

Blind Source Separation and Deconvolution: The Dynamic Component Analysis Algorithm

H. Attias and C.E. Schreiner

Sloan Center for Theoretical Neurobiology and
W.M. Keck Foundation Center for Integrative Neuroscience
University of California at San Francisco
513 Parnassus Avenue
San Francisco, CA 94143-0444

Abstract

We derive a novel family of unsupervised learning algorithms for blind separation of mixed and convolved sources. Our approach is based on formulating the separation problem as a learning task of a spatio-temporal generative model, whose parameters are adapted iteratively to minimize suitable error functions, thus ensuring stability of the algorithms. The resulting learning rules achieve separation by exploiting high-order spatio-temporal statistics of the mixture data. Different rules are obtained by learning generative models in the frequency and time domains, whereas a hybrid frequency/time model leads to the best performance. These algorithms generalize independent component analysis to the case of convolutive mixtures and exhibit superior performance on instantaneous mixtures. An extension of the relative-gradient concept to the spatio-temporal case leads to fast and efficient learning rules with equivariant properties. Our approach can incorporate information about the mixing situation when available, resulting in a ‘semi-blind’ separation method. The spatio-temporal redundancy reduction performed by our algorithms is shown to be equivalent to information-rate maximization through a simple network. We illustrate the performance of these algorithms by successfully separating instantaneous and convolutive mixtures of speech and noise signals.

1 Sources as Dynamic Components

The problem of blind source separation is defined as follows. Consider L independent signal sources (e.g., different speakers in a room) and L' sensors (e.g., microphones at several locations). Each sensor receives a mixture of the source signals. The task is to recover the unobserved sources from the observed sensor signals. This separation should be performed in the absence of any information about the mixing process or the sources, apart from their mutual statistical independence, hence is termed ‘blind’.

Successful techniques for blind separation can have many applications in areas involving processing of multi-sensor signals, such as speech recognition and enhancement, the analysis and classification of biomedical recordings, and target localization and tracking by radar and sonar devices. Such real-world situations generally involve source signals that are delayed and attenuated by different amounts on their way to the different sensors, as well as multi-path propagation, resulting in a situation termed ‘convolutive mixing’. Mathematically, the mixing is described by a matrix of filters operating on the sources. The problem is further complicated by fact that the number of sources L is unknown and may be larger than the number of sensors L' , the source properties (e.g., location) are time-dependent, the mixing may be non-linear due to the impulse response of the medium and sensors, and the signals are corrupted by propagation and sensor noise. Currently, there exists no algorithm that can solve the general problem. The human auditory system, however, can solve it under some conditions for $L' = 2$ (the ‘cocktail party’ effect; see Bregman 1990).

Given the complexity of the actual problem, current work on blind separation focuses on an idealized version thereof where the mixing is square ($L' = L$), invertible, linear, noiseless, and time-independent. Even for this version, significant progress has been made only recently and on a further simplified case where the mixing is instantaneous (non-convolutive), i.e., involves no delays or frequency distortion. A robust and efficient set of algorithms which perform ‘independent component analysis’ (ICA) has been proven effective for a large class of instantaneous mixtures (Comon 1994; Bell and Sejnowski 1995; Cardoso and Laheld 1996; Pham 1996; Pearlmutter and Parra 1997). In that case we seek a separating transformation G_{ij} that, when applied to the sensor signals $y_i(t)$ which are generally correlated, will produce a new set of signals $\hat{x}_i(t)$,

$$\hat{x}_i(t) = \sum_{j=1}^L G_{ij} y_j(t). \quad (1)$$

If successful, the separation results in these new signals being the original sources $x_i(t)$ (to within an order permutation and scaling), and thus mutually statistically independent. Methods to find G_{ij} turn this observation around and impose the independence of $\hat{x}_i(t)$ as a condition on G_{ij} . Expressed in equal-time moments, G_{ij} should be chosen such that the resulting signals satisfy $\langle \hat{x}_i(t)^m \hat{x}_j(t)^n \rangle = \langle \hat{x}_i(t)^m \rangle \langle \hat{x}_j(t)^n \rangle$ for $i \neq j$ and any powers m, n ; the average is taken over time t . Thus, ICA methods attempt to deduce G_{ij} from the equal-time (or ‘spatial’, referring to the sensor coordinate i) sensor statistics.

However, realistic situations are characterized by convolutive mixing, where many delayed and attenuated versions of each source signal arrive at each sensor. Since a sensor signal $y_i(t)$ consists of the sources not only at the present time t but also at preceding times $t' < t$, recovering the sources must, in turn, be done using both present and past sensor signals $y_i(t' \leq t)$. Hence, in contrast with instantaneous mixtures which require a spatial separating transformation G_{ij} (1), convolutive mixtures require a spatio-temporal separating transformation $G_{ij}(t)$,

$$\hat{x}_i(t) = \sum_{j=1}^L \int_0^{\infty} dt' G_{ij}(t') y_j(t - t'). \quad (2)$$

The simple time-dependence $G_{ij}(t) = G_{ij} \delta(t)$ reduces the convolutive (2) to the instantaneous (1) case. In general, $G_{ij}(t)$ has a non-trivial time dependence as it couples mixing with filtering, and cannot be found by ICA methods. In fact, equal-time statistics are not sufficient to deduce G_{ij} even for some instantaneous mixtures, e.g., of colored Gaussian signals, which are consequently inseparable to ICA.

In this paper we present a novel family of unsupervised learning algorithms for blind separation of convolutive mixtures, termed *dynamic component analysis* (DCA). ‘Dynamic’ refers to the temporal nature of separating transformation $G_{ij}(t)$. The DCA algorithms learn $G_{ij}(t)$ from the spatio-temporal sensor statistics. Whereas the sensors $y_i(t)$ are generally correlated, the resulting new signals $\hat{x}_i(t)$ are mutually independent both spatially and temporally, and are termed the ‘dynamic components’ (DC’s) of the sensor data. For successful separation, the DC’s correspond to the original sources.

The key to the DCA approach lies in formulating the convolutive blind separation problem as an *unsupervised learning task of a spatio-temporal generative model*, which in turn becomes an optimization problem using the Kullback-Leibler distance as a cost function.

We now give a quick overview of the DCA procedure. First, we observe that the condition on the equal-time moments described above is not sufficient to identify $G_{ij}(t)$. However, invoking the statistical independence of the sources we can impose the stronger condition on the non equal-time moments, $\langle \hat{x}_i(t)^m \hat{x}_j(t + \tau)^n \rangle = \langle \hat{x}_i(t)^m \rangle \langle \hat{x}_j(t + \tau)^n \rangle$ for $i \neq j$ and any powers m, n at any τ . This is because the amplitude of source i at time t is independent of the amplitude of source $j \neq i$ at any time $t + \tau$. This condition requires processing the sensor signals in time blocks in order to exploit their temporal statistics to deduce $G_{ij}(t)$. Of course, requiring spatio-temporal independence results in separation of instantaneous mixtures as well. Indeed, the resulting instantaneous DCA algorithms are more powerful than the spatial-based ICA ones.

Second, we demonstrate that an effective way to impose spatio-temporal independence is via the use of a generative model (Everitt 1984) for the probability density of the sensor signals. Specifically, we construct a parametric model for the *joint density of the L sensors over N -point time blocks*, $p_y[y_1(t_0), \dots, y_1(t_{N-1}), \dots, y_L(t_0), \dots, y_L(t_{N-1})]$. The adaptive model parameters describe the densities and auto-correlations of the independent sources and the convolutive mixing process. Different models can be constructed using time- and frequency-domain representations. To learn a model we derive the appropriate error function, which measures the information-theory distance between the model and observed sensor densities, and optimize the parameters to minimize this error, resulting in the DCA learning rules. The optimized parameter values provide an estimate either of the mixing process, from which the separating transformation $G_{ij}(t)$ is readily available as its inverse, or of $G_{ij}(t)$ directly.

The DCA approach has several advantages: (1) The generative model formulation which includes adaptive source parameters facilitates effective separation for arbitrary source densities; (2) Stability is ensured by the error function optimization procedure; (3) Thanks to our generalization of the *relative gradient* concept (Cardoso and Laheld 1996; Amari et al. 1996) to the convolutive case, the learning is accelerated and possesses the property of equivariance, which guarantees uniform performance across the space of all invertible mixing processes, and thus robustness; (4) It is sufficiently flexible to allow incorporating information about the mixing filters when available (resulting in ‘semi-blind’ separation), thus improving performance.

DCA is designed to recover the sources from the sensor data when the latter are an unknown linear combination of the former. However, it actually performs the more general task of *spatio-temporal redundancy reduction*, and as such can be applied to any temporal multi-variable data set to extract its dynamic components.

This paper is organized as follows. Section 2 discusses the DCA-I algorithms for separating instantaneous mixtures using different generative models in the time and frequency domains and a ‘hybrid’ time/frequency one, and derives the corresponding error functions and learning rules. The DCA-C algorithms for separating convolutive mixtures are presented in Section 3, based on three analogous models. In Section 4 we discuss the advantage of learning the mixing rather than separating transformation and present the resulting semi-blind separation algorithm DCA-CS. The learning of more general separating transformations using DCA-CR is discussed in Section 5. An equivalent formulation of DCA algorithms in terms of maximizing the *information rate* through a network is presented in Section 6. Finally, in Section 7 we describe an effective use of DCA methods to achieve separation when there are more sensors than sources. Most derivations and technical details are relegated to appendices.

We demonstrate and compare the performance of different DCA algorithms throughout the paper by applying them to instantaneous and convolutive mixtures of speech signals, as well as random signals with different densities and auto-correlations.

Notation

Throughout this paper, vectors are denoted by bold-faced lower-class letters and matrices by bold-faced upper-class letters. Vector and matrix elements are not bold-faced. The complex conjugate of z is z^* . The inverse of a matrix \mathbf{A} is denoted by \mathbf{A}^{-1} , its transposition by \mathbf{A}^T ($A_{ij}^T = A_{ji}$), and its complex transposition by \mathbf{A}^\dagger ($A_{ij}^\dagger = A_{ji}^*$).

Frequency-domain quantities are distinguished from their time-domain counterparts by the symbol \sim above. We work in discrete time and usually consider N -point time blocks, thus $t = t_m$, $m = 0, \dots, N - 1$. To these correspond in the frequency domain the discrete frequencies $\omega = \omega_k = 2\pi k/N$ with $k = 0, \dots, N - 1$, which are related to the actual sound frequencies f_k by $\omega_k = 2\pi f_k/f_s$, where f_s is the sampling frequency. For discrete-time signal processing issues see (Oppenheim and Schaffer 1989).

For example, $\mathbf{x}(t_m)$ is a vector of time-domain signals $x_i(t_m)$. The corresponding frequency-domain vector of signals is $\tilde{\mathbf{x}}(\omega_k)$; the two are related by the discrete Fourier transform (DFT), provided here for

reference:

$$\text{DFT : } \tilde{x}_i(\omega_k) = \sum_{m=0}^{N-1} e^{-i\omega_k m} x_i(t_m), \quad \text{Inverse DFT : } x_i(t_m) = \frac{1}{N} \sum_{k=0}^{N-1} e^{i\omega_k m} \tilde{x}_i(\omega_k). \quad (3)$$

Similarly, $\mathbf{H}(t_m)$ denotes a matrix of filters $H_{ij}(t_m)$; this time-domain representation is called the filter impulse responses. Its DFT $\tilde{\mathbf{H}}(\omega_k)$ contains the filter frequency responses $\tilde{H}_{ij}(\omega_k)$.

This notation will often be simplified by converting the times and frequencies to subscripts: $\mathbf{x}_m = \mathbf{x}(t_m)$, $\tilde{\mathbf{x}}_k = \tilde{\mathbf{x}}(\omega_k)$, and similarly $\mathbf{H}_m = \mathbf{H}(t_m)$, $\tilde{\mathbf{H}}_k = \tilde{\mathbf{H}}(\omega_k)$.

For a filter (or signal) vector \mathbf{g}_m , we define a diagonal matrix \mathbf{D}_m^g which contains it by

$$D_{ij,m}^g = g_{i,m} \delta_{ij}. \quad (4)$$

Finally, we define two linear operations on signals. The linear convolution of $x_{i,m}$ and $y_{j,m}$ is denoted by $*$:

$$(x_i * y_j)_m = \sum_{n=-\infty}^{\infty} x_{i,n} y_{j,m-n}. \quad (5)$$

The convolution produces a new signal $z_m = (x_i * y_j)_m$, $-\infty < m < \infty$. In practice the signals are finite and the actual limits on m, n are determined by their lengths; to use (5) for a finite signal, e.g., $x_{i,m}$, we define $x_{i,m} = 0$ at time points t_m where it is not defined.

The cross-correlation of $x_{i,m}$ and $y_{j,m}$ is denoted by \times :

$$(x_i \times y_j)_m = \sum_{n=-\infty}^{\infty} x_{i,n} y_{j,n+m}. \quad (6)$$

Like the convolution, the cross-correlation produces a new signal $w_m = (x_i \times y_j)_m$ with $-\infty < m < \infty$. Note from (6) that the averaging implied by the term ‘correlation’ extends only over time. The averaging over both time and an ensemble of N -point signals will be denoted by $\langle (x_i \times y_j)_m \rangle$.

We shall often use convolution and cross-correlation in a matrix notation. Thus, $\mathbf{y}_m = (\mathbf{H} * \mathbf{x})_m$ is a signal vector whose i -th entry at time point t_m is given by the convolution $y_{i,m} = \sum_j (H_{ij} * x_j)_m = \sum_{in} H_{ij,n} x_{j,m-n}$. Similarly, the cross-correlation matrix $(\mathbf{x} \times \mathbf{y}^T)_m$ is a $L \times L$ matrix whose ij -element is given by (6). Note from (6) that $m \rightarrow -m$ simply means $(x_i \times y_j)_{-m} = \sum_n x_{i,n} y_{j,n-m}$. It also transposes the cross-correlation matrix while exchanging the order of \mathbf{x}_m and \mathbf{y}_m : $(\mathbf{x} \times \mathbf{y}^T)_{-m} = (\mathbf{y} \times \mathbf{x}^T)_m^T$.

We recall (Oppenheim and Schaffer 1989) that the DFT of a convolution is simply the product of the DFT’s of the convolved signals, hence the DFT of the matrix $(\mathbf{x} * \mathbf{y}^T)_m$ is $\tilde{\mathbf{x}}_k \tilde{\mathbf{y}}_k^T$. The DFT of the cross-correlation matrix $(\mathbf{x} \times \mathbf{y}^T)_{-m}$ is $\tilde{\mathbf{x}}_k \tilde{\mathbf{y}}_k^\dagger$. These relations can be verified using (3).

2 Instantaneous Mixing

In this section we derive the DCA learning rules for separating instantaneous mixtures. We also demonstrate that the use of temporal statistics of the sensor signals facilitates the separation of mixtures which equal-time ICA algorithms (e.g., Bell and Sejnowski 1995) fail to separate.

Note that the instantaneous mixing problem is obtained from the convolutive case in the limit where the propagation delays are negligible compared to the auto-correlation times of the source signals. Hence, algorithms derived in this section can be used to achieve approximate separation of convolutive mixtures when the mixing filters are sufficiently short.

We denote the original, unobserved source signals by $x_{i,m}$ and the observed sensor signals by $y_{i,m}$, $i = 1, \dots, L$, $m = 0, \dots, N - 1$. The $L \times L$ mixing matrix H_{ij} relates them by $y_{i,m} = \sum_j H_{ij} x_{j,m}$, or in matrix notation

$$\mathbf{y}_m = \mathbf{H}\mathbf{x}_m \quad (7)$$

for all m . This mixing is termed ‘instantaneous’ since the sensor signals at t_m depend on the sources at the same, but no earlier, time point. Were the mixing matrix given, its inverse could have been applied to the sensor signals to recover the sources by $\mathbf{x}_m = \mathbf{H}^{-1}\mathbf{y}_m$. In the absence of any information about the mixing, the blind separation problem consists of estimating a separating matrix \mathbf{G} from the observed sensor signals alone. The source signals can then be recovered by

$$\hat{\mathbf{x}}_m = \mathbf{G}\mathbf{y}_m . \quad (8)$$

Generally, the sources can be recovered only to within a scaling factor and an order permutation (see Section 2.4). Hence, $\hat{x}_{i,m} = \lambda_i x_{\pi(i),m}$ for arbitrary scaling factors $\lambda_i \neq 0$ and an arbitrary permutation π of $1, \dots, L$, and \mathbf{G} is a correspondingly scaled and permuted version of \mathbf{H}^{-1} .

In this paper we solve the separation problem by first converting it into an optimization problem. For this purpose we construct a generative model (Everitt 1984) of the observed sensor signals \mathbf{y}_m , where the hidden variables are the unobserved source signals. Since our approach is statistical, we model the *density* of the sensor signals p_y ; note that p_y describes L jointly distributed stochastic processes, and is thus the joint density of all sensor signals at all time points, i.e., $p_y = p_y(y_{1,0}, \dots, y_{1,N-1}, \dots, y_{L,0}, \dots, y_{L,N-1})$. The sensor density can be expressed in terms of the separating matrix \mathbf{G} and the densities of the independent sources. Each source, in turn, is modeled as a stochastic non-Gaussian process described by its marginal (one time-point) density and auto-correlation function, which are parametrized by ξ_i and $g_{i,m}$, respectively, as detailed below.

To complete the optimization formulation, we shall define an error function that measures the distance between our model p_y and the observed p_y^o sensor densities. This error is a function of ξ_i , $g_{i,m}$ and G_{ij} , termed the ‘separation parameters’, which are then optimized to minimize the error, so that the model p_y best approximates the observed p_y^o . The optimal separating matrix \mathbf{G} is subsequently used to recover the sources according to (8).

We assume that the sources are independent, stationary, and zero-mean ($\langle x_{i,m} \rangle = 0$). We also consider their auto-correlations $\langle (x_i \times x_i)_m \rangle = \langle \sum_n x_{i,n} x_{i,n+m} \rangle$. A simple way to model them is to represent the source $x_{i,m}$ as a filtered version of a white (i.e., δ -correlated), zero-mean signal $u_{i,m}$, so that

$$\mathbf{y}_m = \mathbf{H}\mathbf{x}_m , \quad x_{i,m} = (h_i * u_i)_m , \quad (9)$$

where $u_{i,m}$ satisfies

$$\langle u_{i,m} \rangle = 0 , \quad \frac{1}{N} \langle (u_i \times u_i)_m \rangle = \delta_{m,0} \quad (10)$$

and $*$ denotes linear convolution (see (5)). Recall that we are working with N -time point signal segments, hence the $1/N$ factor in (10); $\langle \cdot \rangle$ denotes averaging over an ensemble of such signals. The filter $h_{i,m}$ then determines the source auto-correlations through $\langle (x_i \times x_i)_m \rangle = (h_i \times h_i)_m$. According to (9,10), therefore, the sensor signals are produced by filtering independent white sources individually and then mixing them. Note that since $u_{i,m}$ is white, the power spectrum of $h_{i,m}$ equals that of $x_{i,m}$: $|\tilde{h}_{i,k}|^2 = \langle |\tilde{x}_{i,k}|^2 \rangle$ (see (3)).

For the purpose of estimating the separating matrix, however, rather than considering $h_{i,m}$, it is convenient to use their inverses $g_{i,m}$:

$$u_{i,m} = (g_i * x_i)_m , \quad \mathbf{x}_m = \mathbf{G}\mathbf{y}_m . \quad (11)$$

Here, the $g_{i,m}$ operate on the recovered sources to produce white signals and are therefore termed ‘whitening filters’; their spectra are the inverse source spectra: $|\tilde{g}_{i,k}|^2 = 1/\langle |\tilde{x}_{i,k}|^2 \rangle$. The resulting signals $u_{i,m}$ are

termed ‘whitened sources’. Note that the hat symbol that distinguishes recovered $\hat{x}_{i,m}$ from actual $x_{i,m}$ sources (8) has been omitted. In the rest of this paper, $x_{i,m}$ denotes a recovered source unless otherwise noted.

