CL	ICA	GMM	T-CUM	T-KGV	T-KDE
4	**1**	**1.4**	**1.4**	**1.0**	**1.1**	**0.5**	**0.4**	**0.3**	**0.3**
4	2	1.3	1.6	0.8	1.1	0.3	0.6	0.4	0.5
6	**1**	**2.3**	**2.7**	**2.2**	**2.0**	**1.2**	**0.9**	**0.5**	**0.4**
6	2	2.2	2.9	1.9	1.8	0.9	1.1	1.0	0.7
6	3	1.9	2.9	1.8	1.7	0.8	1.1	1.1	0.8
8	**1**	**3.4**	**3.9**	**3.4**	**2.9**	**2.3**	**2.1**	**1.0**	**0.5**
8	2	3.1	4.3	3.2	2.7	1.8	1.6	1.7	1.0
8	3	2.9	4.2	3.0	2.5	1.6	1.9	1.9	1.2
8	4	3.0	4.2	3.1	2.6	1.6	2.1	2.1	1.3
12	**1**	**5.1**	**5.9**	**5.4**	**4.5**	**4.4**	**3.0**	**1.5**	**0.7**
12	2	5.1	7.2	5.9	4.5	3.8	4.4	3.4	1.7
12	3	4.8	7.2	5.7	4.3	3.5	3.9	3.7	1.9
12	4	4.6	6.9	5.4	4.1	3.2	3.3	3.5	2.1

Table 2: Density estimation for increasing number of components m and treewidth τ of the generating model (all results are averaged over 20 replications).

W and T very accurately, with the contrast based on kernel density estimation leading to the best performance. Moreover, using an ICA algorithm for this problem leads to significantly worse performance. An interesting fact that is not apparent in the table is that our results are quite insensitive to the "density" of the tree that was used to generate the data: bushy trees yield roughly the same performance as sparse trees.

9.2 Density Estimation

Here we focus on density estimation, comparing the following models: Gaussian (GAU), Gaussian mixture (GMM), independent Gaussian mixtures (IND), Chow-Liu with Gaussian mixtures (CL), ICA using marginal Gaussian mixtures (ICA), and TCA using Gaussian mixtures (T-CUM, T-KDE, T-KGV).

We generated data as follows: we designed a set of graphical models with given treewidth[8] τ between 1 (trees) and 4 (maximal cliques of size 5). Then data were generated using one of these models and rotated using a random matrix A. We report results (averaged over 20 replications) in Table 2, where performance is measured as the average log likelihood of a held-out test set, minus the same average under the (known) generating model.

When the treewidth τ is equal to 1 (lines in bold in Table 2), the data exactly follow the TCA model and it is no surprise that our TCA algorithm outperforms the other models. However, when τ is greater than one, the TCA model assumptions do not hold, but our models still exhibit good performance, especially with the contrast function based on kernel density estimation (KDE). Note that when the generating model becomes too connected (e.g., $m = 8, \tau = 4$), the performance of the TCA models starts to degrade, which simply illustrates the fact that in those conditions the tree approximation is too loose.

9.3 Finding Clusters in ICA

In this section, we study the problem of finding clusters in independent component analysis. In all of our simulations, data were generated from q independent clusters $C_1, \ldots, C_q$, and were then rotated by a random but known matrix A. We measure the demixing performance by comparing the results of our algorithms $\hat{W}$ to $W = A^{-1}$. We also compare the true and estimated clusters.

9.3.1 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, such as the one presented in the previous section. Indeed, what needs to be measured is how much $B = \hat{W}W^{-1}$ differs from a diagonal matrix, up to permutation. In our case, however, we need to measure how much B differs from a block diagonal matrix, up to permutation (Cardoso, 1998).

