An online Hebbian learning rule that performs Independent Component Analysis

Claudia Clopath
School of Computer Science and Brain Mind Institute
Ecole polytechnique federale de Lausanne
1015 Lausanne EPFL
claudia.clopath@epfl.ch

Andre Longtin
Center for Neural Dynamics
University of Ottawa
150 Louis Pasteur, Ottawa
alongtin@uottawa.ca

Wulfram Gerstner
School of Computer Science and Brain Mind Institute
Ecole polytechnique federale de Lausanne
1015 Lausanne EPFL
wulfram.gerstner@epfl.ch

Abstract

Independent component analysis (ICA) is a powerful method to decouple signals. Most of the algorithms performing ICA do not consider the *temporal* correlations of the signal, but only higher moments of its *amplitude* distribution. Moreover, they require some preprocessing of the data (whitening) so as to remove second order correlations. In this paper, we are interested in understanding the neural mechanism responsible for solving ICA. We present an online learning rule that exploits delayed correlations in the input. This rule performs ICA by detecting joint variations in the firing rates of pre- and postsynaptic neurons, similar to a local rate-based Hebbian learning rule.

1 Introduction

The so-called cocktail party problem refers to a situation where several sound sources are simultaneously active, e.g. persons talking at the same time. The goal is to recover the initial sound sources from the measurement of the mixed signals. A standard method of solving the cocktail party problem is independent component analysis (ICA), which can be performed by a class of powerful algorithms. However, classical algorithms based on higher moments of the signal distribution [1] do not consider temporal correlations, i.e. data points corresponding to different time slices could be shuffled without a change in the results. But time order is important since most natural signal sources have intrinsic temporal correlations that could potentially be exploited. Therefore, some algorithms have been developed to take into account those temporal correlations, e.g. algorithms based on delayed correlations [2, 3, 4, 5] potentially combined with higher-order statistics [6], based on innovation processes [7], or complexity pursuit [8]. However, those methods are rather algorithmic and most of them are difficult to interpret biologically, e.g. they are not online or not local or require a preprocessing of the data.

Biological learning algorithms are usually implemented as an online Hebbian learning rule that triggers changes of synaptic efficacy based on the correlations between pre- and postsynaptic neurons. A Hebbian learning rule, like Oja's learning rule [9], combined with a linear neuron model, has been shown to perform principal component analysis (PCA). Simply using a nonlinear neuron combined with Oja's learning rule allows one to compute higher moments of the distributions which yields ICA if the signals have been preprocessed (whitening) at an earlier stage [1]. In this paper, we are

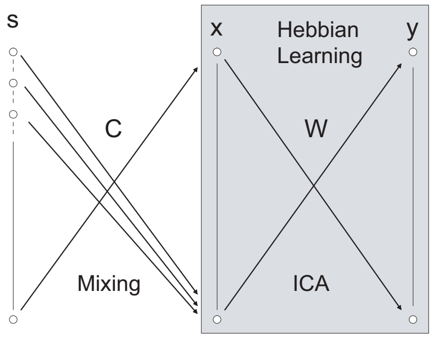

Figure 1: The sources $\mathbf{s}$ are mixed with a matrix C, $\mathbf{x} = C\mathbf{s}$, $\mathbf{x}$ are the presynaptic signals. Using a linear neuron $\mathbf{y} = W\mathbf{x}$, we want to find the matrix W which allows the postsynaptic signals $\mathbf{y}$ to recover the sources, $\mathbf{y} = P\mathbf{s}$, where P is a permutation matrix with different multiplicative constants.

interested in exploiting the correlation of the signals at different time delays, i.e. a generalization of the theory of Molgedey and Schuster [4]. We will show that a linear neuron model combined with a Hebbian learning rule based on the joint firing rates of the pre- and postsynaptic neurons of different time delays performs ICA by exploiting the temporal correlations of the presynaptic inputs.