In order to construct a generative model of the sensor density, we must provide a model *source* density. In fact, we shall be modeling the density p_u of the whitened sources. In the following, we shall formulate generative models in the time and frequency domains, as well as a hybrid frequency/time model, resulting in three different error functions and learning rules.

2.1 DCA-IF: Frequency-domain generative model

It is convenient to work in the frequency domain since the problem simplifies there in the following sense. Applying DFT to (11), we have

$$\tilde{u}_{i,k} = \tilde{g}_{i,k} \tilde{x}_{i,k}, \quad \tilde{\mathbf{x}}_k = \mathbf{G} \tilde{\mathbf{y}}_k. \quad (12)$$

Whereas the time-domain formulation couples the signals at different times t_m (by the convolution in (11)), here we have a separate problem at each ω_k . Of course, these N problems are not independent since they all involve the same parameters $g_{i,m}$ and G_{ij} .

For the whitened source density we use a factorial frequency-domain model

$$p_{\tilde{u}} = \prod_{i=1}^L \prod_{k=0}^{N/2} P_{i,k}(\tilde{u}_{i,k}), \quad (13)$$

where N is assumed even (with no loss of generality). Note that k runs only up to $N/2$ since $\tilde{\mathbf{u}}_{N-k} = \tilde{\mathbf{u}}_k^*$ (see (3)). Also, since for $1 \leq k \leq N/2 - 1$ the Fourier components $\tilde{\mathbf{u}}_k$ are complex, $P_{i,k}$ is in fact the joint distribution of $\text{Re}(\tilde{u}_{i,k})$ and $\text{Im}(\tilde{u}_{i,k})$.

In Appendix A.1 we derive the model sensor density $p_{\tilde{\mathbf{y}}}$ (62) from (12,13). As our error function we choose the Kullback-Leibler (KL) distance (Cover and Thomas 1991) $E(p_{\tilde{\mathbf{y}}}^o, p_{\tilde{\mathbf{y}}})$ (64), an asymmetric measure of the distance between the correct density $p_{\tilde{\mathbf{y}}}^o$ and the model $p_{\tilde{\mathbf{y}}}$. As shown in Appendix A.1, the DCA-IF (I=instantaneous mixing, F=frequency domain) error function is given by

$$E^{DCA-IF} = -\log |\det \mathbf{G}| - \frac{1}{N} \sum_{i=1}^L \sum_{k=0}^{N-1} \log |\tilde{g}_{i,k}| - \frac{1}{N} \sum_{i=1}^L \sum_{k=0}^{N/2} \log P_{i,k}, \quad (14)$$

where the term $\log P_{i,k}$ in (14) actually represents $\langle \log P_{i,k}(\tilde{u}_{i,k} = \tilde{g}_{i,k} \sum_j G_{ij} \tilde{y}_{j,k}) \rangle$, the average being taken over the observed sensor signals $\tilde{\mathbf{y}}_k$. Thus this term is a function of \mathbf{G} , \mathbf{g}_m , and the functional form of $P_{i,k}$, as is the error function itself. We emphasize that the filters $g_{i,m}$ may have any lengths M_i (i.e., $g_{i,m \geq M_i} = 0$) and are usually much shorter than N .

Before deriving the learning rules, let us make the whitened source model (13) more specific. First, we shall use the same parametrized functional form for all sources. This is consistent with our report (Attias and Schreiner 1997), which showed that a large class of natural sounds are characterized by the same parametric functional form of their frequency-domain density. Second, we shall omit the frequency-dependence of $P_{i,k}$. Hence

$$P_{i,k}(\tilde{u}_{i,k}) = P(\tilde{u}_{i,k}, \boldsymbol{\xi}_i), \quad (15)$$

where $\boldsymbol{\xi}_i$ is a vector of parameters for source i . A convenient form for P is a Gaussian mixture with the means, variances and weights of the Gaussians contained in $\boldsymbol{\xi}_i$ (see Appendix C). Note that the form (15) implies white signals, since their power spectra $\langle |\tilde{u}_{i,k}|^2 \rangle$ are frequency-independent.

The separation parameters \mathbf{G} , \mathbf{g}_m and $\boldsymbol{\xi}_i$ should now be optimized to minimize the error (14). This minimization can be done iteratively using the gradient-descent method. As discussed in Appendix A.1, the

learning rules obtained from the ordinary gradient of E^{DCA-IF} with respect to \mathbf{G} and \mathbf{g}_m are less efficient than those obtained from the relative gradient (70), which therefore constitute the DCA-IF learning rules:

$$\begin{aligned}\delta^R \mathbf{G} &= \epsilon \mathbf{G} - \epsilon \sum_m \mathbf{D}_m^g (\boldsymbol{\phi} \times \mathbf{x}^T)_{-m} \mathbf{G}, \\ \delta^R g_{i,m} &= \epsilon g_{i,m} - \epsilon \sum_n (\phi_i \times u_i)_n g_{i,n+m}, \\ \delta \boldsymbol{\xi}_i &= \epsilon \frac{1}{N} \sum_{k=0}^{N/2} \frac{\partial \log P_{i,k}}{\partial \boldsymbol{\xi}_i},\end{aligned}\tag{16}$$

where δ and δ^R denote increments derived from the ordinary and relative gradients of the error, respectively, and ϵ sets the learning rate. The rule for $\boldsymbol{\xi}_i$ is further specified in Appendix C.

Notation. The rule for \mathbf{G} is given in a matrix form, where $(\boldsymbol{\phi} \times \mathbf{x}^T)_{-m}$ is a $L \times L$ matrix whose ij -element is the cross-correlation between $x_{i,m}$ and $\phi_{j,m}$ (see (6) and below). The signal $\phi_{i,m}$ is a non-linear function of the whitened source $u_{i,m}$, termed ‘modified whitened source’. It is defined in the frequency domain by (68). The matrix \mathbf{D}_m^g (see (4)) is a diagonal matrix containing the separating filters \mathbf{g}_m . The summation limits on m in (79) are set by the lengths of $\mathbf{g}_m, \boldsymbol{\phi}_m, \mathbf{x}_m$. In component notation, the increment in \mathbf{G} is given by $\delta G_{ij} = \epsilon G_{ij} - \epsilon \sum_{lmn} g_{i,m} \phi_{i,n+m} x_{l,n} G_{lj}$.

The rules (16) have a form common to all DCA learning rules. They involve three kinds of signals: the (recovered) sources \mathbf{x}_m and whitened sources \mathbf{u}_m , related to each other and to the sensors via (12), and the modified whitened sources $\boldsymbol{\phi}_m$. One can view those signals as forming three successive output layers of a simple network with inputs \mathbf{y}_m and weights \mathbf{G}, \mathbf{g}_m . The weight increments are computed by cross-correlating the different outputs across layers and with the weights. Note that those correlations involve high-order sensor statistics since one output layer is a non-linear modification of the other.

To interpret the learning rules (16) we point out that the rule for \mathbf{G} converges when the cross-correlation $(\phi_i \times x_j)_m = 0$ for $i \neq j$, whereas the rule for \mathbf{g}_m converges when $(\phi_i \times u_i)_m = 0$. Therefore, the first makes the recovered sources $x_i, x_{j \neq i}$ independent and the second attempts to whiten them. Note that we are not interested in the whitened sources \mathbf{u}_m and filters \mathbf{g}_m themselves; however, introducing them into the generative model enables the algorithm to exploit high-order spatio-temporal (rather than just spatial) sensor statistics to achieve separation. In fact, in practice the filters $g_{i,m}$ are of lengths $M_i \ll N$ to minimize model complexity, resulting in \mathbf{u}_m whose spectra differ from the source spectra but may not be white (see Figure 2).

The rules (16) can be used in either batch or on-line learning by processing successive (possibly overlapping) N -point segments of the sensor signals \mathbf{y}_m . In on-line mode, the increments are computed from (16) using the current segment, corresponding to a stochastic gradient descent minimization of the error function (14). In batch mode, the increments are computed by averaging (16) over a long sequence of the sensor signals, resulting in a deterministic gradient descent minimization of (14). In practical applications, the computation of (16) can be accelerated by using their frequency-domain version (73), where FFT can be exploited.

Section 2.4 analyzes the symmetries of the DCA-I error function. Here we only point out that (14) depends on G_{ij} and $g_{i,0}$ only via their product $g_{i,0} G_{ij}$ and hence possesses continuous symmetry, which can be avoided by setting $g_{i,0} = 1$. However, this cannot be done explicitly by keeping $\delta^R g_{i,0} = 0$ in a relative-gradient rule such as (16). To impose this constraint, one must allow $g_{i,0}$ to change by $\delta^R g_{i,0}$ and normalize $G_{ij} \rightarrow G_{ij} g_{i,0}$ and $g_{i,m} \rightarrow g_{i,m} / g_{i,0}$ at each iteration (note that this leaves the error unchanged).

2.2 DCA-IT: Time-domain generative model

We now derive the error function and learning rules for the separation parameters by learning the time-domain generative model (11). As in the frequency-domain case (13), we use a factorial form for the whitened source

density,

$$p_u = \prod_{i=1}^L \prod_{m=0}^{N-1} p_{i,m}(u_{i,m}) . \quad (17)$$

In Appendix A.2 we show that (17) leads to the model sensor density p_y (74), which in turn generates the DCA-IT (I=instantaneous mixing, T=time domain) error function

$$E^{DCA-IT} = -\log |\det \mathbf{G}| - \sum_{i=1}^L \log |g_{i,0}| - \frac{1}{N} \sum_{i=1}^L \sum_{m=0}^{N-1} \log p_{i,m} , \quad (18)$$

where the term $\log p_{i,m}$ in (14) actually represents $\langle \log p_{i,m}(u_{i,m} = (g_i * \sum_j G_{ij} y_j)_m) \rangle$, the average being taken over the observed sensor signals \mathbf{y}_m .

To make the general form (17) more specific we note that, assuming stationary sources, the marginal densities $p_{i,m}$ are independent of the particular time point t_m . Also, we use the same functional form for all whitened sources, parametrized by the vector $\boldsymbol{\xi}_i$. Hence

$$p_{i,m}(u_{i,m}) = p(u_{i,m}, \boldsymbol{\xi}_i) . \quad (19)$$

The learning rules for the separation parameters \mathbf{G} , \mathbf{g}_m and $\boldsymbol{\xi}_i$ are derived in Appendix A.2 for both ordinary- and relative-gradient descent. The relative-gradient rule for \mathbf{G} is more efficient, but, unlike the DCA-IF case, the one for \mathbf{g}_m is not. Here are DCA-IT learning rules:

$$\begin{aligned} \delta^R \mathbf{G} &= \epsilon \mathbf{G} - \epsilon \frac{1}{N} \sum_m \mathbf{D}_m^g (\boldsymbol{\psi} \times \mathbf{x}^T)_{-m} \mathbf{G} , \\ \delta g_{i,m} &= -\epsilon (\boldsymbol{\psi}_i \times x_i)_{-m} , \\ \delta \boldsymbol{\xi}_i &= \epsilon \frac{1}{N} \sum_{m=0}^{N-1} \frac{\partial \log p_{i,m}}{\partial \boldsymbol{\xi}_i} . \end{aligned} \quad (20)$$

The signals $\boldsymbol{\psi}_{i,m}$, termed ‘modified whitened sources’ (but are different than $\phi_{i,m}$ appearing in (16)), are non-linear functions of the whitened sources $u_{i,m}$ and are defined in (78). \mathbf{D}_m^g (see (4)) is a diagonal matrix containing \mathbf{g}_m . These rules should be iterated keeping $g_{i,0} = 1$, $\delta g_{i,0} = 0$. Like the DCA-IF rules, they can be used for either batch or on-line learning, and their performance is accelerated by exploiting FFT in their frequency-domain version (79).

Note that, when considering single-time-point segments ($N = 1$) such that $g_{i,m} = \delta_{m,0}$, and fixed whitened source densities $p_{i,m}$, the rules for \mathbf{g}_m and $\boldsymbol{\xi}_m$ in (20) become irrelevant, and the rule for \mathbf{G} reduces to Bell and Sejnowski’s (1995) ICA rule.

2.3 DCA-IFT: Frequency/time-domain generative model

In the DCA-IF error function (14), both the sources and whitening filters appear as frequency-domain quantities $\tilde{\mathbf{u}}_k, \tilde{\mathbf{g}}_k$, whereas in the DCA-IT error (18) they both appear as time-domain quantities $\mathbf{u}_m, \mathbf{g}_m$. Two ‘hybrid’ error functions can also be derived. The DCA-IFT error, derived in Appendix A.3, includes the sources and whitening filters as time- and frequency-domain quantities, respectively:

$$E^{DCA-IFT} = -\log |\det \mathbf{G}| - \frac{1}{N} \sum_{i=1}^L \sum_{k=0}^{N-1} \log |\tilde{g}_{i,k}| - \frac{1}{N} \sum_{i=1}^L \sum_{m=0}^{N-1} \log p_{i,m} , \quad (21)$$

where the full form of $p_{i,m}$ is specified below (18). Notice that (21) is a cross between (14) and (18). The DCA-IFT learning rules are similarly a cross between (16) and (20):

$$\delta^R \mathbf{G} = \epsilon \mathbf{G} - \epsilon \sum_m \mathbf{D}_m^g (\boldsymbol{\psi} \times \mathbf{x}^T)_{-m} \mathbf{G} ,$$

$$\begin{aligned}
\delta^R g_{i,m} &= \epsilon g_{i,m} - \epsilon \sum_n (\psi_i \times u_i)_n g_{i,n+m}, \\
\delta \xi_i &= \epsilon \frac{1}{N} \sum_{m=0}^{N-1} \frac{\partial \log p_{i,m}}{\partial \xi_i}.
\end{aligned} \tag{22}$$

We present this hybrid error not merely to make a technical point. A comparison between all the DCA-I learning rules, presented in Section 2.5, demonstrates that the DCA-IFT rules are the most efficient. A fourth error function, DCA-ITF, which combines frequency-domain sources with time-domain filters, can also be derived, but results in slow learning.

2.4 Symmetries in parameter space

As mentioned below (7), it is well-known that the sources can be recovered only to within a scaling factor and an order permutation. Indeed, the observed signals $y_{i,m}$ in (7) could also have arisen from the scaled and permuted sources $x'_{j,m} = \lambda_j x_{\pi(j),m}$ and mixing matrix $H'_{ij} = H_{i\pi(j)}/\lambda_j$, where $\lambda_j \neq 0$ are arbitrary factors and $\pi(j)$ is an arbitrary permutation of $1, \dots, L$. This implies that there exists an infinite family of separating matrices $\mathbf{G} \in \{\mathbf{G}'\}$ which cannot be distinguished from $\mathbf{G} = \mathbf{H}^{-1}$ on the basis of the observed signals alone, and whose application to the \mathbf{y}_m would recover the original sources while modifying their intensities and order.

This indistinguishability is manifested in the corresponding error function E as a family of global minima, since all separating matrices \mathbf{G}' satisfy $E(\mathbf{G}') = E(\mathbf{H}^{-1}) = \min_{\mathbf{G}} E(\mathbf{G})$. Hence, the global minimum of the error is invariant under the family of transformations $\mathbf{H}^{-1} \rightarrow \mathbf{G}'$. In fact, not only the minimum but the error function itself is invariant under this family, as we shall see shortly. In other words, the error possesses symmetries in parameter space.

The symmetries of the error function may be discrete (e.g., permutation) or continuous (e.g., scaling). The presence of symmetries does not cause a problem in principle, since the resulting family of minima includes only separating solutions. In practice, however, a continuous symmetry may lead to a slow convergence since the algorithm may spend long periods of time on equipotential surfaces. It is therefore advantageous for the error to have minimal symmetry. In the following we analyze the symmetries of the DCA-I error functions.

It is easy to see that the errors DCA-IF (14), DCA-IT (18) and DCA-IFT (21) all possess permutation symmetry, i.e., are invariant under the transformation $G_{ij} \rightarrow G'_{ij} = G_{\pi(i)j}$, $g_{i,m} \rightarrow g'_{i,m} = g_{\pi(i),m}$, $\xi_i \rightarrow \xi'_i = \xi_{\pi(i)}$ for an arbitrary permutation π .

Regarding continuous symmetries, we observe from (12) that transforming to $G'_{ij} = \alpha_i G_{ij}$, $\tilde{g}'_{i,k} = \tilde{z}_{i,k} \tilde{g}_{i,k}$ for arbitrary $\alpha_i, \tilde{z}_{i,k} \neq 0$ results in $\tilde{\mathbf{u}}'_k$ with spectra $\langle |\tilde{u}'_{i,k}|^2 \rangle = \alpha_i^2 |\tilde{z}_{i,k}|^2 \langle |\tilde{u}_{i,k}|^2 \rangle$. However, the form (15) (or (19)) restricts those spectra to be white, i.e. ω_k -independent, thus only $\tilde{z}_{i,k} = \beta_i e^{i\theta_{i,k}}$ for $\beta_i > 0$ will leave the errors invariant.

We are left with the continuous symmetries parametrized by α_i, β_i and $\theta_{i,k}$. To fix β_i , we choose the source parameters ξ_i so as to make the variance $\langle |\tilde{u}_{i,k}|^2 \rangle$ computed from p (15) (or, equivalently, $\langle u_{i,m}^2 \rangle$ computed from p (19)) a source-independent constant, e.g., 1. Then $\beta_i = 1/|\alpha_i|$. Next, to fix α_i we allow only whitening filters that satisfy $g_{i,0} = 1$. Thus, the above transformation $\tilde{g}'_{i,k} = (e^{i\theta_{i,k}}/\alpha_i)\tilde{g}_{i,k}$ will leave the error invariant only if $\alpha_i = \sum_k e^{i\theta_{i,k}} \tilde{g}_{i,k}/N$.

Finally, we note that, for general P and p , the errors are not invariant under a change in the phases $\theta_{i,k}$, which therefore do not form a continuous symmetry. However, certain choices of P do leave the error phase-invariant, e.g., $P \propto e^{-|\tilde{u}_{i,k}|}$. In such cases we can restrict the phases $\theta_{i,k}$ by choosing the lengths M_i of the whitening filters $g_{i,m}$ to be much smaller than N , the number of time points (or frequencies), thus imposing the $L(N - M)$ conditions $g_{i,m} = 0$ for $M \leq m \leq N - 1$ on the $LN/2$ phases. Note that $N \gg M_i$ still does not guarantee a unique solution for $\theta_{i,k}$, although it limits the allowed phases to a small and possibly discrete set.

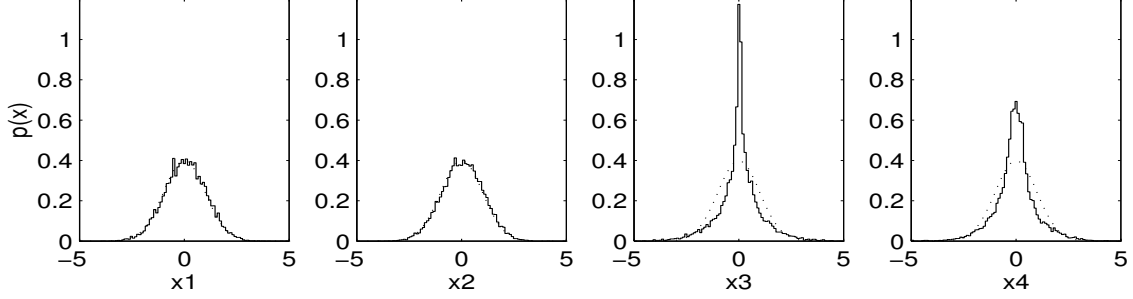


Figure 1: Probability densities of the colored Gaussian signals x_1 , x_2 and the speech signals x_3 , x_4 (solid line), compared with a Gaussian (dotted line). All signals are scaled to have unit variance.