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

$$Q_{ji} = 1 - \left(\sum_{p \in C_k} |B_{pi}| \right) \Big/ \left(\sum_{p=1}^{m} |B_{pi}| \right),$$

which is the cost of assigning the estimated component i to the cluster C_k. For each permutation σ over m elements, we define the cost of the assignment of estimated components to clusters defined by σ to be $e(\sigma) = \frac{100}{m} \sum_i Q_{\sigma(i)i}$. Finally, the performance metric is defined as the minimum of $e(\sigma)$ over all permutations:

$$e_W = \max_\sigma e(\sigma) = \frac{100}{m} \max_\sigma \sum_i Q_{\sigma(i)i},$$

which can be computed in polynomial time by the Hungarian method (Bertsimas and Tsitsiklis, 1997). The metric e_W is always between 0 and 100 and is equal to zero if and only if $\hat{W}$ is equivalent to W. Roughly, as with the classical ICA metric, e_W measures the average proportion of components that are missing. As in Section 9.1, the performance metric W also outputs assignments of estimated components to true components (in this case, the assignments are one-to-one); we can now compare

8. The treewidth of a graph G characterizes the complexity of exact inference for distributions that factorize according to G. In particular it is equal to one if and only if the graph is a tree (Lauritzen, 1996).

		e_W					e_C				
		ICA		TCA			ICA		TCA		
m	comp	JADE	F-ICA	CUM	KGV	KDE	JADE	F-ICA	CUM	KGV	KDE
4	22	4.3	8.0	3.9	2.9	3.5	0	7.1	0	0	0
6	33	12.9	18.5	9.3	7.9	15.4	5.0	7.6	4.5	4.0	2.7
6	321	10.6	15.7	8.1	5.9	7.3	2.8	6.3	2.3	3.3	0.5
6	222	7.4	14.2	6.7	5.1	6.1	0	6.0	0	0	0
8	332	17.2	24.8	13.0	10.3	16.9	4.2	6.4	2.3	2.3	1.2
8	3221	14.5	20.3	11.0	9.3	11.0	2.7	4.9	1.6	1.5	1.1
8	2222	11.7	20.6	9.6	7.9	9.0	1.3	5.7	0	0	0.2
12	43221	30.9	37.0	23.6	16.9	23.2	5.1	6.4	1.6	1.6	2.1
12	3333	39.6	41.5	25.5	22.8	29.0	5.8	6.3	1.3	1.1	3.2
12	222222	22.9	31.7	16.1	14.0	12.7	2.4	5.5	0.1	0	0.2

Table 3: Finding clusters: results for temporally independent sources. The sizes of each cluster is indicated in the column "comp." See text for details on the definitions of the performance measures e_W and e_C.

the two clusterings by computing the percentage e_C of disagreements between the two clusterings of components.

9.3.2 COMPARISONS

For temporally independent sources, we compare our algorithm—with the three different contrast functions J^{CUM}, J^{KGV} and J^{KDE}—to two ICA algorithms, JADE (Cardoso, 1999) and Fast-ICA (Hyvärinen et al., 2001b). For Gaussian stationary sources, we compare our algorithm to three ICA algorithms for time series, SOBI (Belouchrani et al., 1997), TDSEP (Ziehe and Müller, 1998) and an algorithm that minimizes with respect to W the contrast function used by Pham (2001a). This contrast function corresponds exactly to our contrast function J^{STAT} when no edges are allowed in the graph; that is, $J^{STAT}(f,W,\varnothing)$.

In order to cluster components after unmixing using the ICA algorithm results, we use a mutual-information-based criterion, based on kernel density estimation in the *iid* case, and on the spectral density in the Gaussian stationary case (with the same threshold as that used to build the prior distribution over trees for TCA).