2 Mathematical derivation of the learning rule

2.1 The problem

We assume statistically independent autocorrelated source signals s_i with mean $< s_i >= 0$ ($<>$ means averaging over time) and correlations $< s_i(t)s_j(t') >= K_i(|t-t'|)\delta_{ij}$. The sources $\mathbf{s}$ are mixed by a matrix C

$$\mathbf{x} = C\mathbf{s}, \tag{1}$$

where $\mathbf{x}$ are the mixed signals recorded by a finite number of receptors (bold notation refers to a vector). We think of the receptors as presynaptic neurons that are connected via a weight matrix W to postsynaptic neurons. We consider linear neurons [9], so that the postsynaptic signals $\mathbf{y}$ can be written

$$\mathbf{y} = W\mathbf{x}. \tag{2}$$

The aim is to find a learning rule that adjusts the appropriate weight matrix W to W^* (* denotes the value at the solution) so that the postsynaptic signals $\mathbf{y}$ recover the independent sources $\mathbf{s}$ (Fig 1), i.e. $\mathbf{y} = P\mathbf{s}$ where P is a permutation matrix with different multiplicative constants (the sources are recovered in a different order up to a multiplicative constant), which means that, neglecting P,

$$W^* = C^{-1}. \tag{3}$$

To solve this problem we extend the theory of Molgedey and Schuster [4] in order to derive an online biological hebbian rule.

2.2 Theory of Molgedey and Schuster and generalization

The paper of Molgedey and Schuster [4] focuses on the instantaneous correlation matrix but also the time delayed correlations $M_{ij} =< x_i(t)x_j(t+\tau) >$ of the incoming signals. Since the correlation matrix M_{ij} is symmetric, it has up to $n(n+1)/2$ independent elements. However, the unknown mixing matrix C has potentially n^2 elements (for n sources and n detectors). Therefore, we need to evaluate two delayed correlation matrices M and $\bar{M}$ with two different time delays defined as

$$M_{ij} =< x_i(t)x_j(t+\tau_2) > \qquad \bar{M}_{ij} =< x_i(t)x_j(t+\tau_1) > \tag{4}$$

to get enough information about the mixing process [10].

From equation 1, we obtain the relation $M_{ij} = \sum_l C_{il}C_{jl}\Lambda_{ll}$ and similarly $\bar{M}_{ij} = \sum_l C_{il}C_{jl}\bar{\Lambda}_{ll}$ where $\Lambda_{ij} = \delta_{ij}K_i(\tau_2)$ and $\bar{\Lambda}_{ij} = \delta_{ij}K_i(\tau_1)$ are diagonal matrices. Since $M = C\Lambda C^T$ and $\bar{M} = C\bar{\Lambda}C^T$, we have

$$(M\bar{M}^{-1})C = C(\Lambda\bar{\Lambda}^{-1}). \tag{5}$$

It follows that C can be found from an eigenvalue problem. Since C is the mixing matrix, a simple algorithmic inversion allows Molgedey and Schuster to recover the original sources [4].

2.3 Our learning rule

In order to understand the putative neural mechanism performing ICA derived from the formalism developed above, we need to find an online learning rule describing changes of the synapses as a function of pre- and postsynaptic activity. Taking the inverse of (5), we have $C^{-1}\bar{M}M^{-1} = \bar{\Lambda}\Lambda^{-1}C^{-1}$. Therefore, for weights that solve the ICA problem we expect because of (3) that

$$W^*\bar{M} = \bar{\Lambda}\Lambda^{-1}W^*M, \tag{6}$$

which defines the weight matrix W^* at the solution.

For the sake of simplicity, consider only one linear postsynaptic neuron. The generalization to many postsynaptic neurons is straightforward (see section 4). The output signal y of the neuron can be written as $y = \mathbf{w}^{*\mathbf{T}}\mathbf{x}$, where $\mathbf{w}^{*\mathbf{T}}$ is a row of the matrix W^*. Then equation 6 can be written as

$$\mathbf{w}^{*T}\bar{M} = \lambda \mathbf{w}^{*\mathbf{T}}M, \tag{7}$$

where λ is one element of the diagonal matrix $\bar{\Lambda}\Lambda^{-1}$.