2.5 Results

We used the DCA-I rules (16,20,22) to separate an instantaneous mixture of four 10-second long signals. Two speech signals $x_{3,m}$, $x_{4,m}$ were obtained from a commercial CD at the original sampling rate of 44.1kHz and down-sampled to $f_s = 4.41$ kHz. In addition, two colored Gaussian signals $x_{1,m}$, $x_{2,m}$ were created by generating temporally independent signals $u_{i,m}$ by a random number generator at this sampling rate and were colored by filters $h_{i,m}$ (see (9)), making each signal temporally correlated. The densities of those signals, scaled to have unit variance, are shown in Figure 1. As typical for natural sounds (Attias and Schreiner 1997), the speech signals have a sharply peaked density.

The signals were mixed by the arbitrary matrix

$$\mathbf{H} = \begin{pmatrix} 0.4380 & 0.7884 & -0.3542 & -0.5844 \\ 0.9691 & -1.5841 & -0.8480 & 0.4336 \\ -0.3856 & -0.0050 & 0.5450 & -0.7037 \\ -0.7057 & -0.8339 & -0.1307 & -1.2325 \end{pmatrix}. \quad (23)$$

We iterated the learning rules in batch mode, using $N = 512$ -point (116msec) overlapping blocks to compute the required cross-correlations. We started from a random matrix \mathbf{G} and filters \mathbf{g}_m with $0 \leq m \leq 4$, kept a constant learning rate $\epsilon = 0.05$, and stopped when the relative increment of the separating matrix $\frac{1}{\epsilon}[\sum_{ij}(\delta G_{ij})^2 / \sum_{ij} G_{ij}^2]^{1/2}$ and whitening filters $\frac{1}{\epsilon}[\sum_{im}(\delta g_{i,m})^2 / \sum_{im} g_{i,m}^2]^{1/2}$ both decreased below 10^{-4} (see Figure 3 for the exponential convergence of the error functions). The resulting separating matrices, when operating on \mathbf{H} (23), produced

$$\mathbf{G}^{DCA-IF}\mathbf{H} = \begin{pmatrix} \mathbf{2.5673} & 0.0420 & -0.0261 & -0.0019 \\ -0.0123 & -\mathbf{2.7358} & -0.0295 & -0.0112 \\ 0.0006 & -0.0155 & \mathbf{7.4880} & -0.0172 \\ -0.0158 & 0.0030 & -0.0041 & -\mathbf{4.5740} \end{pmatrix}, \quad (24)$$

$$\mathbf{G}^{DCA-IT}\mathbf{H} = \begin{pmatrix} \mathbf{1.9891} & 0.0309 & -0.0207 & 0.0031 \\ -0.0062 & -\mathbf{2.1202} & -0.0299 & -0.0157 \\ 0.0018 & -0.0191 & \mathbf{4.7581} & -0.0074 \\ -0.0059 & 0.0101 & -0.0086 & -\mathbf{2.8909} \end{pmatrix}, \quad (25)$$

$$\mathbf{G}^{DCA-IFT}\mathbf{H} = \begin{pmatrix} \mathbf{1.9909} & 0.0303 & -0.0196 & 0.0027 \\ -0.0072 & -\mathbf{2.1326} & -0.0293 & -0.0155 \\ -0.0029 & -0.0172 & \mathbf{5.0227} & -0.0069 \\ -0.0077 & 0.0062 & -0.0070 & -\mathbf{2.8590} \end{pmatrix}, \quad (26)$$

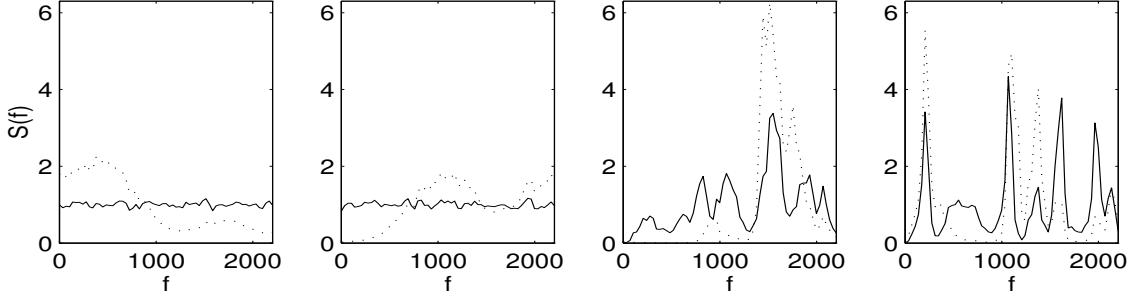


Figure 2: Power spectra of the whitened sources obtained by DCA-IF (solid line) compared with the original sources (dotted line). 5-point whitening filters were used.

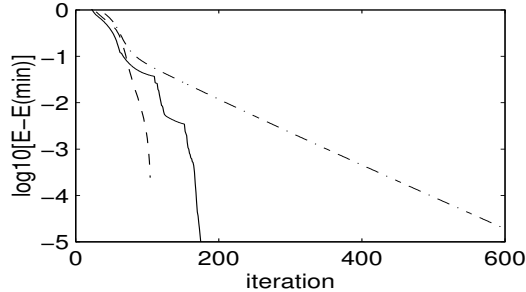


Figure 3: Convergence of the DCA-IF (solid line), DCA-IT (dashed-dotted line), and DCA-IFT (dashed line) error functions to their minimum.

Notice that each row in the three matrices above has a single element (bold-faced) that is significantly larger than zero. Thus the separated signals $\mathbf{G}\mathbf{y}_m$ would each consist of one of the (scaled) sources with the rest being attenuated by more than 30dB. A larger attenuation can generally be obtained by increasing the sample size. No attempt was made to optimize the minimization process, e.g., by reducing ϵ with increasing step number.

We point out that the two Gaussian signals were also separated by DCA-I. To emphasize this point we applied Bell and Sejnowski's (1995) ICA to the sensors, resulting in

$$\mathbf{G}^{ICA}\mathbf{H} = \begin{pmatrix} \mathbf{0.9559} & \mathbf{1.4828} & 0.0316 & 0.0175 \\ \mathbf{1.4662} & \mathbf{-0.9258} & -0.0506 & -0.0380 \\ 0.0078 & -0.0432 & \mathbf{1.9740} & -0.0056 \\ -0.0317 & 0.0260 & -0.0042 & \mathbf{-1.9677} \end{pmatrix}. \quad (27)$$

Whereas each of the speech signals were separated by ICA (see bottom-right 2×2 matrix), the Gaussian signals remained mixed, as manifested by the non-diagonal top-left 2×2 matrix in (27). This illustrates the fact, pointed out by Pearlmutter and Parra (1997), that a mixture of Gaussian signals can be separated only by exploiting their temporal statistics, which is used by, e.g., DCA-IT (20) through the cross-correlation $(\boldsymbol{\psi} \times \mathbf{x}^T)_m$, but is ignored by ICA.

In Figure 2 we show the power spectra of the recovered sources \mathbf{x}_m after processing them by the whitening filters \mathbf{g}_m (i.e., $\langle |\tilde{\mathbf{u}}_k|^2 \rangle$ for the signals \mathbf{u}_m in (11)). Those spectra were computed using 64-point DFT with overlapping windows. A comparison with the original source spectra shows that \mathbf{g}_m indeed act to equalize the power in all frequencies. Since the learned filters were short (5 time points) compared to the auto-correlation time of the speech sources, the latter have not been completely whitened.

To further demonstrate the effectiveness of DCA-I, we mixed two white signals $x_{1,m}, x_{2,m}$ with a uniform density ($p(x_i) = 1$ for $-0.5 \leq x_i \leq 0.5$ and $p(x_i) = 0$ otherwise) by the matrix \mathbf{H} (28) and applied the

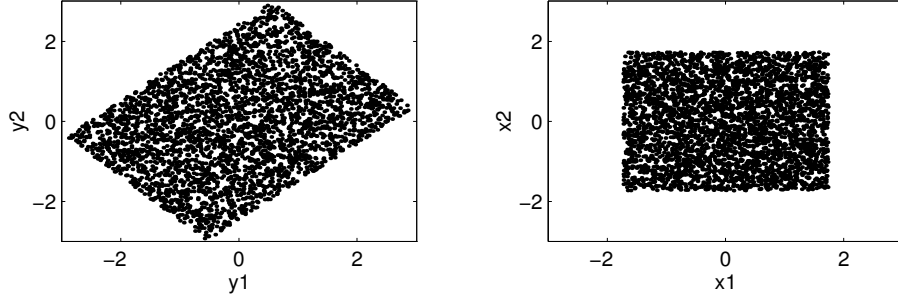


Figure 4: Joint density $p_y(y_1, y_2)$ of sensor (left) and $p_x(x_1, x_2)$ of separated source (right) signals, for instantaneous mixing of two uniformly distributed sources.

DCA-IFT learning rules (16) to the mixtures, resulting in

$$\mathbf{H} = \begin{pmatrix} -1.0107 & 0.6912 \\ -0.9605 & -0.7586 \end{pmatrix}, \quad \mathbf{G}^{DCA-IFT} \mathbf{H} = \begin{pmatrix} -1.8699 & 0.0052 \\ -0.0113 & -1.9457 \end{pmatrix}. \quad (28)$$

The nearly vanishing non-diagonal elements in the matrix on the right indicate almost complete separation, with more than 30dB attenuation of the undesired sources. To illustrate the separation performed by DCA-IFT, we display in Figure 4 the joint density of the mixtures, $p_y(y_1, y_2)$, against the joint density of the separated sources, $p_x(x_1, x_2)$. Note that uniform sources have negative kurtosis and therefore cannot be separated by ICA (Bell and Sejnowski 1995).

3 Convolutive Mixing

In realistic situations, the signal from a given source arrives at the different sensors at different times due to propagation delays. Denoting by d_{ij} the time required for propagation from source j to sensor i , this mixing situation can be described by

$$y_{i,m} = \sum_{j=1}^L H_{ij} x_{j,m-d_{ij}}. \quad (29)$$

More generally, sensor i may receive several progressively delayed and attenuated versions of source signal j , due to the multi-path propagation in a reflective environment, creating multiple echoes. Each version may also be distorted by the frequency response of the propagation medium and sensors. This general convolutive mixing situation is described by

$$\mathbf{y}_m = (\mathbf{H} * \mathbf{x})_m, \quad (30)$$

which in component notation reads $y_{i,m} = \sum_j (H_{ij} * x_j)_m = \sum_{j,n} H_{ij,n} x_{j,m-n}$. Convolutive mixing thus involves mixing coupled with filtering. Technically, the simple mixing matrix \mathbf{H} of the instantaneous case (7) becomes a matrix of filters \mathbf{H}_m , where $H_{ij,m}$ is the impulse response of the filter operating on source signal j on its way to sensor i . Viewed another way, (30) says that the sensor signals \mathbf{y}_m at time t_m are generated not only by a mixture of the source signals \mathbf{x}_m originating at that same time, but also by mixtures of the delayed signals \mathbf{x}_{m-n} that originated at previous times $t_m - t_n$, with a different mixing matrix \mathbf{H}_n for each delay $t_n \geq 0$.

However, the description (30) is problematic since it does not determine the mixing filters \mathbf{H}_m uniquely. This can be seen when expressing the sources as filtered white processes as in (9):

$$\mathbf{y}_m = (\mathbf{H} * \mathbf{x})_m, \quad x_{i,m} = (h_i * u_i)_m. \quad (31)$$

The sensor signals $y_{i,m} = \sum_j (H'_{ij} * u_j)_m$ are thus given by mixing and filtering the whitened sources $u_{i,m}$ using the mixing filter matrix $H'_{ij,m} = (H_{ij} * h_j)_m$. However, there exists a whole family of filters $H_{ij,m}, h_{j,m}$ that produce the actual $H'_{ij,m}$ upon convolution, only one of which corresponds to the situation at hand. This can be seen clearly in the frequency domain where $\tilde{H}'_{ij,k} = \tilde{H}_{ij,k} \tilde{h}_{j,k}$: for any set of complex $\tilde{z}_{j,k} \neq 0$, one can transform $\tilde{H}_{ij,k} \rightarrow \tilde{H}_{ij,k} \tilde{z}_{j,k}$ and $\tilde{h}_{j,k} \rightarrow \tilde{h}_{j,k} / \tilde{z}_{j,k}$ while leaving the observed $y_{i,m}$ unchanged. In other words, given the observed signals alone, the spectra and phases of the sources are indistinguishable from those of the mixing filters. Hence, in the absence of any information about the convolutive mixing process and the sources (the blind case), only the whitened sources can be recovered. Section 4 on semi-blind separation shows how such information, when available, can be incorporated into the separation method to facilitate recovering the sources unwhitened.

In the present section we shall be focusing on estimating the separating filter matrix $G_{ij,m}$ which recovers the whitened sources $u_{i,m}$ through

$$\mathbf{u}_m = (\mathbf{G} * \mathbf{y})_m, \quad (32)$$

where the properties of $u_{i,m}$ are described in (10). As in the instantaneous case, the problem will be given an optimization formulation by learning a generative model in either the frequency, time, or frequency/time domains, resulting in different error functions and learning rules.

3.1 DCA-CF: Frequency-domain generative model

In the frequency domain, (32) becomes

$$\tilde{\mathbf{u}}_k = \tilde{\mathbf{G}}_k \tilde{\mathbf{y}}_k. \quad (33)$$

A comparison with (12) shows that the separating matrix and filters $\mathbf{G}, \tilde{\mathbf{g}}_k$ of the instantaneous case are generalized to a matrix of filters $\tilde{\mathbf{G}}_k$ for convolutive mixing.

To generate a model sensor density $p_{\tilde{\mathbf{y}}}$, we start from the factorial whitened source density $p_{\tilde{\mathbf{u}}}$ (13). In Appendix B.1 we derive $p_{\tilde{\mathbf{y}}}$ (82) and the resulting DCA-CF (C=convolutive mixing, F=frequency domain) error function

$$E^{DCA-CF} = -\frac{1}{N} \sum_{k=0}^{N-1} \log |\det \tilde{\mathbf{G}}_k| - \frac{1}{N} \sum_{i=1}^L \sum_{k=0}^{N/2} \log P_{i,k}, \quad (34)$$

where $\log P_{i,k}$ stands for $\langle \log P_{i,k}(\tilde{u}_{i,k} = \sum_j \tilde{G}_{ij,k} \tilde{y}_{j,k}) \rangle$ and the average is taken over the observed $\tilde{\mathbf{y}}_k$. The filters \mathbf{G}_m are M -point long (i.e., \mathbf{G}_m may be non-zero only for $m = 0, \dots, M-1$) and are usually much shorter than N .

The error (34) is now minimized with respect to \mathbf{G}_m and ξ_i using the gradient-descent method. As in the instantaneous case, we show in Appendix B.1 that the resulting learning rule for \mathbf{G}_m (87) derived from the ordinary gradient of E^{DCA-CF} is quite expensive, requiring the inversion of the complex $L \times L$ matrix $\tilde{\mathbf{G}}_k$ for each of the $N/2$ frequencies ω_k at each iteration. However, it is shown there that the concept of the relative gradient, introduced above for the DCA-I rules, can be extended to the convolutive case and produce efficient rules which avoid matrix inversions. The resulting DCA-CF learning rules are given by

$$\begin{aligned} \delta^R \mathbf{G}_m &= \epsilon \mathbf{G}_m - \epsilon \sum_n (\phi \times \mathbf{u}^T)_n \mathbf{G}_{n+m}, \\ \delta \xi_i &= \epsilon \frac{1}{N} \sum_{k=0}^{N/2} \frac{\partial \log P_{i,k}}{\partial \xi_i}, \end{aligned} \quad (35)$$

where $\phi_{i,m}$ are the modified whitened sources defined in (68), and the parametrized form (15) was used. The rule for the adaptive source density parameters ξ_i is further specified in Appendix C. In component

notation, the increment in \mathbf{G}_m is $\delta G_{ij,m} = \epsilon G_{ij,m} - \epsilon \sum_{l,n,n'} \phi_{i,n} u_{l,n+n'} G_{lj,n'+m}$. Note the formal similarity between this rule and the DCA-IF rule for $g_{i,m}$ in (16), which stems from the fact that $u_{i,m} = (g_i * x_i)_m$ in (11) is the one-dimensional version of (32) with \mathbf{y}_m replaced by \mathbf{x}_m .

The rules (35) have the form common to all DCA learning rules (compare with DCA-IF (16) and comments following it). It involves the whitened (recovered) sources \mathbf{u}_m , related to the sensors via (33), and the modified whitened sources ϕ_m . Those signals can be viewed as forming successive output layers of a simple network with inputs \mathbf{y}_m and weights \mathbf{G}_m . The weight increments are computed by cross-correlating the different outputs across layers and with the weights. Those cross-correlations involve high-order sensor statistics since one output layer is a non-linear function of the other.

Note that the rules (35) converge when the cross-correlation $(\phi_i \times u_j)_m = \delta_{ij} \delta_{m,0}$, meaning $u_{i,m}$ are mutually independent and white.

3.2 DCA-CT: Time-domain generative model

To construct a time-domain model sensor density p_y corresponding to (32) we borrow the factorial whitened source density (17) from the instantaneous case. In Appendix B.2 we derive p_y (91), from which the DCA-CT (C=convolutive mixing, T=time domain) error function is obtained:

$$E^{DCA-CT} = -\log |\det \mathbf{G}_0| - \frac{1}{N} \sum_{i=1}^L \sum_{m=0}^{N-1} \log p_{i,m}, \quad (36)$$

where $\log p_{i,m}$ stands for $\langle \log p_{i,m}(u_{i,m} = \sum_{j,n} G_{ij,n} y_{j,m-n}) \rangle$ and the average is taken over the observed \mathbf{y}_m .

The DCA-CT learning rules for the separation parameters, derived from (36) in Appendix B.2, are given by

$$\begin{aligned} \delta^R \mathbf{G}_0 &= \epsilon \mathbf{G}_0 - \epsilon (\boldsymbol{\psi} \times \mathbf{y}^T)_0 \mathbf{G}_0^T \mathbf{G}_0, & \delta \mathbf{G}_m &= -\epsilon (\boldsymbol{\psi} \times \mathbf{y}^T)_{-m}, \\ \delta \boldsymbol{\xi}_i &= \epsilon \frac{1}{N} \sum_{m=0}^{N-1} \frac{\partial \log p_{i,m}}{\partial \boldsymbol{\xi}_i}, \end{aligned} \quad (37)$$

where $\psi_{i,m}$ are the modified whitened sources defined by (78), and the parametrized form (19) was used. In component notation $(\psi_i \times y_j)_{-m} = \sum_n \psi_{i,n} y_{j,n-m}$.

The rules (37) are ordinary-gradient learning rules except for $m = 0$. As in the case of \mathbf{g}_m in (20), the optimization of the time-domain error does not benefit from the relative gradient approach; in fact, as shown in Appendix B.2, the relative-gradient rule for \mathbf{G}_m (97) is more complicated and less efficient than (37).