9.3.3 TEMPORALLY INDEPENDENT SOURCES

We used different patterns of components (different numbers and sizes of clusters). For each component we generated N *iid* samples from a mixture of three Gaussians with random means and covariance matrices. Then the data were rotated by a known 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 3. The TCA methods recover the components more consistently than the "plain" ICA algorithms.

		e_W				e_C			
		ICA			TCA	ICA			TCA
m	comp.	Pham	SOBI	TDSEP	T-STAT	Pham	SOBI	TDSEP	T-STAT
4	22	6.7	12.0	20.7	4.7	1.7	5.7	6.7	0.8
6	33	8.9	17.0	26.5	5.4	1.2	5.5	7.0	0
6	321	10.7	16.7	34.0	7.9	0.9	4.7	4.1	0.5
6	222	12.3	18.2	37.7	8.9	1.6	4.2	5.2	1.0
8	332	15.8	23.1	37.3	9.6	2.6	4.3	5.6	0.4
8	3221	15.0	21.6	40.4	11.2	1.1	3.7	5.0	0.6
8	2222	17.9	22.8	45.6	11.8	1.3	3.0	4.1	0.8
12	43221	23.8	33.0	51.6	16.2	1.5	3.6	5.1	0.5
12	3333	23.4	32.2	50.0	15.6	1.9	4.3	5.0	0.3
12	222222	26.1	33.9	56.3	18.8	1.1	2.8	4.3	0.4
16	43333	30.2	40.0	55.8	20.4	1.4	3.5	4.4	0.3
16	22222222	34.8	45.1	63.4	24.6	0.9	2.4	3.6	0.3

Table 4: Finding clusters: results for Gaussian stationary sources. The sizes of each cluster is indicated in the column "comp." See text for details on the definitions of the performance measures e_W and e_C.

9.3.4 STATIONARY GAUSSIAN PROCESSES

Given the numbers and sizes of each component, the data for each component were generated from random spectral densities that cannot be modelled by an ICA model. We performed simulations with various numbers of sources, from $m = 4$ to $m = 12$. We report results obtained from 20 replications in Table 4, where, as in the temporally independent case, our algorithm outperforms the extant ICA algorithms.

10. Conclusion

We have presented a model that extends the classical ICA model, by allowing tree-structured dependence among the components. The tree T and the demixing matrix W are determined by minimizing contrast functions within a semiparametric estimation framework. Once W and T are found, the remaining densities are readily estimated.

There are a number of further potential generalizations of the methods discussed in this paper. In general, we believe that TCA and ICA provide appealing examples of the utility of applying an adaptive transformation to data before fitting a model, and thereby extending the range of problems to which graphical models can be usefully applied. Moreover, kernel generalized variances provide a fast and flexible way of computing model scores, not only for continuous variables but potentially also for discrete variables and discrete structures, such as strings and trees (Lodhi et al., 2001). Finally, although we have limited ourselves to a generalization of ICA that allows tree-structured dependency among the latent variables, it is clearly of interest to make use of the general graphical model toolbox and consider broader classes of dependency. An interesting direction to consider in this regard is the general class of decomposable models.

Acknowledgments

We would like to acknowledge support for this project from the National Science Foundation (NSF grant IIS-9988642), and the Multidisciplinary Research Program of the Department of Defense (MURI N00014-00-1-0637). The MATLAB codes for the ICA algorithms that we used for comparison were downloaded from the homepages of their respective authors, except for SOBI, for which we used the implementation in ICALAB (Cichocki et al., 2002).

Appendix A. Scaling Issues

In the experiments reported in this paper, we have limited ourselves to problems in which the number of components m is less than 16. While our algorithms can be directly applied to large m, for problems in which m is significantly larger, additional numerical techniques are needed, for two reasons. The first one is running time complexity. Indeed, the contrast function based on KGV scales as $O(mN)$ but the one based on KDE scales as $O(m^2N)$. Second, with increasing number of sources, both ICA and TCA contrast functions tend to have multiple local minima. We now present five optimization techniques aimed at dealing with large-scale problems.

A.1 Initialization Using ICA

We can obtain a good initialization for TCA using the result of any ICA algorithm. Intuitively, this is helpful in our setting, because ICA is known to find components that are as "non-Gaussian" as possible (Hyvärinen et al., 2001b), and therefore the components that ICA finds should be linear combinations of only a few of the original non-Gaussian components (combinations of large numbers of components are subject to the central limit theorem and should approach Gaussianity). Thus by initializing with an ICA solution, the search for a TCA solution can effectively be limited to a subspace of lower dimension.