In order to solve this equation, we can use the following iterative update rule with update parameter γ.

$$\dot{\mathbf{w}} = \gamma[\mathbf{w}^{\mathbf{T}}\bar{M} - \lambda\mathbf{w}^{\mathbf{T}}M]. \tag{8}$$

The fixed point of this update rule is giving by (7), i.e. $\mathbf{w} = \mathbf{w}^*$. Furthermore, multiplication of (7) with $\mathbf{w}$ yields $\lambda = \frac{\mathbf{w}^{\mathbf{T}}\bar{M}\mathbf{w}}{\mathbf{w}^{\mathbf{T}}M\mathbf{w}}$.
If we insert the definition of M from (2), we obtain the following rule

$$\dot{\mathbf{w}} = \gamma[< y(t)\mathbf{x}(\mathbf{t}+\tau_\mathbf{1}) > -\lambda < y(t)\mathbf{x}(\mathbf{t}+\tau_\mathbf{2}) >], \tag{9}$$

with a parameter λ given by

$$\lambda = \frac{< y(t)y(t+\tau_1) >}{< y(t)y(t+\tau_2) >}.$$

It is possible to show that $\dot{\mathbf{w}}$ is orthogonal to $\mathbf{w}$. This implies that to first order (in $|\dot{\mathbf{w}}/\mathbf{w}|$), $\mathbf{w}$ will keep the same norm during iterations of (9).

The rule 9 we derived is a batch-rule, i.e. it averages over all sample signals. We convert this rule into an online learning rule by taking a small learning rate γ and using an online estimate of λ.

$$\begin{aligned} \dot{\mathbf{w}} &= \gamma[y(t)\mathbf{x}(\mathbf{t}+\tau_\mathbf{1}) - \frac{\lambda_1}{\lambda_2}y(t)\mathbf{x}(\mathbf{t}+\tau_\mathbf{2})] \qquad (10)\\ \tau_\lambda\dot{\lambda_1} &= -\lambda_1 + y(t)y(t+\tau_1)\\ \tau_\lambda\dot{\lambda_2} &= -\lambda_2 + y(t)y(t+\tau_2). \end{aligned}$$

Note that the rule defined in (10) uses information on the correlated activity $\mathbf{x}y$ of pre- and postsynaptic neurons as well as an estimate of the autocorrelation $< yy >$ of the postsynaptic neuron. τ_λ is taken sufficiently long so as to average over a representative sample of the signals and $|\gamma| \ll 1$ is a small learning rate. Stability properties of updates under rule (10) are discussed in section 4.

3 Performances of the learning rule

A simple example of a cocktail party problem is shown in Fig 2 where two signals, a sinus and a ramp (saw-tooth signal), have been mixed. The learning rule converges to a correct set of synaptic