3.3 DCA-CFT: Frequency/time-domain generative model

As in the instantaneous case, a hybrid frequency/time error function can also be derived, which includes the separating filters in the frequency and the whitened sources in the time domain. This error, which is a cross between (34) and (36), is given by (see Appendix B.3)

$$E^{DCA-CFT} = -\frac{1}{N} \sum_{k=0}^{N-1} \log |\det \tilde{\mathbf{G}}_k| - \frac{1}{N} \sum_{i=1}^L \sum_{m=0}^{N-1} \log p_{i,m}, \quad (38)$$

where the full expression for $p_{i,m}$ is given below (36). The corresponding DCA-CFT learning rule for \mathbf{G}_m , obtained from the relative gradient of (38), is

$$\begin{aligned} \delta^R \mathbf{G}_m &= \epsilon \mathbf{G}_m - \epsilon \sum_n (\boldsymbol{\psi} \times \mathbf{u}^T)_n \mathbf{G}_{n+m}, \\ \delta \boldsymbol{\xi}_i &= \epsilon \frac{1}{N} \sum_{m=0}^{N-1} \frac{\partial \log p_{i,m}}{\partial \boldsymbol{\xi}_i}, \end{aligned} \quad (39)$$

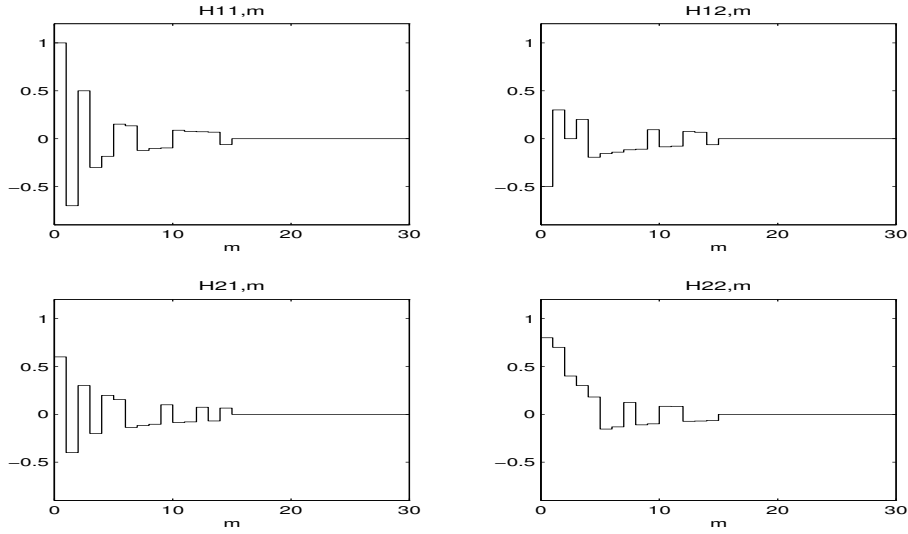


Figure 5: Impulse response of the mixing filters $H_{ij,m}$.

and has the same form as the DCA-CF rule (34) but with the modified whitened sources ϕ_m (68), defined in terms of their frequency-domain density, replaced by ψ_m (78) which is derived from their time-domain density.

3.4 Symmetries in parameter space

Whereas for instantaneous mixing the sources can be recovered to within an order permutation and scaling, in the convolutive case (as mentioned below (31)) the scaling becomes arbitrary filtering. This is manifested in the error corresponding to (31), $E = -\sum_k \log |\det \mathbf{G}_k| / N - \sum_{ik} \log |\hat{g}_{i,k}| / N - \sum_{im} \log p_{i,m} / N$ (in the hybrid FT approach; compare to the semi-blind error (43) and the surrounding discussion), which is invariant under the transformation $\hat{g}_{i,k} \rightarrow \tilde{z}_{i,k} \hat{g}_{i,k}$, $\tilde{G}_{ij,k} \rightarrow \tilde{G}_{ij,k} / \tilde{z}_{i,k}$, with the frequency-dependent scaling factors $\tilde{z}_{i,k}$ being the arbitrary filters $z_{i,m}$, as long as the separating filters are not constrained. Hence, the arguments of the DCA-C error functions do not include the whitening filters $g_{i,m}$. In Section 4 we reintroduce the latter and show how to make inaccessible as many $z_{i,m}$ as possible by constraining the separating filters according to available information.

The discussion of the instantaneous case in Section 2.4 can be repeated to show that, after fixing the source parameters ξ_i in (15,19) to make the variances $\langle |\hat{u}_{i,k}|^2 \rangle$ and $\langle u_{i,m}^2 \rangle$ source-independent, we are left with the order permutation $\tilde{G}_{ij,k} \rightarrow \tilde{G}_{\pi(i)j,k}$ and the continuous transformation $\tilde{G}_{ij,k} \rightarrow e^{i\theta_{i,k}} \tilde{G}_{ij,k}$. However, the latter does not leave the errors invariant, except for special choices of the whitened source densities, as discussed there.

The DCA-CF error (34) forms an exception, however, since it appears to be invariant not only under the source ordering π but under an arbitrary permutation π_k at each frequency, which may lead to low separation quality. Note that the errors DCA-CT, CFT do not possess this permutation invariance. Nevertheless, this invariance can be restricted by choosing the length M of the separating filters $G_{ij,m}$ to be much smaller than N , thus imposing the $L^2(N-M)$ conditions $G_{ij,m} = 0$ for $M \leq m \leq N-1$. If $G_{ij,m}$ are the desired separating filters, then $G'_{ij,m} = \sum_k e^{i\omega_k m} \tilde{G}_{\pi_k(i)j,k} / N$, albeit formally leaving the error invariant, are generally longer than M for M/N sufficiently small, and thus correspond to inaccessible minima. In practical experiments we found DCA-CF to be no less effective than CT and CFT.

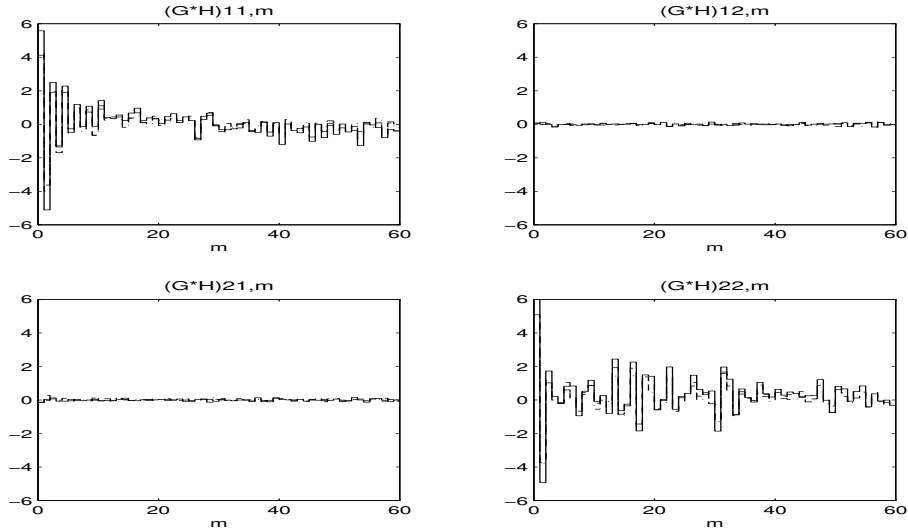


Figure 6: Convolution of the separating filters $G_{ij,m}$ learned by DCA-CF (solid line), DCA-CT (dashed-dotted line), and DCA-CFT (dashed line) with the mixing filters $H_{ij,m}$.

3.5 Results

To demonstrate the performance of the DCA-C algorithms we present an experiment in which we applied them to a convolutive mixture of speech signals. We mixed two 10sec-long signals, obtained from a commercial CD at the original sampling rate of 44.1KHz and down-sampled to $f_s = 4.41$ KHz, by filters \mathbf{H}_m whose impulse response is displayed in Figure 5. We then used the learning rules (35,37,39) to find the separating filters \mathbf{G}_m . The signals were processed in $N = 512$ -point overlapping blocks with a constant learning rate of $\epsilon = 0.05$. The iteration stopped when the relative increment of separating filters, $\frac{1}{\epsilon}[\sum_{ijm}(\delta G_{ij,m})^2 / \sum_{ijm} G_{ij,m}^2]^{1/2}$, decreased below 10^{-4} .

It is important to allow the learned separating filters to be sufficiently long to be able to invert the mixing filters. In the present example we chose 60-point separating filters.

For the frequency-domain whitened source density employed in DCA-CF we used the exponential form $P(\tilde{u}_{i,k}) \propto e^{-|\tilde{u}_{i,k}|/\sqrt{N}}$, where the N -scaling arises from the fact that $\langle |\tilde{u}_{i,k}|^2 \rangle = N \langle u_{i,m}^2 \rangle$. This form approximates well the density of a large class of natural sounds, as we reported in (Attias and Schreiner 1997). We also experimented with $P(\tilde{u}_{i,k}) = P_1(\text{Re}(\tilde{u}_{i,k})/\sqrt{N})P_1(\text{Im}(\tilde{u}_{i,k})/\sqrt{N})$, where P was either the sigmoid-derivative form $P(v) \propto e^{-v}/(1+e^{-v})^2$ used by Bell and Sejnowski (1995), or a mixture of Gaussians parametrized by ξ_i (see Appendix C). However, the simple exponential form was sufficient to achieve separation in all our experiments with DCA-CF. Similarly, for DCA-CT, CFT we found the simple exponential form $p(u_{i,m}) = e^{-|u_{i,m}|}$ to be appropriate.

To demonstrate that separation has actually been accomplished, we present in Figure 6 the convolution with the mixing filters $(\mathbf{G} \star \mathbf{H})_{ij,m} = \sum_n G_{il,n} H_{lj,m-n}$ of all three separating filters $\mathbf{G} = \mathbf{G}^{DCA-CF}, \mathbf{G}^{DCA-CT}, \mathbf{G}^{DCA-CFT}$.

The non-diagonal filters $(\mathbf{G} \star \mathbf{H})_{i \neq j,m}$ are strongly attenuated (≥ 30 dB) compared to the diagonal ones, indicating high separation quality with low cross-talk. Note that the recovered sources have modified power spectra, as is evident from the fact that $(\mathbf{G} \star \mathbf{H})_{ii,m} \neq 0$ for $m > 0$.

The separating filters learned by DCA-CF are shown for illustration in Figure 7, and are similar to those learned by the other two algorithms.

An interesting comparison between the different algorithms is presented in Figure 8, which shows that the frequency-domain algorithms achieve separation significantly faster than the time-domain one. In particular, the hybrid approach DCA-CFT was the fastest to converge. Note that this figure shows batch learning; convergence in on-line mode for DCA-CF, CFT was achieved in 15–20 passes through the data. To emphasize

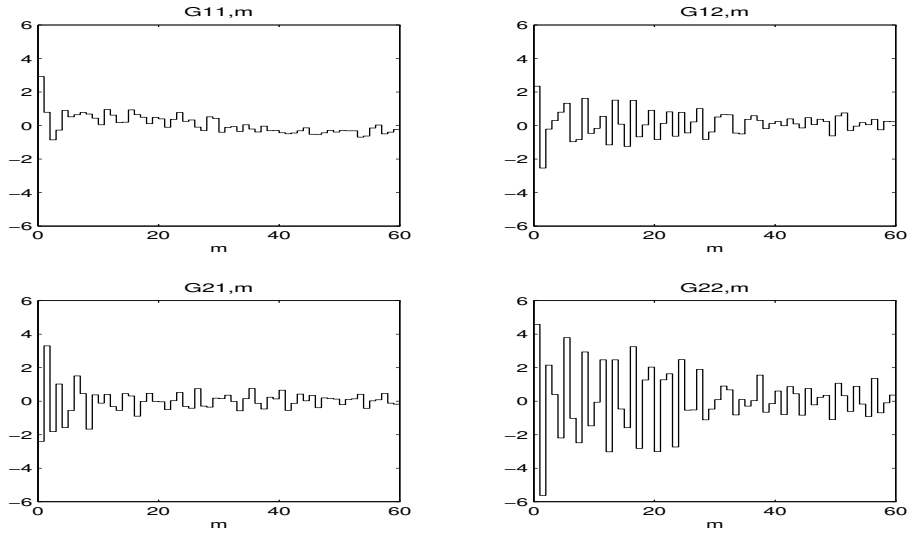


Figure 7: Impulse response of the separating filters learned by DCA-CF.

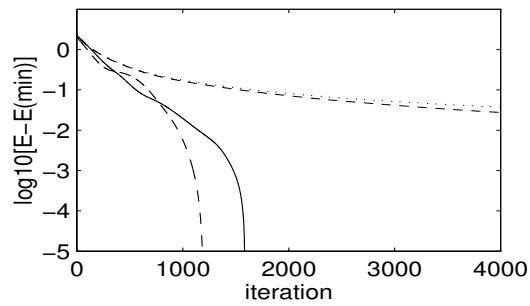


Figure 8: Convergence of the DCA-CF (solid line), DCA-CT (short-dashed line) and DCA-CFT (dashed line) error functions to their minimum. Compare with convergence using the ordinary-gradient version of the DCA-CF learning rule (dotted line).

the efficiency of the relative-gradient rules, we also plot the minimization of the DCA-CFT error using the ordinary-gradient version of the DCA-CF rule (87) (see Appendix B.3), which is as slow as the time-domain rule.

4 DCA-CS: Semi-Blind Separation

In the previous section we derived learning rules for the separating filters \mathbf{G}_m of convolutive mixtures. Those rules were made efficient by exploiting the relative-gradient concept. However, it is often advantageous to learn the mixing filters \mathbf{H}_m rather than \mathbf{G}_m , e.g., when the latter are much longer. As an example, consider a situation where the mixing includes a single time-point delay,

$$\mathbf{H}_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \mathbf{I}, \quad \mathbf{H}_1 = \begin{pmatrix} a & b \\ b & a \end{pmatrix}, \quad (40)$$

and $\mathbf{H}_m = \mathbf{0}$ for $m \geq 2$. In the frequency domain $\tilde{\mathbf{H}}_k = \mathbf{I} + e^{-i\omega_k} \mathbf{H}_1$. The separating filters are obtained using $\tilde{\mathbf{G}}_k = \tilde{\mathbf{H}}_k^{-1}$ and expanding in powers of \mathbf{H} :

$$\begin{aligned} \mathbf{G}_m &= \frac{1}{N} \sum_{k=0}^{N-1} e^{i\omega_k m} \sum_{n=0}^{\infty} e^{-i\omega_k n} (-\mathbf{H}_1)^n = \sum_{l=0}^{\infty} (-\mathbf{H}_1)^{m+Nl} \\ &= [\mathbf{I} - (-\mathbf{H}_1)^N]^{-1} (-\mathbf{H}_1)^m, \end{aligned} \quad (41)$$

where the second equality was obtained using $\sum_k e^{i\omega_k(m-n)}/N = \sum_l \delta_{n,m+Nl}$ and the last one using the identity $\sum_{l \geq 0} \mathbf{A}^l = (\mathbf{I} - \mathbf{A})^{-1}$, valid for any matrix \mathbf{A} as long as the sum converges. The latter condition depends on the eigenvalues of \mathbf{H}_1 , which are $\lambda_{\pm} = a \pm |b|$, and is satisfied when $|\lambda_{\pm}| < 1$. However, convergence guarantees the existence of causal and finite separating filters, but implies nothing about their length, which is determined by the eigenvalues of \mathbf{H}_1 . In particular, we notice from (41) that \mathbf{G}_m goes to zero at a rate that depends on the largest eigenvalue: $\mathbf{G}_m \sim \lambda_{max}^m$, which, if $|\lambda_{max}|$ is close to 1, would result in very long separating filters, even though the mixing filters are very short. In this case, learning \mathbf{H}_m requires a significantly smaller number of adaptive parameters.

Another advantage of learning \mathbf{H}_m is that, in some cases, it may be combined with learning the source filters \mathbf{h}_m in (31) and facilitate recovering the sources without whitening. In general, as discussed above, the formulation of the blind separation problem assumes that nothing is known about the mixing situation, except for the statistical independence of the sources. Hence we must allow an arbitrary mixing filter matrix \mathbf{H}_m in (30), and consequently the distinction between the source filters \mathbf{h}_m and the mixing filters is lost. However, in situations where some information is available about the mixing process, it can be incorporated into the learned mixing filters \mathbf{H}_m , enabling the learning of \mathbf{h}_m as well. We now derive the learning rules for such ‘semi-blind’ cases.

In the frequency domain, the description (31) gives

$$\tilde{\mathbf{y}}_k = \tilde{\mathbf{H}}_k \tilde{\mathbf{x}}_k, \quad \tilde{x}_{i,k} = \tilde{h}_{i,k} \tilde{u}_{i,k}, \quad (42)$$

producing a relation between the time-domain densities $p_y(y)$ and $p_u(u)$: $p_y(y) = \prod_k |\det \tilde{\mathbf{H}}_k|^{-1} \prod_{ik} |\tilde{h}_{i,k}|^{-1} \prod_{im} p_{i,m}(u_{i,m})$. The resulting DCA-CS (C=convolutive mixing, S=semi-blind separation) error function is therefore analogous to the hybrid DCA-CFT error (38) and is given by

$$E^{DCA-CS} = \frac{1}{N} \sum_{k=0}^{N-1} \log |\det \tilde{\mathbf{H}}_k| + \frac{1}{N} \sum_{k=0}^{N-1} \log |\tilde{h}_{i,k}| - \frac{1}{N} \sum_{m=0}^{N-1} \sum_{i=1}^L \log p_{i,m}, \quad (43)$$

where the term $p_{i,m}$ represents $p_{i,m} = \langle p(u_{i,m} = \sum_n g_{i,n} \sum_{jl} G_{ij,l} y_{j,m-n-l}) \rangle$ and the average is taken over the observed \mathbf{y}_m . We define \mathbf{G}_m and \mathbf{g}_m as the inverses of $\tilde{\mathbf{H}}_m$ and \mathbf{h}_m by

$$\tilde{\mathbf{G}}_k = \tilde{\mathbf{H}}_k^{-1}, \quad \tilde{g}_{i,k} = \frac{1}{\tilde{h}_{i,k}}. \quad (44)$$

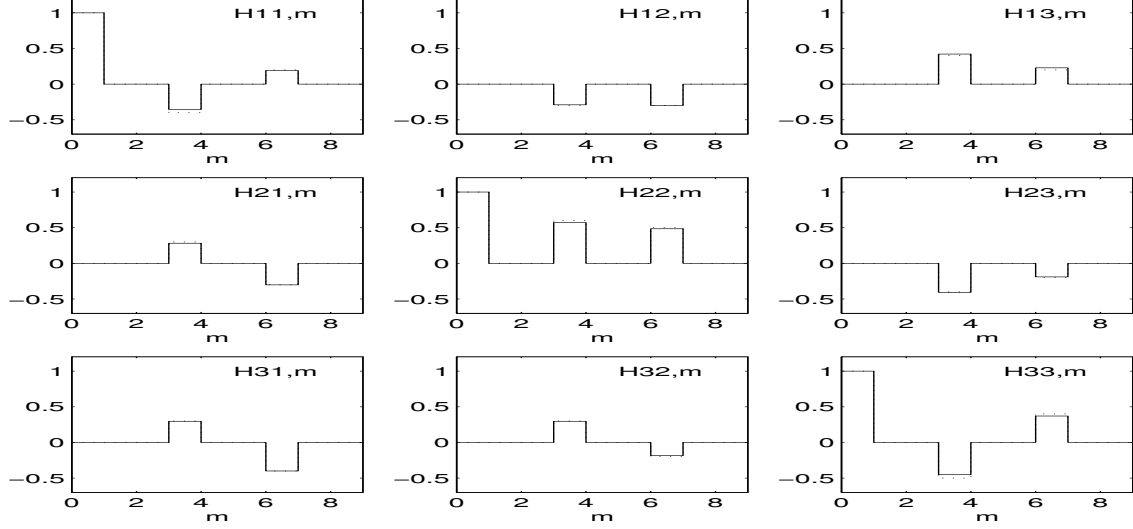


Figure 9: Impulse response of the filters learned by DCA-CS (solid line) compared with the original mixing filters (dotted line).

The DCA-CS learning rules are obtained from the ordinary gradient of (43) and are given here in terms of frequency-domain signals:

$$\begin{aligned}\delta\tilde{\mathbf{H}}_k &= -\epsilon\tilde{\mathbf{G}}_k^\dagger \left(\mathbf{I} - \mathbf{D}_k^{\tilde{g}^*} \tilde{\boldsymbol{\psi}}_k \tilde{\mathbf{x}}_k^\dagger \right), \\ \delta\tilde{h}_{i,k} &= -\epsilon\tilde{g}_{i,k}^* \left(1 - \tilde{\psi}_{i,k} \tilde{u}_{i,k}^* \right),\end{aligned}\quad (45)$$

where $D_k^{\tilde{g}}$ is the diagonal matrix containing the whitening filters (see (4)), and the modified whitened sources $\tilde{\boldsymbol{\psi}}_m$ are defined by (78). The rule for the source parameters ξ_i is the same as the one in (39). The time-domain increments are obtained by inverse DFT, e.g., $\delta\mathbf{H}_m = \sum_k e^{i\omega_k m} \delta\tilde{\mathbf{H}}_k / N$.