A.2 Exhaustive Line Searches

In order to avoid local minima, we first start by performing all local searches (in one or two dimensions depending on the manifold) using exhaustive search on a mesh grid, then we use local search algorithms until convergence. This technique has shown to be quite efficient at escaping local minima in the context of ICA (see e.g. Miller and Fisher III, 2003).

A.3 Growing Trees

An efficient heuristic to avoid local minima is to start from ICA and add edges to the tree T sequentially from zero to $m-1$. When no prior on the tree is used, we find that the best tree is necessary spanning (i.e., with $m-1$ edges), and thus the last tree is kept. In the presence of a prior that penalizes edges, at the end of the algorithm, $m-1$ results have to be compared and the best one is kept.

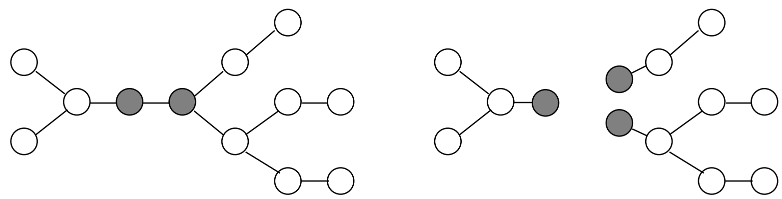

Figure 5: (Left) original tree T and subtree V (shaded). (Right) subtrees U_i (non-shaded) with their neighbor in T (shaded).

A.4 Optimizing Subtrees

Gradient descent can be performed sequentially on limited subspaces of the space of the matrix W. Indeed, given current estimates for T and W, we can perform optimization over a subset of the rows of W whose indices span a connected subtree U of T. In this case, the overall contrast function can be approximated by the contrast function for the subtree that contains U and its neighbors in T. Note that in the case of TCA, it is not possible a priori to use "one-unit contrast functions" that enable to find one component at a time.

To select the subtrees U, we use the following procedure: we generate all the subtrees V of small sizes (less than 4) and we measure how well V "separates the graph"; that is, if we let $U_1,\ldots,U_p$ denote the connected components of the complement of V in T (see Figure 5), and if $s = Wx$ are the estimated sources, we measure the conditional independence of $s_{U_1},\ldots,s_{U_p}$ given s_V. The KGV provides an approximate measure (Bach and Jordan, 2003), by simply computing $J(V) = I(s_V, s_{U_1},\ldots,s_{U_p}) - \sum_{i=1}^{p} I(s_V, s_{U_i})$, where the mutual informations are estimated using the Gaussian variables in feature space: once the Cholesky decompositions of each component s_i are performed and cached, computing all these scores only involves determinants of small matrices, and thus many subtrees V can be scored. The subtrees V with small score $J(V)$ do not need to be improved, and the subtrees that are selected for further optimization are the connected components $U_1,\ldots,U_p$ corresponding to those subtrees V.

A.5 Covariance Constraint

When the whitening constraint cannot be imposed, we can constrain the W matrix to yield a solution in the Gaussian case, as detailed in Section 4.2. We can optimize over matrices that belong to $\mathcal{C}^T$ and thus reduce the dimension of the search space from $m(m-1)$ to $m(m-1)/2 + (m-1)$.

References

S. Akaho, Y. Kiuchi, and S. Umeyama. MICA: Multimodal independent component analysis. In *Proceedings of the International Joint Conference on Neural Networks*, 1999.

S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In *Advances in Neural Information Processing Systems 8*, 1996.

H. Attias. Independent factor analysis. *Neural Computation*, 11(4):803–851, 1999.

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

F. R. Bach and M. I. Jordan. Learning graphical models with Mercer kernels. In *Advances in Neural Information Processing Systems 15*, 2003.

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

A. Belouchrani, K. Abed Meraim, J.-F. Cardoso, and É. Moulines. A blind source separation technique based on second order statistics. *IEEE Transactions on Signal Processing*, 45(2):434–444, 1997.

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

P. J. Bickel, C. A. J. Klaassen, Y. Ritov, and J. A. Wellner. *Efficient and Adaptive Estimation for Semiparametric Models*. Springer Verlag, 1998.