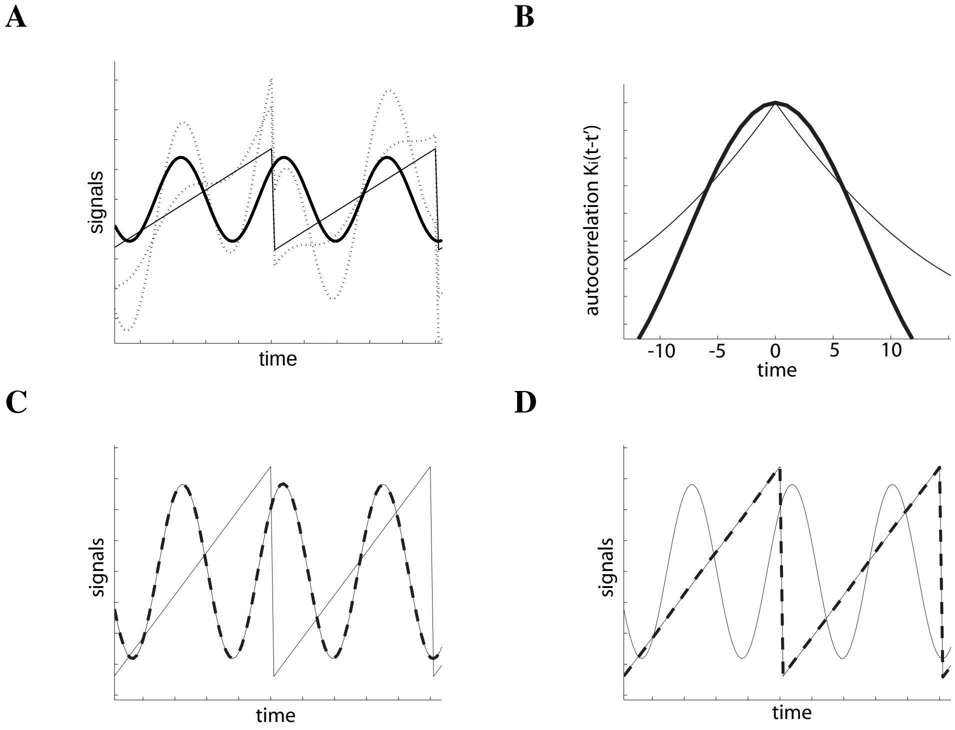

Figure 2: A. Two periodic source signals, a sinus (thick solid line) and a ramp (thin solid line), are mixed into the presynaptic signals (dotted lines). B. The autocorrelation functions of the two source signals are shown (the sinus in thick solid line and the ramp in thin solid line). The sources are normalized so that $\Lambda(0) = 1$ for both. C. The learning rule with $\tau_1 = 3$ and $\tau_2 = 0$ extracts the sinusoidal output signal (dashed) composed to the two input signals. In agreement with the calculation of stability, $\gamma > 0$, the output is recovering the sinus source because $\Lambda_{sin}(3) > \Lambda_{ramp}(3)$. D. The learning rule with $\tau_1 = 10$, $\tau_2 = 0$, converges to the other signal (dashed line), i.e. the ramp, because $\Lambda_{ramp}(10) > \Lambda_{sin}(10)$. Note that the signals have been rescalled since the learning rule recovers the signals up to a multiplicative factor.

weights so that the postsynaptic signal recovers correctly one of the sources. Postsynaptic neurons with different combinations of τ_1 and τ_2 are able to recover different signals (see the section 4 on Stability). In the simulations, we find that the convergence is fast and the performance is very accurate and stable. Here we show only a two-sources problem for the sake of visual clarity. However, the rule can easily recover several mixed sources that have different temporal characteristics.

Fig 3 shows an ICA problem with sources $s(t)$ generated by an Ornstein-Uhlenbeck process of the form $\tau_{s_i}\dot{s_i} = -s_i + \xi$, where ξ is some gaussian noise. The different sources are characterized by different time constants. The learning rule is able to decouple these colored noise signals with gaussian amplitude distribution since they have different temporal correlations.

Finally, Fig 4 shows an application with nine different sounds. We used 60 postsynaptic neurons with time delays τ_1 chosen uniformly in an interval [1,30ms] and $\tau_2 = 0$. Globally 52 of the 60 neurons recovered exactly 1 source (A, B) and the remaining 8 recovered mixtures of 2 sources (E). One postsynaptic neuron is recovering one of the sources depending on the source's autocorrelation at time τ_1 and τ_2 (.i.e. the source with the biggest autocorrelation at time τ_1 since $\tau_2 = 0$ for all neurons, see section Stability). A histogram (C) shows how many postsynaptic neurons recover each source. However, as it will become clear from the stability analysis below, a few specific postsynaptic neurons tuned to time delays, where the autocorrelation functions intersect (D, at time $\tau_1 = 3$ms and $\tau_2 = 0$), cannot recover one of the sources precisely (E).

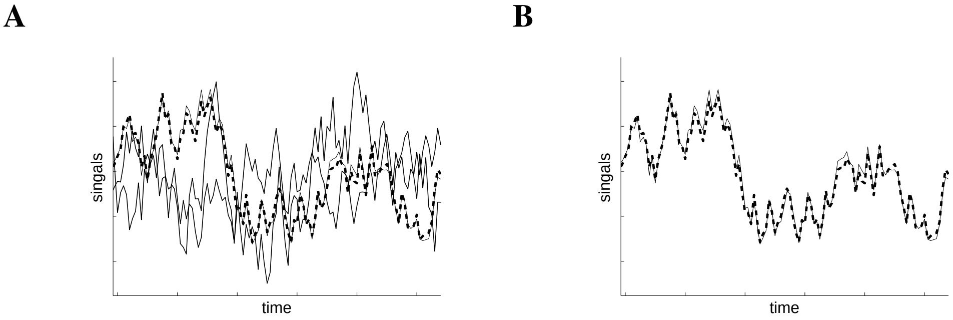

Figure 3: A. The 3 source signals (solid lines generated with the equation $\tau_{s_i}\dot{s_i} = -s_i + \xi$ with different time constants, where ξ is some gaussian noise) are plotted together with the output signal (dashed). The learning rule is converging to one of the sources. B. Same as before, but only the one signal (solid) that was recovered is shown together with the neuronal output (dashed).