To derive the rules (45) we used the relations

$$\begin{aligned}\frac{\partial\tilde{G}_{ij,k}}{\partial H_{lp,k}} &= -\tilde{G}_{il,k} \tilde{G}_{pj,k}, \\ \frac{\partial u_{i,m}}{\partial H_{jl,n}} &= -\frac{1}{N} \sum_{k=0}^{N-1} e^{i\omega_k(m-n)} \tilde{g}_{i,k} \tilde{G}_{ij,k} \tilde{x}_{l,k}, \\ \frac{\partial u_{i,m}}{\partial h_{j,n}} &= -\frac{1}{N} \sum_{k=0}^{N-1} e^{i\omega_k(m-n)} \tilde{g}_{i,k}^2 \tilde{x}_{i,k} \delta_{ij},\end{aligned}\quad (46)$$

derived from (42,44).

Information available about the mixing process can now be incorporated into (45). For instance, if the mixing consists of a small number of echoes separated by known intervals, the appropriate elements of $H_{ij,m}$ are set to zero as (45) are being iterated. As another example, a useful application emerges in situations where a few-parameter description of the mixing process is available, based, e.g., on the physical properties of the propagation medium. In this case, the dependence of the mixing filters on the physical parameters, denoted $\boldsymbol{\alpha}$, is given in a functional form $\mathbf{H}_m = \mathbf{H}_m(\boldsymbol{\alpha})$. The learning rules for those parameters are deduced from (45): $\delta\boldsymbol{\alpha} = \sum_{ijm} \delta H_{ij,m} \partial H_{ij,m} / \partial \boldsymbol{\alpha}$.

Of course, the rules (45) can also be used to recover whitened sources, by learning only \mathbf{H}_m , setting $h_{i,m} = \delta_{m,0}$ and fixing $\delta h_{i,m} = 0$.

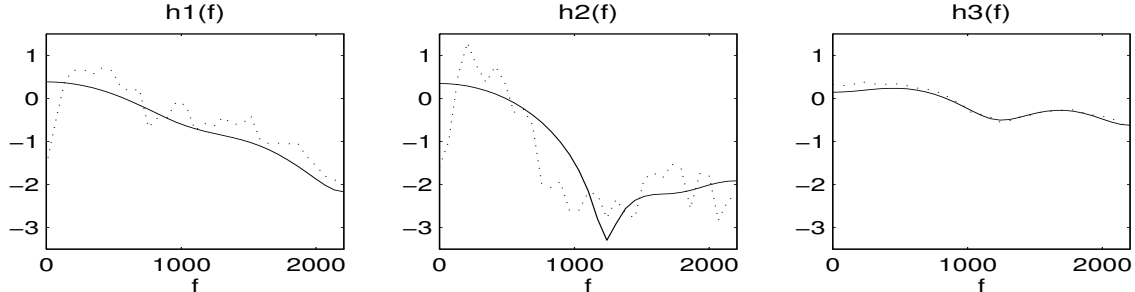


Figure 10: Spectra of the DCA-CS source filters (solid line) compared with the original source spectra (dotted line).

4.1 Results

Here we present a test of DCA-CS on a convolutive mixture of two speech signals and one colored Gaussian signal. The mixing filters had equally spaced taps with the non-zero elements of \mathbf{H}_m being

$$\mathbf{H}_0 = \mathbf{I}, \quad \mathbf{H}_3 = \begin{pmatrix} -0.4 & 0.3 & 0.3 \\ -0.3 & 0.6 & 0.3 \\ 0.4 & -0.4 & -0.5 \end{pmatrix}, \quad \mathbf{H}_6 = \begin{pmatrix} 0.2 & -0.3 & -0.4 \\ -0.3 & 0.5 & -0.2 \\ 0.2 & -0.2 & 0.4 \end{pmatrix}. \quad (47)$$

Thus, each mixture contained 2 or 3 delayed versions of each source, creating multiple echoes.

We applied the rules (45) to the resulting mixtures, simulating the situation in which information about the structure of the mixing filters (i.e., the tap spacing) is available by fixing the appropriate elements of the learned \mathbf{H}_m at zero. Source filters \mathbf{h}_m of length 5 were allowed. We worked in batch mode using $N = 512$ -point time blocks. Figure 9 shows that the algorithm learned the mixing filters successfully.

To illustrate the role of the source filters \mathbf{h}_m , we display their 64-point DFT spectra $|\tilde{\mathbf{h}}_k|^2$ in Figure 10, together with the source spectra $\langle |\tilde{\mathbf{x}}_k|^2 \rangle$, computed using 64-point DFT with overlapping windows. As discussed above, those filters model the source auto-correlations. More precise correspondence than shown in the figure requires longer \mathbf{h}_m , but the 5-point filters used here were sufficient to achieve separation in this situation.

It is interesting to compare DCA-CS with an algorithm that learns the separating filters, e.g., DCA-CFT, in a situation where the latter are much longer than the mixing filters, as discussed above (see (41)). We used two-point mixing \mathbf{H}_m of the form (40) with $a = .47$, $b = .50$, to mix two white signals $x_{1,m}, x_{2,m}$ with an exponential density $p(x_i) \propto e^{-|x_i|}$, sampled at each time point independently. The required length of \mathbf{G}_m is determined by the largest eigenvalue of \mathbf{H}_1 , $\lambda_+ = .97$, to be ~ 100 ; for illustration, on the left of Figure 11 we present $G_{21,m}$ learned by DCA-CFT (39). As shown on the right, DCA-CFT required more than three times as many iterations to learn \mathbf{G}_m as DCA-CS (used with $h_{i,m} = \delta_{m,0}$) required to learn \mathbf{H}_m . However, the superiority of DCA-CS in this situation notwithstanding, this result also underscores the efficiency of relative-gradient algorithms.

5 DCA-CR: Separation of rational mixtures

The algorithms presented in Section 3 for blind separation of convolutive mixtures are designed to learn the coefficients $G_{ij,m}$ of the separating filters, which then produce the (whitened) sources via $u_{i,m} = \sum_j (G_{ij} \star y_j)_m$. However, this approach becomes problematic in situations where the required separating filters are very long, increasing the number of adaptive parameters and possibly of undesired local minima of the error.

One approach to overcome this problem, described in Section 4, is to learn the coefficients of the mixing filters $H_{ij,m}$ rather than $G_{ij,m}$. This approach is advantageous when the mixing filters are significantly shorter, and has the additional benefit that information on the mixing process can be taken into account, facilitating the recovery of the sources unwhitened.

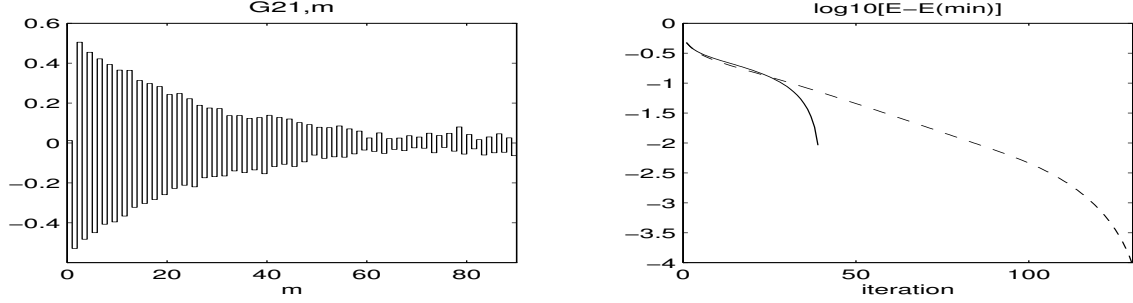


Figure 11: Left: separating filter $G_{21,m}$ learned by DCA-CFT for 2-point mixing. Right: convergence of the DCA-CS (solid line) and DCA-CFT (dashed line) error functions to their minimum.

Here we present a different approach to learning long separating filters, which is not restricted to short mixing situations. The key idea is to use a more general model for the separating filters $G_{ij,m}$. Its form is given in the frequency domain in terms of two filter matrices \mathbf{A}_m , \mathbf{B}_m by

$$\tilde{G}_{ij,k} = \frac{\tilde{B}_{ij,k}}{\tilde{A}_{ij,k}} = \frac{\sum_m e^{-i\omega_k m} B_{ij,m}}{\sum_m e^{-i\omega_k m} A_{ij,m}}, \quad (48)$$

and is termed ‘rational system function’ (Oppenheim and Schaffer 1989). The task is to learn \mathbf{A}_m and \mathbf{B}_m from the observed sensor signals \mathbf{y}_m . Notice that when $\mathbf{A}_m = \mathbf{I}\delta_{m,0}$, learning \mathbf{B}_m reduces to the learning of \mathbf{G}_m in the models of Section 3. However, $\mathbf{A}_m \neq 0$ can generate arbitrarily long filters even for a small number of $m \geq 1$.

Since we have $p_y(y) = \prod_k |\det \tilde{\mathbf{G}}_k| p_u(u)$, the DCA-CR (C=convolutive mixing, R=rational system function) error has the same form of the DCA-CFT error (38), but is now a function of \mathbf{A}_m and \mathbf{B}_m through (48):

$$E^{DCA-CR} = -\frac{1}{N} \sum_{k=0}^{N-1} \log |\det \tilde{\mathbf{G}}_k| - \frac{1}{N} \sum_{m=0}^{N-1} \sum_{i=1}^L \log p_{i,m}, \quad (49)$$

where the full expression for $p_{i,m}$ is given below (36).

The DCA-CR learning rules are obtained from the ordinary gradient of (49). Defining the frequency-domain $L \times L$ matrices

$$\tilde{\mathbf{C}}_k = \left(\mathbf{I} - \tilde{\psi}_k \tilde{\mathbf{u}}_k^\dagger \right) \tilde{\mathbf{H}}_k^\dagger, \quad (50)$$

with $\tilde{\mathbf{H}}_k = \tilde{\mathbf{G}}_k^{-1}$, we have

$$\delta \tilde{A}_{ij,k} = -\epsilon \tilde{G}_{ij,k}^* \frac{\tilde{C}_{ij,k}}{\tilde{A}_{ij,k}^*}, \quad \delta \tilde{B}_{ij,k} = \epsilon \frac{\tilde{C}_{ij,k}}{\tilde{A}_{ij,k}^*}, \quad (51)$$

where $\delta \mathbf{A}_m$ and $\delta \mathbf{B}_m$ follow by inverse DFT. The rule for the source parameters ξ_i is the same as the one in (39).

The rules (51) learn the separating filters \mathbf{G}_m (48). It is straightforward to derive analogous rules for the mixing \mathbf{H}_m , which may be convenient to use in some situations. With a rational system function parametrization

$$\tilde{H}_{ij,k} = \frac{\tilde{B}'_{ij,k}}{\tilde{A}'_{ij,k}} = \frac{\sum_m e^{-i\omega_k m} B'_{ij,m}}{\sum_m e^{-i\omega_k m} A'_{ij,m}}, \quad (52)$$

we have the learning rules

$$\delta \tilde{A}'_{ij,k} = -\epsilon \tilde{H}_{ij,k}^* \frac{\tilde{C}'_{ij,k}}{\tilde{A}'_{ij,k}}, \quad \delta \tilde{B}'_{ij,k} = \epsilon \frac{\tilde{C}'_{ij,k}}{\tilde{A}'_{ij,k}}, \quad (53)$$

where $\delta \mathbf{A}'_m$ and $\delta \mathbf{B}'_m$ follow by inverse DFT. The matrices $\tilde{\mathbf{C}}'_k$ in (53) are defined by

$$\tilde{\mathbf{C}}'_k = \tilde{\mathbf{G}}_k^\dagger \left(\mathbf{I} - \tilde{\boldsymbol{\psi}}_k \tilde{\mathbf{u}}_k^\dagger \right), \quad (54)$$

with $\tilde{\mathbf{G}}_k = \tilde{\mathbf{H}}_k^{-1}$ for $\tilde{\mathbf{H}}_k$ in (52) and should not be confused with (48). Note the symmetry between (50,51) and (54,53).

Whereas we saw before (Section 3) that the relative-gradient idea produces more efficient learning rules for the separating than the mixing filters, this is not the case when using the rational system function parametrization, hence the rules (53) are comparable to (51) in efficiency.

6 The Information Connection

Bell and Sejnowski's (1995) ICA algorithm was originally derived by maximizing the mutual information between the inputs and outputs of a linear network with sigmoidal output units. Specifically, the sensor signals \mathbf{y}_m are fed as inputs to a single-layer network with a $L \times L$ weight matrix \mathbf{G} that produces outputs $\mathbf{x}_m = \mathbf{G}\mathbf{y}_m$. Those are passed through a sigmoidal function, usually chosen to be $f(x) = 1/(1 + e^{-x})$, giving $z_{i,m} = f(x_{i,m}) + \eta_{i,m}$, where η_m are independent noises. The network weights are then optimized to maximize the mutual information between \mathbf{z}_m and \mathbf{x}_m in the zero-noise limit $\langle \eta_m^2 \rangle \rightarrow 0$ (assuming the noise has zero mean), resulting in independent signals \mathbf{x}_m which are the (scaled and permuted) original sources. In this limit, the mutual information becomes the output entropy. It was pointed out by Pearlmutter and Parra (1997) that maximizing the output entropy is equivalent to minimizing the Kullback-Leibler distance between the observed sensor density and a model sensor density, with the model parameter being the Bell and Sejnowski (1995) weight matrix \mathbf{G} and the corresponding source density being the derivative of their sigmoid $f(x)$.

In this section we show that the DCA algorithm is also equivalent to an information-maximization formulation involving a linear network with sigmoidal output units. However, here the network weights are dynamic, in the sense that they connect outputs at time t_m to inputs at the same but also previous times $t_n \leq t_m$, and the relevant quantity to maximize is the output entropy *rate* \mathcal{H}_{p_z} , which is a spatio-temporal (rather than spatial) quantity.

Assume we have a linear spatio-temporal network with weights \mathbf{G}_m , which receives the sensor signals \mathbf{y}_m as inputs and produces outputs $u_{i,m} = \sum_{jn} G_{ij,n} y_{i,m-n}$ (see (32)). Let \mathbf{u}_m be fed into a sigmoidal function f , producing $z_{i,m} = f_{i,m}(u_{i,m})$. We shall now consider the entropy of the outputs \mathbf{z}_m . However, unlike ICA which maximizes the joint entropy of all outputs at equal times, here we consider the joint entropy H_{p_z} of all outputs at all time points $\mathbf{z}_0, \dots, \mathbf{z}_{N-1}$. As $N \rightarrow \infty$, this quantity (divided by N) approaches the output entropy rate $\mathcal{H}_{p_z} = \lim_{N \rightarrow \infty} H_{p_z}/N$ (Cover and Thomas 1991).

Using the relation (98), we have

$$p_y^o(y) = \prod_{k=0}^{N-1} |\det \mathbf{G}_k| \prod_{i=1}^L \prod_{m=0}^{N-1} |f'_{i,m}(u_{i,m})| p_z(z), \quad (55)$$

where y stands for $\{\mathbf{y}_0, \dots, \mathbf{y}_{N-1}\}$ and $p_y^o(y)$ is the joint density of these $N \times L$ observed variables; the same holds for z . The entropy rate is then given by

$$\mathcal{H}_{p_z} = -\frac{1}{N} \int dz p_z(z) \log p_z(z)$$

$$\begin{aligned}
&= -\frac{1}{N} \int dy p_y^o(y) \left[\log p_y^o(y) - \sum_{k=0}^{N-1} \log |\det \mathbf{G}_k| - \sum_{i=1}^L \sum_{m=0}^{N-1} \log |f'_{i,m}(u_{i,m})| \right] \\
&= \mathcal{H}_{p_y^o} + \frac{1}{N} \sum_{k=0}^{N-1} \log |\det \mathbf{G}_k| + \frac{1}{N} \sum_{i=1}^L \sum_{m=0}^{N-1} \langle \log |f'_{i,m}(u_{i,m})| \rangle, \tag{56}
\end{aligned}$$

where \mathbf{u}_m are related by \mathbf{G}_m to the observed \mathbf{y}_m and the average is taken over the latter.

We now point out that $\mathcal{H}_{p_z} = \mathcal{H}_{p_y^o} - E^{DCA-CFT}$ (see (38,65)) if we identify $|f'_{i,m}|$ with the whitened source density $p_{i,m}$. Since $\mathcal{H}_{p_y^o}$ is parameter-independent, minimizing the DCA-CFT error function is equivalent to maximizing the output entropy rate of our linear-sigmoidal network. Similar proofs can be provided for all the other DCA algorithms, both for the instantaneous and convolutive cases, presented in this paper.

7 More sensors than sources

Blind separation algorithms, including those presented here so far, address the square problem where the number of sensor signals L' equals the number of sources L . Unlike the non-square $L' < L$ problem, which requires a conceptually different approach, the non-square $L' > L$ problem ought to be solvable using the optimization method underlying DCA. However, our derivation of the DCA learning rules was given for square $L \times L$ matrices and cannot formally be extended to the non-square case. Alternatively, one can apply them to the observed L' sensors and seek a $L' \times L'$ separating matrix. However, the algorithms may attempt to produce L' recovered sources by splitting the L original sources apart.

There is a simple resolution to this problem. Focusing first on the instantaneous case, it relies of the observation that L' linear mixtures of $L < L'$ signals cannot be linearly independent, and that it is possible to extract from them L mixtures that *are* linearly independent. Of course, this assumes that the mixing matrix is of rank $r = L$; in general $r \leq L$, but the case $r < L$ corresponds to a situation with effectively less mixtures than sources and lies outside the scope of this paper.

To see how this can be done, assume $\mathbf{y}_m = \mathbf{H}\mathbf{x}_m$ for a $L' \times L$ matrix \mathbf{H} , and consider the sensor correlation matrix $\mathbf{C} = \langle \mathbf{y}_m \mathbf{y}_m^T \rangle$. This $L' \times L'$ matrix has L positive eigenvalues and the rest vanish. Specifically, we can write

$$\mathbf{P}^T \mathbf{C} \mathbf{P} = \mathbf{\Lambda}, \tag{57}$$

where \mathbf{P} is a real orthogonal matrix ($\mathbf{P}^{-1} = \mathbf{P}^T$) containing the eigenvectors of \mathbf{C} , and $\mathbf{\Lambda}$ is a diagonal matrix containing its eigenvalues. We now order the columns of \mathbf{P} such that the L positive eigenvalues are Λ_{ii} for $i = 1, \dots, L$. Next, we consider the principal components (PC's) of the sensor signals,

$$\mathbf{y}'_m = \mathbf{\Lambda}^{-1/2} \mathbf{P}^T \mathbf{y}_m, \tag{58}$$

where we define $(\mathbf{\Lambda}^{-1/2})_{ii} = 0$ if $\Lambda_{ii} = 0$, and observe that only the first L of them are non-zero, since $\langle y_{i,m}^2 \rangle = 0$ for $i = L + 1, \dots, L'$. Those first L PC's satisfy $\langle y_{i,m} y_{j,m} \rangle = \delta_{ij}$, hence are linearly independent and constitute appropriate inputs for instantaneous DCA algorithms. We point out that this method is advantageous to simply picking L sensor signals as inputs, since those may not be linearly independent, resulting in a situation with effectively less mixtures than sources.