D. Brillinger. Remarks concerning graphical models for time series and point processes. *Revista de Econometria*, 16:1–23, 1996.

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

J.-F. Cardoso. Multidimensional independent component analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1998.

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

C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14:462–467, 1968.

A. Cichocki, S. Amari, and K. Siwek. ICALAB toolboxes, 2002. `http://www.bsp.brain.riken.go.jp/ICALAB`.

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

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

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

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

A. P. Dawid and S. L. Lauritzen. Hyper Markov laws in the statistical analysis of decomposable graphical models. *Annals of Statistics*, 21(3):1272–1317, 1993.

N. Friedman and M. Goldszmidt. Learning Bayesian networks with local structure. In M. I. Jordan, editor, *Learning in Graphical Models*. MIT Press, 1998.

K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. Technical Report 641, Department of Statistics, University of California, Berkeley, 2003.

E. J. Hannan. *Multiple Time Series*. John Wiley & Sons, 1970.

T. Hastie and R. Tibshirani. Independent component analysis through product density estimation. In *Advances in Neural Information Processing Systems 15*, 2003.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

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.

A. Hyvärinen, P.O. Hoyer, and M. Inki. Topographic independent component analysis. *Neural Computation*, 13(7):1525–1558, 2001a.

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

R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

R. Jirousek. Solution of the marginal problem and decomposable distributions. *Kybernetika*, 27(5): 403–412, 1991.

M. I. Jordan. Graphical models. *Statistical Science (Special Issue on Bayesian Statistics)*, 2002. In press.

S. L. Lauritzen. *Graphical Models*. Clarendon Press, 1996.

H. Lodhi, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. In *Advances in Neural Information Processing Systems 13*, 2001.

M. R. Luettgen, W. C. Karl, and A. S. Willsky. Efficient multiscale regularization with applications to the computation of optical flow. *IEEE Transactions on Image Processing*, 3(1):41–64, 1994.

E. G. Miller and J. W. Fisher III. ICA using spacings estimates of entropy. In *Proceedings of the Fourth Symposium on Independent Component Analysis and Blind Source Separation*, 2003.

S. A. Murphy and A. W. van der Vaart. On profile likelihood. *Journal of the American Statistical Association*, 95:449–485, 2000.

H. C. Ombao, J. A. Raz, R. L. Strawderman, and R. Von Sachs. A simple GCV method of span selection for periodogram smoothing. *Biometrika*, 88:1186–1192, 2001.

J. Pearl. *Causality: Models, Reasoning and Inference*. Cambridge University Press, 2000.

D. T. Pham. Blind separation of instantaneous mixture of sources via an independent component analysis. *IEEE Transactions on Signal Processing*, 44(11):225–229, 1995.

D. T. Pham. Blind separation of instantaneous mixture of sources via the Gaussian mutual information criterion. *Signal Processing*, 81:850–870, 2001a.

D. T. Pham. Contrast functions for blind separation and deconvolution of sources. In *Proceeding of the ICA 2001 Conference*, 2001b.

D. T. Pham. Mutual information approach to blind separation of stationary sources. *IEEE Transactions on Information Theory*, 48(7):1935–1946, 2002.

D. T. Pham. Fast algorithm for estimating mutual information, entropies and score functions. In *Proceedings of the Fourth Symposium on Independent Component Analysis and Blind Source Separation*, 2003.

J. Rissanen. Modeling by Shortest Data Description. *Automatica*, 14:465–471, 1978.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2001.

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

M. Welling and M. Weber. A constrained EM algorithm for independent component analysis. *Neural Computation*, 13(3):677–689, 2001.

A. S. Willsky. Multiresolution statistical models for signal and image processing. *Proceedings of the IEEE*, 90(8):1396–1458, 2002.

A. Ziehe and K.-R. Müller. TDSEP—an efficient algorithm for blind separation using time structure. In *Proceedings of the International Conference on Artificial Neural Networks*, 1998.