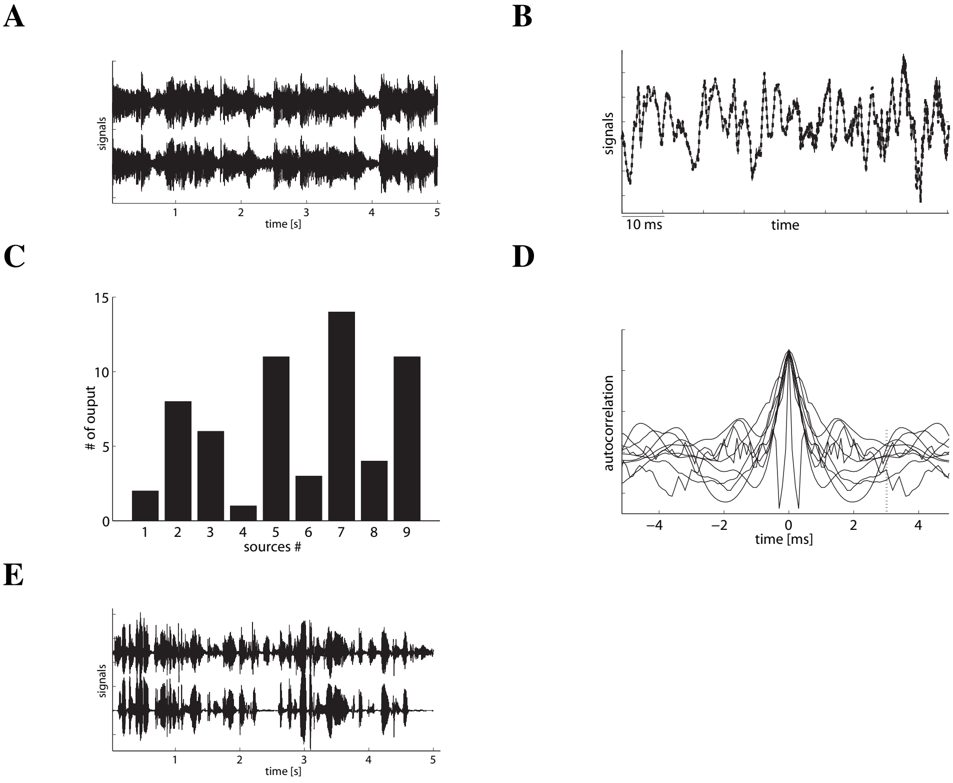

Figure 4: Nine different sound sources from [11] were mixed with a random matrix. 60 postsynaptic neurons tuned to different τ_1 and τ_2 were used in order to recover the sources, i.e. τ_1 varies from 1ms to 30ms by steps of 0.5ms and $\tau_2 = 0$ for all neurons. A. One source signal (below) is recovered by one of the postsynaptic neurons (above, for clarity reason, the output is shifted upward). B. Zoom on one source (solid line) and one output (dashed line). C. Histogram of the number of postsynaptic neurons recovering each sources. D. Autocorrelation of the different sources. There are several sources with the biggest autocorrelation at time 3ms. E. The postsynaptic neuron tuned to a $\tau_1 = 3$ms and $\tau_2 = 0$ (above) is not able to recover properly one of the sources even though it still performs well except for the low amplitude parts of the signal (below).

4 Stability of the learning rule

In principle our online learning rule (10) could lead to several solutions corresponding to different fixed points of the dynamics. Fixed points will be denoted by $\mathbf{w}^* = \mathbf{e}_k$, which are by construction the row vectors of the decoupling matrix W^* (see (5) and (7)). The rule 10 has two parameters, i.e. the delays τ_1 and τ_2 (the τ_λ is considered fixed). We assume that in our architecture, these delays characterize different properties of the postsynaptic neuron. Neurons with different choices of τ_1 and τ_2 will potentially recover different signals from the same mixture. The stability analysis will show which fixed point is stable depending on the autocorrelation functions of the signals and the delays τ_1 and τ_2.