The convolutive mixing case can be treated similarly. However, here the use of PCA is not sufficient, since the sensor equal-time correlation matrix \mathbf{C} may be of rank $r > L$ even if $L' > L$, due to the contribution of delayed versions of the source signals to the mixtures. Hence, one should consider the cross-correlation matrix $\mathbf{C}_m = \langle \sum_n \mathbf{y}_n \mathbf{y}_{n-m}^T \rangle$. Equivalently, we consider its DFT, the cross-spectrum matrix $\tilde{\mathbf{C}}_k = \langle \tilde{\mathbf{y}}_k \tilde{\mathbf{y}}_k^\dagger \rangle$, which measures the correlation between the sensors in the frequency domain. For each frequency ω_k , this matrix has at most L positive eigenvalues and the rest vanish. In analogy with the instantaneous case above, we can write

$$\tilde{\mathbf{P}}_k^\dagger \tilde{\mathbf{C}}_k \tilde{\mathbf{P}}_k = \tilde{\mathbf{\Lambda}}_k, \tag{59}$$

where $\tilde{\mathbf{P}}_k$ is a complex unitary matrix ($\tilde{\mathbf{P}}_k^{-1} = \tilde{\mathbf{P}}_k^\dagger$) containing the eigenvectors of $\tilde{\mathbf{C}}_k$, and $\tilde{\mathbf{\Lambda}}_k$ is a diagonal matrix containing its eigenvalues. Next, we consider the PC's of the frequency-domain sensor signals at ω_k ,

$$\tilde{\mathbf{y}}'_k = \tilde{\mathbf{\Lambda}}_k^{-1/2} \tilde{\mathbf{P}}_k^\dagger \tilde{\mathbf{y}}_k, \quad (60)$$

where again we define $(\tilde{\mathbf{\Lambda}}_k^{-1/2})_{ii} = 0$ if $\tilde{\Lambda}_{ii,k} = 0$, and observe, as above, that only the first L of them are non-zero. Going back to the time domain, those L new signals $y'_{i,m} = \sum_j (P'_{ji} \times y_j)_m$, $i = 1, \dots, L$ (where \mathbf{P}'_m is the inverse DFT of $\tilde{\mathbf{\Lambda}}_k^{-1/2} \tilde{\mathbf{P}}_k$) are convolutive mixtures of the sensor signals and form suitable inputs to DCA algorithms.

Note that not only the equal-time correlations of \mathbf{y}'_m vanish, but also their cross-correlations: $\langle y'_{i,m} y'_{j,n} \rangle = \delta_{ij} \delta_{mn}$. Hence, the filter matrix \mathbf{P}'_m performs decorrelation in space and time simultaneously, generalizing the ordinary PCA transformation (57) which performs only spatial decorrelation.

In practice, the eigenvalues of the correlation and cross-spectrum matrices seldom actually vanish, due to the presence of noise and finite machine precision. Therefore, as is the case when using SVD, a cut-off has to be determined, based, e.g., on the known noise level, and eigenvalues below it are taken to be zero.

Finally, we point out that the spatio-temporal dimensional reduction procedure, described above in the frequency domain, can also be performed in the time domain. To do this we return to the cross-correlations and define the $L'N \times L'N$ matrix $\bar{C}_{(im)(jn)} = \langle y_{i,m} y_{j,n} \rangle$. This matrix has a Toeplitz structure since $\bar{C}_{(im)(jn)} = C_{ij,m-n}/N$, with \mathbf{C}_m being related to the above cross-spectrum matrix via DFT. It can be diagonalized by an orthogonal $L'N \times L'N$ matrix $\bar{\mathbf{P}}_{(im)(jn)}$ to give $\bar{\mathbf{P}}^T \mathbf{C} \bar{\mathbf{P}} = \bar{\mathbf{\Lambda}}$, ordering the columns of $\bar{\mathbf{P}}$ such that all the positive eigenvalues $\bar{\Lambda}_{(im)(im)}$ are contained in the first LN elements of $\bar{\mathbf{\Lambda}}$. The spatio-temporally decorrelating filter \mathbf{P}'_m above is then obtained by considering only the first LN columns of $\bar{\mathbf{P}}$ and exploiting its Toeplitz structure $\bar{P}_{(im)(jn)} = P_{ij,m-n}/N$, as well as normalizing by $(\bar{\mathbf{\Lambda}}^{-1/2})_{(im)(im)}$.

8 Discussion and Conclusion

In this paper we presented the DCA algorithm for separating instantaneous and convolutive mixtures of independent sources by learning a separating transformation in an unsupervised manner from the sensor high-order spatio-temporal statistics. The DCA approach is based on a generative model of the sensor density in either the time or frequency domains, with the model parameters describing the separating filter matrix, as well as the source densities and auto-correlations.

The DCA-I algorithms for instantaneous mixing (Section 2) are more powerful than ICA methods, as has been demonstrated by two examples. In both cases, it was the use of spatio-temporal statistics facilitated by introducing the whitening filters \mathbf{g}_m that gave DCA-I an advantage. Note that in the case of uniformly distributed sources, ICA could have achieved separation by carefully taking into account the source distribution, whereas for colored Gaussian sources, information on the source auto-correlations obtained by learning \mathbf{g}_m is crucial.

The DCA-C algorithms for convolutive mixing (Section 3) are a direct generalization of DCA-I, obtained by replacing spatio-temporal generative model parametrized by filters \mathbf{g}_m and matrix \mathbf{G} by a matrix of filters \mathbf{G}_m . Time-, frequency-, and hybrid frequency/time-domain error functions were derived for both DCA-I and DCA-C, resulting in different learning rules. The time-domain rules were the slowest learners and the other two were almost comparable, with the hybrid rules being the fastest in both the instantaneous and convolutive cases. The relative-gradient concept, extended from the spatial to the spatio-temporal case and incorporated into DCA in this paper (Appendices A.1 and B.1), is credited with this performance. Note that the frequency-domain DCA-IF,CF and hybrid DCA-IFT,CFT errors include the whitening/separating filters in the frequency domain; it is due to this feature that they benefit from the convolutive relative gradient approach. This improvement is seen particularly clearly by comparing the performance of the relative-gradient DCA-CF rule (35) to its ordinary-gradient version (87) in Figure 8. We point out that no optimization of the error minimization process was attempted; the utility of standard methods like conjugate gradients and Newton's method will be assessed in a subsequent study.

Whereas exploiting the relative gradient accelerates convergence of some DCA rules by minimizing both the number of iterations and of floating-point operations, the use of FFT accelerates all rules equally by minimizing the number of floating-point operations required to compute the cross-correlations, which form a necessary ingredient of all DCA algorithms. The rules in the body of the paper are usually given in terms of time-domain quantities; their FFT-computable versions appear in the appendices. Of course, any learning rule derived from either F-, T-, or FT-type error function can be written using either time- or frequency-domain quantities.

In the frequency domain the convolutive mixing problem factorizes (33), and it appears that we are faced with a separate instantaneous-mixing problem at each frequency with a complex mixing matrix. This may lead one to expect two difficulties. First, frequency-domain signals are obtained by filtering time-domain ones; based on the central limit theorem, this summation over M time points could produce Gaussian frequency-domain densities. However, most naturally-occurring signals (e.g., speech, music) are non-Gaussian and have long-range temporal correlations (see Attias and Schreiner 1997), resulting in their frequency components being also non-Gaussian; in fact, their density is well approximated by the exponential family. This does not contradict the central limit theorem since the latter assumes that we sum over temporally independent variables. Second, since the mixing problem is defined only to within a source permutation, we could be facing an arbitrary permutation at each frequency, necessitating the design of clever schemes for putting the separated signals together by, e.g., exploiting the correlation between frequency components of a given source. However, DCA avoids this difficulty since the mixing problems at different frequencies are not mutually independent, but are solved simultaneously by minimizing an error that couples all of them. This is manifested by the non-invariance of the DCA-C errors under such permutations, as discussed in Section 3.4.

Since working in the frequency domain is a crucial feature of the DCA approach, it is important to make the following comment. A cross-correlation, e.g., $(\phi_i \times u_j)_m$ in (35), equals its DFT version $\sum_k e^{i\omega_k m} \tilde{\phi}_{i,k}^* \tilde{u}_{j,k} / N$ only for periodic signals. Thus, computing the time-domain increments $\delta \mathbf{G}_m$ using the DFT version (90) is an approximation, as can be seen from

$$\begin{aligned} (\phi_i \times u_j)_m &= \sum_{n=0}^{N-1-m} \phi_{i,n} u_{j,n+m} = \frac{1}{N} \sum_{kl=0}^{N-1} e^{i\omega_l m} \tilde{\phi}_{i,k}^* \tilde{u}_{j,l} \frac{1}{N} \sum_{n=0}^{N-1-m} e^{i(\omega_k + \omega_l)n} \\ &= \frac{1}{N} \sum_{k=0}^{N-1} \tilde{\phi}_{i,k}^* \tilde{u}_{j,k} - \sum_{kl=0}^{N-1} e^{i\omega_l m} \tilde{\phi}_{i,k}^* \tilde{u}_{j,l} \left[\frac{1}{N} \frac{1 - e^{i(\omega_k + \omega_l)m}}{1 - e^{i(\omega_k + \omega_l)}} \right], \end{aligned} \quad (61)$$

obtained using (3) and $\sum_n e^{i(\omega_k + \omega_l)n} / N = \delta_{k,-l}$. The second term on the last line of (61) is a correction to the DFT version (first term there). However, since the term in the square brackets is of the order of m/N , this correction is negligible for $m \ll N$. Next, note from (90) that m is limited by the length M of \mathbf{G}_m , thus our approximation is valid if we keep $M \ll N$ so the separating filters much shorter than the time blocks. In fact, the root of this approximation is in the transition from the time- (11,32) to frequency-domain (12,33) formulations.

The other reason to keep $M \ll N$ is our very use of N -block processing, rather than processing the full signals which can be arbitrarily long. The filtering $\mathbf{x}_m = \sum_n \mathbf{G}_n \mathbf{y}_{m-n}$ (32) for $0 \leq m < M - 1$ thus approximates the \mathbf{y}_m preceding the current block as zero. Again, for cross-correlations computed using the resulting \mathbf{x}_m , this approximation becomes valid when $M \ll N$. In practice we found that $M/N \sim 1/6$ was usually sufficiently small.

Note that $M \ll N$ is actually a condition on the time-block length, since we must choose the separating filters to be sufficiently long to invert the mixing, hence M is determined by the problem at hand. This condition can be justified intuitively by observing that, in order to recognize \mathbf{y}_m as a delayed (shifted) version of \mathbf{x}_m , the shifting involved must be short compared to the signal lengths.

We point out the the stability of DCA is ensured by its formulation as an optimization problem using the KL distance, which is bounded from below (Cover and Thomas 1991), as an error function. Previously

proposed separation methods, such as the original H-J network (Jutten and Herault 1991) and its extension to convolutive mixing by Platt and Faggin (1992), are not derived from an error function, except for special choices of the non-linearities (Comon, Jutten and Herault 1991; Sorouchyari 1991; Comon 1994), and indeed exhibit occasional unstable behavior.

Although the formulation of DCA in terms of learning a generative model assumes that the model approximates well the situation at hand, we found that in practice separation can be achieved even when this condition is relaxed. In particular, the use of whitening filters too short to produce perfect whitening (see, e.g., Figure 2) and non-adaptive model source densities that differ from the actual ones (e.g., using the sigmoid-derivative form for separating speech signals whose density is nearly exponential) did not lead to noticeable degradation in separation quality. Nevertheless, we do expect performance to degrade when the approximation provided by the model is ‘sufficiently far’ from the actual situation; it is well known, e.g., that the use of sigmoid-derivative form for the model source density fails when the sources have negative kurtosis (bell and Sejnowski 1995). The analysis needed to determine the necessary and sufficient conditions for the model to achieve separation is quite difficult and lies beyond the scope of this paper. A brief sketch of such an analysis for ICA was outlined by Cardoso (1997).

Separation of instantaneous mixtures using spatio-temporal statistics of the *second* order was suggested by Molgedey and Schuster (1994) and Belouchrani et al. (1997), and of higher orders by Pearlmutter and Parra (1997). None of these methods has a natural generalization to the convolutive case. Note that DCA-I can be made to exploit only second-order statistics by using Gaussian model sources: $P \propto e^{-|\tilde{u}_{i,k}|^2}$ in (15) and $p \propto e^{-u_{i,m}^2}$ in (19). In this way it essentially uses the full auto-correlations of the sources, rather than their values only at a given time lag as do Molgedey and Schuster (1994). However, one must note that any algorithm which uses only second-order statistics will be completely unable to separate convolutive mixtures, unless strong constraints are imposed on the mixing filters. For instantaneous mixtures it may generally be effective, but will fail to separate sources whose auto-correlations are identical; thus, for example, mixtures of white sources can be separated only by exploiting higher orders as in DCA-I.

Torkkola (1996) proposed the ordinary-gradient rule corresponding to the frequency-domain version of (39), and its relative-gradient form was described by Lee, Bell and Lambert (1997) (see also Lambert 1996); in the absence of a spatio-temporal/spectral error function, both relied on information-maximization considerations in the frequency domain. A rule similar to (39) appeared in (Cochocki et al. 1996). Methods that use cumulant information in the frequency domain (i.e., polyspectra) were suggested (Thi and Jutten 1995; Yellin and Weinstein 1995) but are restricted to $L = 2$ sources. Comon (1996) suggested a polyspectra-based optimization formulation but did not provide a separation algorithm.

DCA is sufficiently flexible to allow a formulation in which information on the mixing process can be exploited when available. As discussed in Sections 3.4 and 4, in the absence of such information, the sources can be recovered only with their spectra modified, whereas incorporating such information when available (the semi-blind case) facilitates recovering the sources unwhitened. As discussed in Section 4, this requires learning the mixing, rather than the separating, filters, using ordinary-gradient rules. Quite general forms of filters, useful since they can model very long filters using relatively few parameters, can also be learned by DCA (Section 5).

All the DCA algorithms can be viewed as simple networks with two output layers (the whitened sources and modified whitened sources in DCA-C; three layers in other versions), sensor signals as inputs and separating (or mixing) filters as weights. The weight increments are determined by cross-correlating the different outputs across layers and with the weights. Convergence is achieved when the cross-correlations between different outputs vanish, producing separation. Note that the cross-correlated signals are whitened sources and a non-linear modification thereof, hence high-order sensor statistics are used to achieve separation. This network was shown in Section 6 to maximize the *information rate* between its inputs and outputs.

Algorithms that solve the problem of blind source separation address, in fact, the more general need for an efficient tool for statistical analysis of spatio-temporal data sets, e.g., biomedical multi-sensor recordings such as EEG (Makeig et al. 1996, 1997) and MEG (Poeppl et al. 1997). The DCA separating filters produce simultaneous spatial and temporal redundancy reduction. Combined with dimensionality reduction as described in Section 7, the resulting dynamic components of a given data set may have a natural interpretation

in terms of its generating mechanisms.

These algorithms may also shed light on the methods employed by the nervous system to process data from its various receptor arrays, in accord with Barlow's (1989) suggestion of redundancy reduction as an important goal of sensory processing. This idea has been formulated as a quantitative theory for the sub-cortical visual system in (Atick and Redlich 1990; Atick 1992; Dong and Atick 1995). In (Bell and Sejnowski 1996), an attempt is made to construct a computational model of primary visual cortex cells based on ICA. Hopfield (1991) applied Jutten and Herault's (1991) ideas to explain odor discrimination. Furthermore, the equivalent information-rate maximization of DCA given in Section 6 connects this approach to a recent line of physiological experiments (see, e.g., Bialek et al. 1991; Rieke et al. 1997; Attias and Schreiner 1998) that suggest that the nervous system is designed to maximize the rate at which spike trains carry information about the stimuli. The results of the present paper motivate us to hypothesize that the nervous system performs dynamic component analysis of its inputs from various sensory modalities.

Future work will explore the consequences of this hypothesis for spatio-temporal characteristics of neural filter properties. On the computational front, progress in designing efficient methods to separate noisy, time-dependent and non-square mixtures will be necessary in order to mount a fresh attack on the notorious cocktail-party separation problem (Bregman 1990), which is routinely and successfully confronted by the human brain.

A Instantaneous mixing: Error functions and learning rules

A.1 DCA-IF

Here we provide a somewhat detailed derivation of the DCA-IF error function (14) and learning rules (16). The relation between the whitened sources $\tilde{\mathbf{u}}_k$, sources $\tilde{\mathbf{x}}_k$, and sensors $\tilde{\mathbf{y}}_k$, given by $\tilde{u}_{i,k} = \tilde{g}_{i,k}\tilde{x}_{i,k} = \tilde{g}_{i,k}\sum_j G_{ij}\tilde{y}_{j,k}$ (12), leads to the following relation between their model densities:

$$p_{\tilde{\mathbf{y}}}(\tilde{\mathbf{y}}) = |\det \mathbf{G}|^N p_{\tilde{\mathbf{x}}}(\tilde{\mathbf{x}}) = |\det \mathbf{G}|^N \prod_{i=1}^L \prod_{k=0}^{N-1} |\tilde{g}_{i,k}| \prod_{k=0}^{N/2} P_{i,k}(\tilde{u}_{i,k}), \quad (62)$$

where (13) has been used. The symbol $\tilde{\mathbf{y}}$ stands for $\{\tilde{\mathbf{y}}_0, \dots, \tilde{\mathbf{y}}_{N/2}\}$, and $p_{\tilde{\mathbf{y}}}(\tilde{\mathbf{y}})$ is the joint density of these $N \times L$ variables (note from the DFT definition (3) that $\text{Im}\tilde{\mathbf{y}}_0 = \text{Im}\tilde{\mathbf{y}}_{N/2} = 0$); similarly for $\tilde{\mathbf{x}}$. The subscript k in the last product runs from 0 only to $N/2$ since, from (3), the frequency components satisfy $\tilde{\mathbf{u}}_{N-k} = \tilde{\mathbf{u}}_k^*$. Note from (62) that the model sensor signals \mathbf{y}_m are generally not white and their density is not factorial.

The relation (62) is a corollary of the transformation rule of a probability density under a linear transformation of variables, with some care taken due to both the transformation and the variables being complex. The first equality is obtained by noting that for each $k \neq 0, N/2$, \mathbf{G} operates on the real and imaginary parts of $\tilde{\mathbf{y}}_k = \tilde{\mathbf{y}}'_k + i\tilde{\mathbf{y}}''_k$ separately, and thus $\tilde{\mathbf{x}}'_k = \mathbf{G}\tilde{\mathbf{y}}'_k$ and $\tilde{\mathbf{x}}''_k = \mathbf{G}\tilde{\mathbf{y}}''_k$ each contribute $|\det \mathbf{G}|$ to the transformation $p_{\tilde{\mathbf{y}}} \rightarrow p_{\tilde{\mathbf{x}}}$; for $k = 0, N/2$ only the real parts contribute. The second equality follows from writing $\tilde{u}_{i,k} = \tilde{g}_{i,k}\tilde{x}_{i,k}$ in a matrix form,

$$\begin{pmatrix} \tilde{u}'_{i,k} \\ \tilde{u}''_{i,k} \end{pmatrix} = \begin{pmatrix} \tilde{g}'_{i,k} & -\tilde{g}''_{i,k} \\ \tilde{g}''_{i,k} & \tilde{g}'_{i,k} \end{pmatrix} \begin{pmatrix} \tilde{x}'_{i,k} \\ \tilde{x}''_{i,k} \end{pmatrix}, \quad (63)$$

which shows that for each i and $k \neq 0, N/2$, this change of variables contributes $|\tilde{g}_{i,k}|^2$, the determinant of the matrix in (63), to the transformation $p_{\tilde{\mathbf{x}}} \rightarrow p_{\tilde{\mathbf{u}}}$; for $k = 0, N/2$ the contribution is just $|\tilde{g}_{i,k}|$. Finally, to extend the sum up to $k = N - 1$ we use $\tilde{g}_{i,N-k} = \tilde{g}_{i,k}^*$.