We analyze the stability, assuming small perturbation of the weights, i.e. $\mathbf{w} = \mathbf{e}_i + \epsilon \mathbf{e}_j$ where $\{\mathbf{e}^k\}$, the basis of the matrix C^{-1}, are the fixed points. We obtain the expression (see Appendix for calculation details)

$$\dot{\epsilon} = \gamma \epsilon \frac{\Lambda_{jj}(\tau_1)\Lambda_{ii}(\tau_2) - \Lambda_{ii}(\tau_1)\Lambda_{jj}(\tau_2)}{\Lambda_{ii}(\tau_2)}, \tag{11}$$

where $\Lambda(\tau)_{ij} =< s_i(t)s_j(t+\tau) >$ is the diagonal correlation matrix.

To illustrate the stability equation (11), let us take $\tau_1 = 0$ and assume that $\Lambda_{ii}(0) = \Lambda_{jj}(0)$, i.e. all signals have the same zero-time-lag autocorrelation. In this case (11) reduces to $\dot{\epsilon} = \gamma\epsilon[\Lambda_{jj}(\tau_1) - \Lambda_{ii}(\tau_1)]$. That is the solution e_i is stable if $\Lambda_{jj}(\tau_1) < \Lambda_{ii}(\tau_1)$ for all directions e_j (with biggest autocorrelation at time τ_1) for $\gamma > 0$. If $\gamma < 0$, the solution e_i is stable for $\Lambda_{jj}(\tau_1) > \Lambda_{ii}(\tau_1)$.

This stability relation is verified in the simulations. Fig 2 shows two signals with different autocorrelation functions. In this example, we chose $\tau_1 = 0$ and $\Lambda(0) = \mathbb{I}$, i.e. the signals are normalized. The learning rule is recovering the signal with the biggest autocorrelation at time τ_1, $\Lambda_{kk}(\tau_1)$, for a positive learning rate.

5 Comparison between Spatial ICA and Temporal ICA

One of the algorithms most used to solve ICA is FastICA [1]. It is based on an approximation of negentropy and is purely spatial, i.e. it takes into account only the amplitude distribution of the signal, but not it's temporal structure. Therefore we show an example (Fig. 5), where three signals generated by Ornstein-Uhlenbeck processes have the same spatial distribution but different time constants of the autocorrelation. With a spatial algorithm data points corresponding to different time slices can be shuffled without any change in the results. Therefore, it cannot solve this example. We tested our example with FastICA downloaded from [11] and it failed to recover the original sources (Fig. 5). However, to our surprise, FastICA could for very few trial solve this problem even though the convergence was not stable. Indeed, since FastICA algorithm is an iterative online algorithm, it takes the signals in the temporal order in which they arrive. Therefore temporal correlations can in some cases be taken into account even though this is not part of the theory of FastICA.

6 Discussions and conclusions

We presented a powerful online learning rule that performs ICA by computing joint variations in the firing rates of pre- and postsynaptic neurons at different time delays. This is very similar to a standard Hebbian rule with exception of an additional factor λ which is an online estimate of the output correlations at different time delays. The different delay times τ_1, τ_2 are necessary to recover different sources. Therefore properties varying between one postsynaptic neuron and the next could lead to different time delays used in the learning rule. We could assume that the time delays are intrinsic properties of each postsynaptic neuron due to for example the distance on the dendrites where the synapse is formed [12], i.e. due to different signal propagation time. The calculation of stability shows that a postsynaptic neuron will recover the signal with the biggest autocorrelation at the considered delay time or the smallest depending of the sign of the learning rates. We assume that for biological signals autocorrelation functions cross so that it's possible with different postsynaptic neurons to recover all the signals.