The separation parameters, which consist of the separating matrix \mathbf{G} , whitening filters \mathbf{g}_m , and source density parameters ξ_i , should now be optimized to minimize the distance between the model $p_{\tilde{\mathbf{y}}}$ and the observed $p_{\tilde{\mathbf{y}}}^o$ sensor densities. To do this we choose our error function to be the Kullback-Leibler distance,

which is bounded from below (Cover and Thomas 1991) and is given by

$$\begin{aligned}
E(p_{\tilde{y}}^o, p_{\tilde{y}}) &= \frac{1}{N} \int d\tilde{y} p_{\tilde{y}}^o(\tilde{y}) \log \frac{p_{\tilde{y}}^o(\tilde{y})}{p_{\tilde{y}}(\tilde{y})} \\
&= \frac{1}{N} \int d\tilde{y} p_{\tilde{y}}^o(\tilde{y}) \log p_{\tilde{y}}^o(\tilde{y}) - \frac{1}{N} \int d\tilde{y} p_{\tilde{y}}^o(\tilde{y}) \log p_{\tilde{y}}(\tilde{y}) \\
&= -\frac{1}{N} H_{p_{\tilde{y}}^o} - \frac{1}{N} \langle \log p_{\tilde{y}}(\tilde{y}) \rangle_{p_{\tilde{y}}^o}.
\end{aligned} \tag{64}$$

We divide by N in (64) to prevent divergence as $N \rightarrow \infty$. Both terms on the last line have simple interpretations. $H_{p_{\tilde{y}}^o}$ is the entropy of the observed frequency-domain sensor signals $\{\tilde{\mathbf{y}}_0, \dots, \tilde{\mathbf{y}}_{N/2}\}$. For stationary signals, the limit $\lim_{N \rightarrow \infty} H_{p_{\tilde{y}}^o}/N = \mathcal{H}_{p_{\tilde{y}}^o}$ exists and is termed the ‘entropy rate’ of the sensors. The second term is the log-likelihood of the data given our model. Since the sensor entropy rate is independent of the separation parameters, minimizing the KL distance is equivalent to maximizing the log-likelihood of the data with respect to those parameters. Substituting (62) in (64) and dropping terms independent of the separation parameters, we obtain the error function (14). Of course, in the time domain we can similarly obtain

$$E(p_y^o, p_y) = -\frac{1}{N} H_{p_y^o} - \frac{1}{N} \langle \log p_y(y) \rangle_{p_y^o}, \tag{65}$$

where $y = \{\mathbf{y}_0, \dots, \mathbf{y}_{N-1}\}$ and p_y is the joint spatio-temporal density of the sensors.

A.1.1 Ordinary gradient descent

In order to derive the learning rules we must differentiate E with respect to the separation parameters. This is facilitated by the following identities:

$$\begin{aligned}
\frac{\partial \log |\det \mathbf{G}|}{\partial G_{ij}} &= (G^{-1})_{ji}, \\
\frac{\partial \tilde{u}_{i,k}}{\partial G_{jl}} &= \tilde{g}_{j,k} \tilde{y}_{l,k} \delta_{ij}, \\
\frac{\partial \tilde{u}_{i,k}}{\partial g_{j,m}} &= e^{-i\omega_k m} \tilde{x}_{j,k} \delta_{ij}, \\
\frac{\partial \log |\tilde{g}_{i,k}|}{\partial g_{j,m}} &= \operatorname{Re} \left(\frac{e^{i\omega_k m}}{\tilde{g}_{j,k}^*} \right) \delta_{ij},
\end{aligned} \tag{66}$$

where \star denotes complex conjugation. The first identity in (66) can be derived using $\log |\det \mathbf{G}| = \frac{1}{2} \log \det \mathbf{G}^2 = \frac{1}{2} \operatorname{Tr} \log \mathbf{G}^2$ and the eigenvalue decomposition of \mathbf{G} . The second and third identities result from (12) and the last one from the DFT definition (3).

The gradient-descent learning rules can now be obtained using the chain rule together with (66):

$$\begin{aligned}
\delta G_{ij} &= -\epsilon \frac{\partial E}{\partial G_{ij}} = \epsilon (\mathbf{G}^{-1})_{ij} - \epsilon \frac{1}{N} \sum_{k=0}^{N-1} \tilde{g}_{i,k}^* \tilde{\phi}_{i,k} \tilde{y}_{j,k}^*, \\
\delta g_{i,m} &= -\epsilon \frac{\partial E}{\partial g_{i,m}} = \epsilon \frac{1}{N} \sum_{k=0}^{N-1} e^{i\omega_k m} \left(\frac{1}{\tilde{g}_{i,k}^*} - \tilde{\phi}_{i,k} \tilde{x}_{i,k}^* \right), \\
\delta \xi_i &= -\epsilon \frac{\partial E}{\partial \xi_i} = \epsilon \frac{1}{N} \sum_{k=0}^{N/2} \frac{\partial \log P_{i,k}}{\partial \xi_i}.
\end{aligned} \tag{67}$$

The first two rules in (67) make use of the modified whitened sources $\tilde{\phi}_k$, which are defined in the frequency domain in terms of $\tilde{\mathbf{u}}_k$ and their density $P_{i,k}(\tilde{u}_{i,k})$ (see (15)) by

$$\tilde{\phi}_{i,k} = -\alpha_k \frac{\partial P_{i,k}}{\tilde{u}'_{i,k}} - i\alpha_k \frac{\partial P_{i,k}}{\tilde{u}''_{i,k}}, \quad \alpha_k = \frac{1}{2}(1 + \delta_{k,0} + \delta_{k,N/2}), \quad (68)$$

where $\tilde{u}_{i,k} = \tilde{u}'_{i,k} + i\tilde{u}''_{i,k}$. The α_k appear essentially to compensate for $\tilde{u}''_{i,k} = 0$ at $k = 0, N/2$.

The learning rules (67) can also be given in terms of time-domain signals, by exploiting their inverse DFT form (i.e., a time-domain quantity given by a sum over all frequency components $k = 0, \dots, N-1$, see (3)):

$$\begin{aligned} \delta \mathbf{G} &= \epsilon (\mathbf{G}^T)^{-1} - \epsilon \sum_m \mathbf{D}_m^g (\phi \times \mathbf{y}^T)_{-m}, \\ \delta g_{i,m} &= \epsilon g'_{i,m} - \epsilon (\phi_i \times x_i)_{-m}, \end{aligned} \quad (69)$$

where the signals $\phi_{i,m}$ and filters $g'_{i,m}$ are the time-domain counterparts of $\tilde{\phi}_{i,k}$ (68) and $\tilde{g}'_{i,k} = 1/\tilde{g}_{i,k}^*$, and \mathbf{D}_m^g is a diagonal matrix containing \mathbf{g}_m , as defined in (4). In component notation, the rule for \mathbf{G} is given by $\delta G_{ij} = \epsilon (\mathbf{G}^{-1})_{ji} - \epsilon \sum_{mn} g_{i,m} \phi_{i,n} y_{j,n-m}$.

A.1.2 Relative gradient descent

The learning rules for \mathbf{G} in (67,69) are expensive since they require matrix inversion at each iteration. This can be avoided by using rules based on the relative gradient of the error function, rather than the ordinary gradient. The concept of the relative gradient was first introduced in (Cardoso and Laheld 1996; Amari et al. 1996) in the context of algorithms for separating instantaneous mixtures using equal-time statistics, and is incorporated into DCA in the following.

We first define the relative gradient of E with respect to \mathbf{G} , denoted $\nabla_{\mathbf{G}} E$, in terms of the ordinary gradient $\partial E / \partial \mathbf{G}$, by

$$\nabla_{\mathbf{G}} E = \frac{\partial E}{\partial \mathbf{G}} \mathbf{G}^T \mathbf{G}. \quad (70)$$

In the previous section we used the well-known fact that incrementing \mathbf{G} in each iteration by $\delta \mathbf{G} = -\epsilon \partial E / \partial \mathbf{G}$ produces a non-positive change in E , as long as ϵ is sufficiently small, to derive ordinary-gradient learning rules for \mathbf{G} . Next we show that incrementing \mathbf{G} by $\delta^R \mathbf{G} = -\epsilon \nabla_{\mathbf{G}} E$ has precisely the same effect. Indeed,

$$\begin{aligned} \delta E &= E(\mathbf{G} + \delta^R \mathbf{G}) - E(\mathbf{G}) = \text{Tr} \frac{\partial E}{\partial \mathbf{G}} \delta^R \mathbf{G}^T = -\epsilon \text{Tr} \frac{\partial E}{\partial \mathbf{G}} (\nabla_{\mathbf{G}} E)^T \\ &= -\epsilon \text{Tr} \left(\frac{\partial E}{\partial \mathbf{G}} \mathbf{G}^T \right) \left(\frac{\partial E}{\partial \mathbf{G}} \mathbf{G}^T \right)^T \leq 0. \end{aligned} \quad (71)$$

To prove the second equality we use component notation to write $E(\mathbf{G} + \delta^R \mathbf{G}) = E(\mathbf{G}) + \sum_{ij} \partial E / \partial G_{ij} \delta^R G_{ij}$, which is true to order ϵ^2 . The fourth equality is obtained using (70). The last inequality can be verified by noting that any matrix \mathbf{A} satisfies $\text{Tr} \mathbf{A} \mathbf{A}^T = \sum_{ij} A_{ij}^2 \geq 0$.

The advantage of the relative gradient goes beyond the fact that the resulting learning rules avoid matrix inversion. These algorithms were shown by Cardoso and Laheld (1996), in a proof that is readily extendable to the spatio-temporal formulation employed here, to have the property of equivariance: the performance of the source estimator (i.e., the learned separating matrix) is independent of the actual mixing matrix (in fact, their definition is different from (70) but the resulting increments are identical).

The relative gradient (70) can be extended to the convolutive mixing case where \mathbf{G} becomes a matrix of filters \mathbf{G}_m . This is done in Appendix B. Applying this extension to the instantaneous case, we now define the relative gradient of our error function with respect to the separating filters $g_{i,m}$. The relation between

$\delta g_{i,m} = -\epsilon \partial E / \partial g_{i,m}$ and $\delta^R g_{i,m} = -\epsilon \nabla_{g_{i,m}} E$, the ordinary- and relative-gradient increments, is defined via an excursion to the frequency domain by $\delta^R \tilde{g}_{i,k} = \delta \tilde{g}_{i,k} / |\tilde{g}_{i,k}|^2$, where $\delta^R \tilde{g}_{i,k}$ and $\delta \tilde{g}_{i,k}$ are the DFT of $\delta^R g_{i,m}$ and $\delta g_{i,m}$. It follows that

$$\nabla_{g_{i,m}} E = \sum_n \left(\frac{\partial E}{\partial g_i} \times g_i \right)_n g_{i,n+m}. \quad (72)$$

That incrementing $g_{i,m}$ using $\delta^R g_{i,m}$ produces a non-positive change in E is a corollary of (89), a convolutive-mixing analog of the proof (71).

Using (70,72), the relative-gradient learning rules for \mathbf{G} and $g_{i,m}$ can easily be obtained from the rules (67):

$$\begin{aligned} \delta^R \mathbf{G} &= -\epsilon \nabla_{\mathbf{G}} E = \epsilon \mathbf{G} - \epsilon \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{D}_k^{\tilde{g}^*} \tilde{\phi}_k \tilde{\mathbf{x}}_k^\dagger \mathbf{G}, \\ \delta^R g_{i,m} &= -\epsilon \nabla_{g_{i,m}} E = \epsilon \frac{1}{N} \sum_{k=0}^{N-1} e^{i\omega_k m} \left(\tilde{g}_{i,k} - \tilde{\phi}_{i,k} \tilde{u}_{i,k}^* \tilde{g}_{i,k} \right), \end{aligned} \quad (73)$$

where (12) has been used. The time-domain version of these rules (16) follows from DFT relation (3).

A.2 DCA-IT

In this section we derive the DCA-IT error (18) and learning rules (20). As in the frequency-domain case, the relation (8) between the sensors \mathbf{y}_m , sources \mathbf{x}_m and whitened sources \mathbf{u}_m leads to a relation between their model densities:

$$p_y(\mathbf{y}) = |\det \mathbf{G}|^N p_x(\mathbf{x}) = |\det \mathbf{G}|^N \prod_{i=1}^L |g_{i,0}|^N \prod_{m=0}^{N-1} p_{i,m}(u_{i,m}), \quad (74)$$

where the factorial form for p_u (17) has been used. \mathbf{y} stands for $\{\mathbf{y}_0, \dots, \mathbf{y}_{N-1}\}$ and $p_y(\mathbf{y})$ is the joint density of these $N \times L$ variables; the same holds for \mathbf{x} .

The first equality in (74) follows from the fact that for each time point m , the change of variables $\mathbf{x}_m = \mathbf{G} \mathbf{y}_m$ contributes $|\det \mathbf{G}|$ to the transformation $p_y \rightarrow p_x$. To derive the second equality we write $u_{i,m} = (g_i * x_i)_m = \sum_n g_{i,n} x_{i,m-n}$ in a matrix form,

$$\begin{pmatrix} x_{i,0} \\ x_{i,1} \\ \vdots \\ x_{i,N-1} \end{pmatrix} = \begin{pmatrix} g_{i,0} & 0 & \cdot & 0 \\ g_{i,1} & g_{i,0} & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot \\ g_{i,N-1} & g_{i,N-2} & \cdot & g_{i,0} \end{pmatrix} \begin{pmatrix} y_{i,0} \\ y_{i,1} \\ \cdot \\ y_{i,N-1} \end{pmatrix}, \quad (75)$$

and notice that the determinant of the $N \times N$ matrix in (75) is $(g_{i,0})^N$.

The error function (18) is now obtained by substituting p_y (74) in the KL distance (65) and omitting terms independent of the separation parameters.

A.2.1 Ordinary gradient descent

The gradient of E with respect to the separation parameters can be computed with the help of the first identity in (66) and the following two identities,

$$\begin{aligned} \frac{\partial u_{i,m}}{\partial G_{jl}} &= \delta_{ij} \sum_{n=0}^m g_{j,n} y_{l,m-n}, \\ \frac{\partial u_{i,m}}{\partial g_{j,n}} &= \delta_{ij} x_{j,m-n}, \end{aligned} \quad (76)$$

obtained from (11). We then have

$$\begin{aligned}
\delta \mathbf{G} &= \epsilon (\mathbf{G}^T)^{-1} - \epsilon \sum_m \mathbf{D}_m^g (\boldsymbol{\psi} \times \mathbf{y}^T)_{-m}, \\
\delta g_{i,m} &= \epsilon \frac{1}{g_{i,0}} \delta_{m,0} - \epsilon (\boldsymbol{\psi}_i \times x_i)_{-m}, \\
\delta \boldsymbol{\xi}_i &= \epsilon \frac{1}{N} \sum_{m=0}^{N-1} \frac{\partial \log p_{i,m}}{\partial \boldsymbol{\xi}_i},
\end{aligned} \tag{77}$$

where the modified whitened sources $\boldsymbol{\psi}_m$ are defined by

$$\boldsymbol{\psi}_{i,m} = -\frac{1}{N} \frac{\partial \log p_{i,m}}{\partial u_{i,m}}, \tag{78}$$

and the diagonal matrix \mathbf{D}_m^g is given in (4). Note that $(\boldsymbol{\psi} \times \mathbf{y}^T)_{-m}$ is a $L \times L$ matrix whose ij element is $(\boldsymbol{\psi}_i \times y_j)_{-m}$ (see (6)). In component notation, G_{ij} should be incremented by $\delta G_{ij} = \epsilon (\mathbf{G})_{ji}^{-1} - \epsilon \sum_{mn} g_{i,m} \boldsymbol{\psi}_{i,n} y_{j,n-m}$.

A.2.2 Relative gradient descent

The relative-gradient rules for \mathbf{G} and \mathbf{g}_m are obtained by considering the frequency-domain version of the corresponding rules in (77) and using (70,72). This yields

$$\begin{aligned}
\delta^R \mathbf{G} &= \epsilon \mathbf{G} - \epsilon \frac{1}{N} \sum_{k=0}^{N-1} \mathbf{D}_k^{g^*} \tilde{\boldsymbol{\psi}}_k \tilde{\mathbf{x}}_k^\dagger \mathbf{G}, \\
\delta^R g_{i,m} &= \epsilon \frac{1}{N} \sum_{k=0}^{N-1} e^{i\omega_k m} \left(\frac{|\tilde{g}_{i,k}|^2}{g_{i,0}} - \tilde{\boldsymbol{\psi}}_{i,k} \tilde{u}_{i,k}^* \tilde{g}_{i,k} \right).
\end{aligned} \tag{79}$$

The time-domain version of the rule for \mathbf{G} is included in the DCA-IT rules (20). As for \mathbf{g}_m , however, the relative-gradient concept produces a more complicated rule, having the time-domain form

$$\delta^R g_{i,m} = \epsilon \frac{1}{g_{i,0}} (g_i \times g_i)_m - \epsilon \sum_n (\boldsymbol{\psi}_i \times u_i)_n g_{i,n+m}, \tag{80}$$

while offering no advantage in this case.

A.3 DCA-IFT

The frequency-domain relation (62) between $p_{\tilde{y}}$ and $p_{\tilde{u}}$ yields a similar time-domain relation,

$$p_y(y) = |\det \mathbf{G}|^N \prod_{i=1}^L \prod_{k=0}^{N-1} |\tilde{g}_{i,k}| \prod_{m=0}^{N-1} p_{i,m}(u_{i,m}), \tag{81}$$

by exploiting the identity $p_y(y) = N^{N/2} p_{\tilde{y}}(\tilde{y})$ (and similarly for $p_u(u)$) obtained from the DFT definition (3). Using (81) in the KL distance (65) produces the error function (21). The resulting ordinary-gradient learning rules consist of the rule for \mathbf{G} in (77), and the rule for \mathbf{g}_m in (67) with $\boldsymbol{\phi}_m$ replaced by $\boldsymbol{\psi}_m$ (78). The relative-gradient rules (22) are then obtained along the lines described above.

B Convolutional mixing: Error functions and learning rules

B.1 DCA-CF

The derivation of the model sensor density $p_{\tilde{y}}$ for convolutional mixing is analogous to the instantaneous case (Section A.1). Starting from the factorial whitened source density (13), the linear relation $\tilde{u}_{i,k} = \sum_j \tilde{G}_{i,j,k} \tilde{y}_{j,k}$ (33) leads to

$$p_{\tilde{y}}(\tilde{y}) = \prod_{k=0}^{N-1} |\det \tilde{\mathbf{G}}_k| \prod_{k=0}^{N/2} \prod_{i=1}^L P_{i,k}(\tilde{u}_{i,k}), \quad (82)$$

where $\tilde{y} = \{\tilde{y}_0, \dots, \tilde{y}_{N/2}\}$ and $p_{\tilde{y}}(\tilde{y})$ is the joint density of these variables.

To derive (82) we write $\tilde{u}_{i,k} = \sum_j \tilde{G}_{i,j,k} \tilde{y}_{j,k}$ in a matrix form,

$$\begin{pmatrix} \tilde{\mathbf{u}}'_k \\ \tilde{\mathbf{u}}''_k \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{G}}'_k & -\tilde{\mathbf{G}}''_k \\ \tilde{\mathbf{G}}''_k & \tilde{\mathbf{G}}'_k \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{y}}'_k \\ \tilde{\mathbf{y}}''_k \end{pmatrix} \quad (83)$$

with $\tilde{\mathbf{u}}_k = \tilde{\mathbf{u}}'_k + i\tilde{\mathbf{u}}''_k$, and observe that, denoting the $2L \times 2L$ matrix in (83) by $\bar{\mathbf{G}}_k$, each ω_k for $k \neq 0, N/2$ contributes a factor of $|\det \bar{\mathbf{G}}_k|$ to the transformation $p_{\tilde{u}} \rightarrow p_{\tilde{y}}$ (82). To see that $\det \bar{\mathbf{G}}_k = |\det \tilde{\mathbf{G}}_k|^2$, assume $\tilde{\mathbf{G}}_k$ has an eigenvalue λ associated with an eigenvector \mathbf{v} . Then $\tilde{\mathbf{G}}_k^\dagger$ has an eigenvalue λ^* , associated with a (left) eigenvector \mathbf{v}^\dagger . Correspondingly, $\bar{\mathbf{G}}_k$ has two eigenvalues λ, λ^* , associated with the eigenvectors $(i\mathbf{v}^T, \mathbf{v}^T)$ and $(-i\mathbf{v}^\dagger, \mathbf{v}^\dagger)$, respectively. Thus we have $\det \bar{\mathbf{G}}_k = \det \tilde{\mathbf{G}}_k \det \tilde{\mathbf{G}}_k^\dagger = \prod_\lambda |\lambda|^2$. Finally, for $k = 0, N/2$ the imaginary parts in (83) vanish and the relevant contribution is $|\det \tilde{\mathbf{G}}_k|$.

The error function (34) follows by substituting $p_{\tilde{y}}$ (82) in the general expression for the KL distance $E(p_{\tilde{y}}^o, p_{\tilde{y}})$ (64). To obtain the learning rules for the separating filters \mathbf{G}_m and the source parameters ξ_i , we use the following identities:

$$\begin{aligned} \frac{\partial \log |\det \tilde{\mathbf{G}}_k|}{G_{ij,m}} &= \operatorname{Re} \left[e^{i\omega_k m} \left(\tilde{\mathbf{G}}_k^{-1} \right)_{ij}^\dagger \right], \\ \frac{\partial \tilde{u}_{i,k}}{G_{jl,m}} &= e^{-i\omega_k m} \tilde{y}_{l,k} \delta_{ij}. \end{aligned} \quad (84)$$

The resulting ordinary-gradient learning rules in the frequency domain are

$$\delta \tilde{\mathbf{G}}_k = \epsilon \left(\tilde{\mathbf{G}}_k^\dagger \right)^{-1} - \epsilon \tilde{\phi}_k \tilde{\mathbf{Y}}_k^\dagger, \quad (85)$$

where $\delta \tilde{\mathbf{G}}_k$ is the DFT of the ordinary-gradient increment $\delta \mathbf{G}_m$,

$$\delta \mathbf{G}_m = -\epsilon \frac{\partial E}{\partial \mathbf{G}_m} = \frac{1}{N} \sum_{k=0}^{N-1} e^{i\omega_k m} \delta \tilde{\mathbf{G}}_k. \quad (86)$$

Note that it cannot be obtained directly by $\delta \tilde{\mathbf{G}}_k = -\epsilon \partial E / \partial \tilde{\mathbf{G}}_k$ since E is not analytic in $\tilde{\mathbf{G}}_k$. The modified whitened sources $\tilde{\phi}_k$ are defined in (68). The time-domain version of the rule (85) is given by

$$\delta \mathbf{G}_m^T = \epsilon \mathbf{G}_m' - \epsilon (\mathbf{y} \times \phi^T)_{-m}, \quad (87)$$

where $\phi_{i,m}$ are the time-domain counterparts of $\tilde{\phi}_{i,k}$ in (85) obtained by inverse DFT, and \mathbf{G}_m' is the impulse response of the filter matrix whose frequency response is $(\tilde{\mathbf{G}}_k^\dagger)^{-1}$ (compare to $g_{i,m}'$ in (69)).

The rule (87) is inefficient in requiring the inversion of the complex $L \times L$ matrix $\tilde{\mathbf{G}}_k$ for all frequencies ω_k , $k = 0, \dots, N/2$ at each iteration. This can be avoided by extending the concept of the relative gradient

(Cardoso and Laheld 1996; Amari et al. 1996), introduced for instantaneous mixtures (70) in Appendix A, to the convolutive mixing case. We denote the relative gradient of E with respect to \mathbf{G}_m by $\nabla_{\mathbf{G}_m} E$, and the relative-gradient increment of \mathbf{G}_m by $\delta^R \mathbf{G}_m = -\epsilon \nabla_{\mathbf{G}_m} E$. Next, we define it in terms of the ordinary-gradient increment $\delta \mathbf{G}_m = -\epsilon \partial E / \partial \mathbf{G}_m$ by the frequency-domain relation $\delta^R \tilde{\mathbf{G}}_k = \delta \tilde{\mathbf{G}}_k \tilde{\mathbf{G}}_k^\dagger \tilde{\mathbf{G}}_k$, resulting in

$$\nabla_{\mathbf{G}_m} E = \sum_n \left(\frac{\partial E}{\partial \mathbf{G}} \times \mathbf{G}^T \right)_n \mathbf{G}_{n+m}, \quad (88)$$

where T denotes transposition (compare with (70)).

It is left to show that incrementing \mathbf{G}_m by $\delta^R \mathbf{G}_m$ produces a non-positive change in E , generalizing the proof (71) from the instantaneous case. Indeed, for sufficiently small ϵ ,

$$\begin{aligned} \delta E &= E(\mathbf{G}_m + \delta^R \mathbf{G}_m) - E(\mathbf{G}_m) = \sum_m \text{Tr} \frac{\partial E}{\partial \mathbf{G}_m} (\delta^R \mathbf{G}_m)^T = -\epsilon \sum_m \text{Tr} \frac{\partial E}{\partial \mathbf{G}_m} (\nabla_{\mathbf{G}_m} E)^T \\ &= -\epsilon \sum_{mnl} \text{Tr} \frac{\partial E}{\partial \mathbf{G}_m} \mathbf{G}_{l+m}^T \mathbf{G}_{n+l} \left(\frac{\partial E}{\partial \mathbf{G}_n} \right)^T \\ &= -\epsilon \sum_l \text{Tr} \left(\frac{\partial E}{\partial \mathbf{G}} \times \mathbf{G}^T \right)_l \left(\frac{\partial E}{\partial \mathbf{G}} \times \mathbf{G}^T \right)_l^T \leq 0, \end{aligned} \quad (89)$$

where we used $\delta^R \mathbf{G}_m = \sum_{nl} \delta \mathbf{G}_n \mathbf{G}_{n+l}^T \mathbf{G}_{l+m}$ (see (88)) to prove the third equality in (89).

The relative-gradient learning rule is now obtained from (85) by multiplying both sides by $\tilde{\mathbf{G}}_k^\dagger \tilde{\mathbf{G}}_k$:

$$\delta \tilde{\mathbf{G}}_k = \epsilon \tilde{\mathbf{G}}_k - \epsilon \tilde{\phi}_k \tilde{\mathbf{u}}_k^\dagger \tilde{\mathbf{G}}_k, \quad (90)$$

where (33) was used.

As in the instantaneous case (see Section A.1.2), the relative gradient benefit us beyond avoiding matrix inversion. In fact, the proof of Cardoso and Laheld (1996) can be extended to the convolutive case to show that relative-gradient algorithms have the equivariance property, i.e., the quality of the obtained source estimates $\mathbf{x}_m = (\mathbf{G} \star \mathbf{y})_m$ is independent of the actual mixing process. This is a desired feature since it implies that the performance of our separation method is uniform across the space of invertible mixing processes and is not affected by, e.g., the closeness of the mixing to being singular. Of course, this holds only in the absence of noise.

B.2 DCA-CT

Using the factorial whitened source density model (17) and the linear relation $u_{i,m} = \sum_{jn} G_{ij,n} y_{j,m-n}$ (32), we get the model sensor density

$$p_y(y) = |\det \mathbf{G}_0|^N \prod_{i=1}^L \prod_{m=0}^{N-1} p_{i,m}(u_{i,m}), \quad (91)$$

where $y = \{\mathbf{y}_0, \dots, \mathbf{y}_{N-1}\}$. To derive (91) we write (17) in a matrix form,

$$\begin{pmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{N-1} \end{pmatrix} = \begin{pmatrix} \mathbf{G}_0 & 0 & \cdot & 0 \\ \mathbf{G}_1 & \mathbf{G}_0 & \cdot & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{G}_{N-1} & \mathbf{G}_{N-2} & \cdot & \mathbf{G}_0 \end{pmatrix} \begin{pmatrix} \mathbf{y}_0 \\ \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_{N-1} \end{pmatrix}, \quad (92)$$

and observe that the determinant of the $LN \times LN$ matrix in (92) is $(\det \mathbf{G}_0)^N$. The error function corresponding to (91) is (36), obtained from the KL distance (65) after terms independent of the separation parameters are dropped.

To derive the learning rule for the separation filters \mathbf{G}_m we use the first identity in (66), together with

$$\frac{\partial u_{i,m}}{\partial G_{jl,n}} = \delta_{ij} y_{l,m-n}. \quad (93)$$

The resulting rule is

$$\delta \mathbf{G}_m = -\epsilon \frac{\partial E}{\partial \mathbf{G}_m} = \epsilon (\mathbf{G}_0^{-1})^T \delta_{m,0} - \epsilon (\boldsymbol{\psi} \times \mathbf{y}^T)_{-m}, \quad (94)$$

where $\boldsymbol{\psi}_m$ is defined by (78). Note that $(\boldsymbol{\psi} \times \mathbf{y}^T)_{-m}$ is a $L \times L$ matrix whose ij -element is the input-output cross-correlation $\sum_n \psi_{i,n} y_{j,n-m}$.

The inversion of \mathbf{G}_0 at each iteration as required by (94) can be avoided by resorting once again to the relative-gradient idea, this time only with respect to \mathbf{G}_0 . Replacing $\partial E / \partial \mathbf{G}_0$ by $\nabla_{\mathbf{G}_0} E$, defined by (70), we obtain the first rule in (37).

Note that we have not used the convolutive relative gradient (88), which produced an efficient learning rule in the frequency-domain case. In the time-domain case, in contrast, this approach produces a rule which is more complicated than (94) while not avoiding the matrix inversion. To see that, consider the frequency-domain version of (94),

$$\delta \tilde{\mathbf{G}}_k = \sum_{m=0}^{N-1} e^{-i\omega_k m} \delta \mathbf{G}_m = \epsilon (\mathbf{G}_0)^{-1} - \epsilon \frac{1}{N} \tilde{\boldsymbol{\psi}}_k \tilde{\mathbf{y}}_k^\dagger. \quad (95)$$

The corresponding relative-gradient rule is obtained using (88),

$$\delta^R \tilde{\mathbf{G}}_k = \epsilon (\mathbf{G}_0)^{-1} \tilde{\mathbf{G}}_k^\dagger \tilde{\mathbf{G}}_k - \epsilon \tilde{\boldsymbol{\psi}}_k \tilde{\mathbf{u}}_k^\dagger \tilde{\mathbf{G}}_k, \quad (96)$$

leading to the time-domain form

$$\delta^R \mathbf{G}_m = \epsilon (\mathbf{G}_0)^{-1} (\mathbf{G}^T \times \mathbf{G})_m - \epsilon \sum_n (\boldsymbol{\psi} \times \mathbf{u}^T)_n \mathbf{G}_{n+m} \quad (97)$$

(compare to (94,37)).

B.3 DCA-CFT

The relation (82) between the frequency-domain densities $p_{\tilde{y}}(\tilde{y})$ and $p_{\tilde{u}}(\tilde{u})$ leads to a similar relation in the time domain:

$$p_y(y) = \prod_{k=0}^{N-1} |\det \tilde{\mathbf{G}}_k| \prod_{i=1}^L \prod_{m=0}^{N-1} p_{i,m}(u_{i,m}), \quad (98)$$

using $p_y(y) = N^{N/2} p_{\tilde{y}}(\tilde{y})$ and its analog for $p_u(u)$. Substituting (98) in the KL distance (65) results in the error function (38). The corresponding learning rules for \mathbf{G}_m have the same forms as the ordinary- (85) and relative-gradient (90) frequency-domain rules, but with ϕ_m (68) replaced by $\boldsymbol{\psi}_m$ (78). The relative-gradient rule is given in (39).

C Learning rules for the whitened source densities

For the time-domain whitened source density $p_{i,m}$ (19) we use a mixture of K Gaussians with weights $w_{i,\alpha}$, means $\mu_{i,\alpha}$ and variances $\sigma_{i,\alpha}^2$:

$$p_{i,m} = p(u_{i,m}, \boldsymbol{\xi}_i) = \sum_{\alpha=1}^K w_{i,\alpha} g_{i,\alpha}(u_{i,m}) = \sum_{\alpha=1}^K \frac{w_{i,\alpha}}{\sqrt{2\pi\sigma_{i,\alpha}^2}} e^{-\frac{(u_{i,m} - \mu_{i,\alpha})^2}{2\sigma_{i,\alpha}^2}}. \quad (99)$$

To satisfy $\sum_{\alpha} w_{\alpha} = 1$ we write $w_{i,\alpha} = e^{\gamma_{i,\alpha}} / \sum_{\beta} e^{\gamma_{i,\beta}}$. The parameter vector is then $\xi_i = \{\gamma_{i,\alpha}, \mu_{i,\alpha}, \sigma_{i,\alpha}\}$. The learning rules for ξ_i use $\partial \log p_{i,m} / \partial \xi_i$, given by

$$\begin{aligned} \frac{\partial \log p_{i,m}}{\partial \gamma_{i,\alpha}} &= p_{i,\alpha} - w_{i,\alpha} , \\ \frac{\partial \log p_{i,m}}{\partial \mu_{i,\alpha}} &= p_{i,\alpha} \frac{u_{i,m} - \mu_{i,\alpha}}{\sigma_{i,\alpha}^2} , \\ \frac{\partial \log p_{i,m}}{\partial \sigma_{i,\alpha}} &= p_{i,\alpha} \left[\frac{(u_{i,m} - \mu_{i,\alpha})^2}{\sigma_{i,\alpha}^2} - \frac{1}{\sigma_{i,\alpha}} \right] , \end{aligned} \quad (100)$$

where

$$p_{i,\alpha} = \frac{w_{i,\alpha} g_{i,\alpha}}{p_{i,m}} \quad (101)$$

(see (99)).

For the frequency-domain whitened source density $P_{i,k}$ (15) we use a product of Gaussian mixtures for the real and imaginary parts, $P(\tilde{u}_{i,k}) = p(\tilde{u}_{i,k}^I) p(\tilde{u}_{i,k}^J)$, using p (99).

Acknowledgements

We thank A. Bell, B. Bonham, J.-F. Cardoso, M. Kvale, K. Miller, S. Nagarajan, and V. de Sa, for helpful discussions during the course of this work and useful comments on a previous version of this manuscript. We also thank the anonymous referees for numerous suggestions that greatly improved the quality and clarity of the manuscript. Supported by The Office of Naval Research (N00014-94-1-0547) and The Sloan Foundation.

References

- Amari, S., Cichocki, A., and Yang, H.H. (1996). A new learning algorithm for blind signal separation. In Touretzky, D.S., Mozer, M.C., and Hasselmo, M.E. (Eds.), *Advances in Neural Information Processing Systems 8*. MIT Press, Cambridge, MA.
- Atick, J.J. and Redlich, A.N. (1990). Towards a theory of early visual processing. *Neural Computation* **2**, 308-320.
- Atick, J.J. (1992). Could information theory provide an ecological theory of sensory processing? *Network* **3**, 213-251.
- Attias, H. and Schreiner, C.E. (1997). Low-order temporal statistics of natural sounds. In Mozer, M.C., Jordan, M.I., and Petsche, T. (Eds.), *Advances in Neural Information Processing Systems 9*. MIT Press, Cambridge, MA.
- Attias, H. and Schreiner, C.E. (1998). Coding of naturalistic stimuli by auditory midbrain neurons. In Jordan, M.I., Kearns, M.J., and Solla, S.A. (Eds.), *Advances in Neural Information Processing Systems 10*, MIT Press, Cambridge, MA.
- Barlow, H.B. (1989). Unsupervised learning. *Neural Computation* **1**, 295-311.
- Bell, A.J. and Sejnowski, T.J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation* **7**, 1129-1159.
- Bell, A.J. and Sejnowski, T.J. (1996). Edges are the independent components of natural scenes. In In Mozer, M.C., Jordan, M.I., and Petsche, T. (Eds.), *Advances in Neural Information Processing Systems 9*. MIT Press, Cambridge, MA.

- Belouchrani, A., Abed-Meraim, K., Cardoso, J.-F., and Moulines, E. (1997). A blind source separation technique using second-order statistics. *IEEE transactions on Signal Processing* **45**, 434-444.
- Bialek, W., Rieke, F., de Ruyter van Steveninck, R., and Warland, D. (1991). Reading the neural code. *Science* **252**, 1854-1857.
- Bregman, A.S. (1990). *Auditory Scene Analysis*. MIT Press, Cambridge, MA.
- Cardoso, J.-F. and Laheld, B.H. (1996). Equivariant adaptive source separation. *IEEE Transactions on Signal Processing* **44**, 3017-3030.
- Cardoso, J.-F. (1997). Infomax and maximum likelihood for source separation. *IEEE Signal Processing Letters* **4**, 112-114.
- Cichocki, A., Amari, S.-I., and Cao, J. (1996). Blind separation of delayed and convolved signals with self-adaptive learning rate. In *Proceedings of the International Symposium on Non-linear Theory and Applications*, Kochi, Japan.
- Comon, P., Jutten, C., and Herault, J. (1991). Blind separation of sources, Part II: Problem Statement. *Signal Processing* **24**, 11-20.
- Comon, P. (1994). Independent component analysis: a new concept? *Signal Processing* **36**, 287-314.
- Comon, P. (1996). Contrasts for multichannel blind deconvolution. *IEEE Signal Processing Letters* **3**, 209-211.
- Cover, T.M. and Thomas, J.A. (1991). *Elements of Information Theory*. Wiley, New York.
- Dong, D.W. and Atick, J.J. (1995). Temporal decorrelation: a theory of lagged and non-lagged responses in the lateral geniculate nucleus. *Network* **6**, 159-178.
- Everitt, B.S. (1984). *An Introduction to Latent Variable Models*. Chapman and Hall, London.
- Haykin, S. (1996). *Adaptive Filter Theory* (3rd Ed.). Prentice-Hall, New Jersey.
- Hopfield, J.J. (1991). Olfactory computation and object perception. *Proceedings of the National Academy of Sciences* **88**, 6462-6466.
- Jutten, C., and Herault, J. (1991). Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing* **24**, 1-10.
- Lambert, R. (1996). Multichannel blind deconvolution: FIR matrix algebra and separation of multipath mixtures. *PhD Thesis*, University of Southern California.
- Lee, T.-W., Bell, A.J., and Lambert, R. (1997). Blind separation of delayed and convolved sources. In Mozer, M.C., Jordan, M.I., and Petsche, T. (Eds.), *Advances in Neural Information Processing Systems 9*. MIT Press, Cambridge, MA.
- Makeig, S., Bell, A.J., Jung, T.-P., and Sejnowski, T.J. (1996). Independent component analysis of electroencephalographic data. In Touretzky, D.S., Mozer, M.C., and Hasselmo, M.E. (Eds.), *Advances in Neural Information Processing Systems 8*. MIT Press, Cambridge, MA.
- Makeig, S., Bell, A.J., Jung, T.-P., Ghahremani, D., and Sejnowski, T.J. (1997). Blind separation of auditory event-related brain responses into independent components. *Proceedings of the National Academy of Sciences USA* **94**, 10979-10984.
- Molgedey, L. and Schuster, H.J. (1994). Separation of independent signals using time-delayed correlations. *Physical Review Letters* **72**, 3634-3637.
- Oppenheim, A.V. and Schaffer, R.W. (1989). *Discrete-Time Signal Processing*. Prentice-Hall, New Jersey.

- Pearlmutter, B.A. and Parra, L.C. (1997). Maximum likelihood blind source separation: A context-sensitive generalization of ICA. In Mozer, M.C., Jordan, M.I., and Petsche, T. (Eds.), *Advances in Neural Information Processing Systems 9*. MIT Press, Cambridge, MA.
- Pham, D.T. (1996). Blind separation of instantaneous mixture of sources via an independent component analysis. *IEEE Transactions on Signal Processing* **44**, 2768-2779.
- Platt, J.C. and Faggin, F. (1992). Networks for the separation of sources that are superimposed and delayed. In Moody, J.E., Hanson, S.J., and Lippmann, R.P. (Eds.), *Advances in Neural Information Processing Systems 4*. Morgan-Kaufmann.
- Poeppl, D., Attias, H., Rowley, H.A., and Schreiner, C.E. (1997). Dynamic component analysis of auditory evoked neuromagnetic fields. In *Society for Neuroscience Abstracts* **23**.
- Rieke, F., Warland, D., de Ruyter van Steveninck, R., and Bialek, W. (1997). *Spikes: Exploring the Neural Code*. MIT Press, Cambridge, MA.
- Sorouchyari, E. (1991). Blind separation of sources, Part III: Stability analysis. *Signal Processing* **24**, 21-30.
- Thi, H.-L.N. and Jutten, C. (1995). Blind source separation for convolutive mixtures. *Signal Processing* **45**, 209-229.
- Torkkola, K. (1996). Blind separation of convolved sources based on information maximization. In *Neural Networks for Signal Processing VI*, IEEE, New York.
- Yellin, D. and Weinstein, E. (1995). Criteria for multichannel signal separation. *IEEE Transactions on Signal Processing* **42**, 2158-2168.