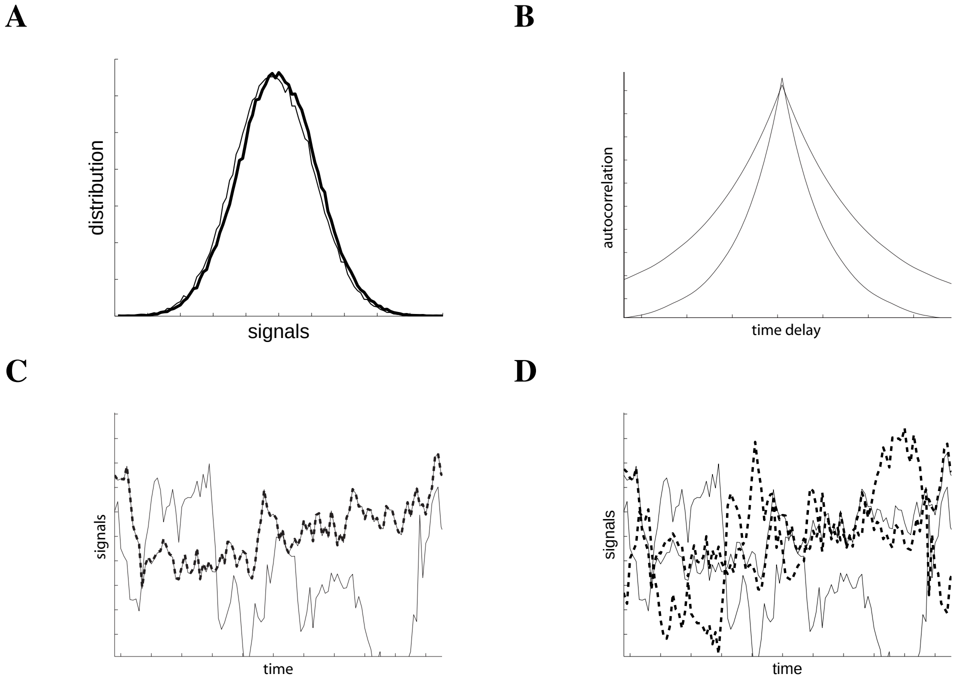

Figure 5: Two signals generated by an Ornstein-Uhlenbeck process are mixed. A. The signals have the same spatial distributions. B. The time constants of the autocorrelations are different. C. Our learning rule converges to an output (dashed line) recovering one of the signals source (solid line). D. FastICA (dashed line) doesn't succeed to recover the sources (solid line).

The algorithm assumes centered signals. However for a complete mapping of those signals to neural rates, we have to consider positive signals. Nevertheless we can easily compute an online estimate of the mean firing rate and remove this mean from the original rates. This way the algorithm still holds taking neural rates as input.

Hyvaerinen proposed an ICA algorithm [8] based on complexity pursuit. It uses the non-gaussianity of the residuals once the part of the signals that is predictable from the temporal correlations has been removed. The update step of this algorithm has some similarities with our learning rule even though the approach is completely different since we want to exploit temporal correlations directly rather than formally removing them by a ”predictor”. We also do not assume pre-whitened data and are not considering nongaussianity.

Our learning rule considers smooth signals that are assumed to be rates. However, it is commonly accepted that synaptic plasticity takes into account spike trains of pre- and postsynaptic neurons looking at the precise timing of the spikes, i.e. Spike Timing Dependent Plasticity (STDP) [13, 14, 15]. Therefore a spike-based description of our algorithm is currently under study.

Appendix: Stability calculation

By construction, the row vectors $\{\mathbf{e}_k,\, k = 1,..,n\}$ of $W^* = C^{-1}$, the inverse of the mixing matrix, are solutions of the batch learning rule 9 (n is the number of sources). Assume one of these row vectors $\mathbf{e}_i^T$, (i.e. a fixed point of the dynamic), and consider $\mathbf{w} = \mathbf{e}_i + \epsilon\mathbf{e}_j$ a small perturbation in direction $\mathbf{e}_j^T$. Note that $\{\mathbf{e}_k\}$ is a basis because $det(C) \neq 0$ (the matrix must be invertible). The rule (9) becomes:

$$
\begin{aligned}
\dot{\epsilon}\mathbf{e}_i =&\gamma[< x(t+\tau_1)(\mathbf{e}_i+\epsilon\mathbf{e}_j)^T x(t) > \\
&- \frac{< (\mathbf{e}_i+\epsilon\mathbf{e}_j)^T x(t)(\mathbf{e}_i+\epsilon\mathbf{e}_j)^T x(t+\tau_1) >}{< (\mathbf{e}_i+\epsilon\mathbf{e}_j)^T x(t)(\mathbf{e}_i+\epsilon\mathbf{e}_j)^T x(t+\tau_2 >)} < x(t+\tau_2 >)(\mathbf{e}_i+\epsilon\mathbf{e}_j)^T x(t) >].
\end{aligned}
\tag{12}
$$

We can expand the terms on the righthand side to first order in ϵ. Multiplying the stability expression by $\mathbf{e}_j^T$ (here we can assume that $\mathbf{e}_j^T\mathbf{e}_j = 1$ since the recovering of the sources are up to a multiplicative constant), we find:

$$
\begin{aligned}
\dot{\epsilon} =&\gamma\epsilon\frac{[\mathbf{e}_j^T C\Lambda(\tau_1)C^T\mathbf{e}_j][\mathbf{e}_i^T C\Lambda(\tau_2)C^T\mathbf{e}_i] - [\mathbf{e}_i^T C\Lambda(\tau_1)C^T\mathbf{e}_i][\mathbf{e}_j^T C\Lambda(\tau_2)C^T\mathbf{e}_j]}{\mathbf{e}_i^T C\Lambda(\tau_2)C^T\mathbf{e}_i} \\
&- \epsilon\frac{4[\mathbf{e}_i^T C\Lambda(\tau_1)C^T\mathbf{e}_j][\mathbf{e}_j^T C\Lambda(\tau_2)C^T\mathbf{e}_i]}{\mathbf{e}_i^T C\Lambda(\tau_2)C^T\mathbf{e}_i}.
\end{aligned}
\tag{13}
$$

where $\Lambda(\tau)_{ij} =< s_i(t)s_j(t+\tau) >$ is the diagonal matrix.

This expression can be simplified because $\mathbf{e}_i^T$ is a row of $W^* = C^{-1}$, so that $\mathbf{e}_i^T C$ is the unit vector of the form (0,0,...,1,0,...) where the position of the "1" indicates the solution number 0. Therefore, we have $e_i^T C\Lambda(\tau)C^T e_k = \Lambda(\tau)_{ik}$.

The expression of stability becomes

$$
\dot{\epsilon} = \gamma\epsilon\frac{\Lambda_{jj}(\tau_1)\Lambda_{ii}(\tau_2) - \Lambda_{ii}(\tau_1)\Lambda_{jj}(\tau_2)}{\Lambda_{ii}(\tau_2)}
\tag{14}
$$

References

[1] A. Hyvaerinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley-Interscience, 2001.

[2] L Tong, R Liu, VC Soon, and YF Huang. Indeterminacy and identifiability of blind identification. *IEEE Trans. on Circuits and Systems*, 1991.

[3] A. Belouchrani, KA. Meraim, JF. Cardoso, and E. Moulines. A blind source separation technique based on second order statistics. *IEEE Trans. on Sig. Proc.*, 1997.

[4] L. Molgedey and H.G. Schuster. Separation of a mixture of independent signals using time delayed correlations. *Phys. Rev. Lett.*, 72:3634–37, 1994.

[5] A. Ziehe and K. Muller. Tdsep – an efficient algorithm for blind separation using time structure.

[6] KR. Mueller, P. Philips, and A. Ziehe. Jade td : Combining higher-order statistics and temporal information for blind source separation (with noise). *Proc. Int. Workshop on ICA*, 1999.

[7] A. Hyvaerinen. Independent component analysis for time-dependent stochastic processes. *Proc. Int. Conf. on Art. Neur. Net.*, 1998.

[8] A. Hyvaerinen. Complexity pursuit: Separating interesting components from time-series. *Neural Computation*, 13:883–898, 2001.

[9] E. Oja. A simplified neuron model as principal component analyzer. *J. Math. Biol.*, 15:267 –273, 1982.

[10] J.J. Hopfield. Olfactory computation and object perception. *PNAS*, 88:6462–6466, 1991.

[11] H. Gavert, J. Hurri, J. Sarela, and A. Hyvarinen. Fastica and cocktail party demo. *http://www.cis.hut.fi/projects/ica/*.

[12] R. C. Froemke, M. Poo, and Y. Dan. Spike-timing dependent synaptic plasticity depends on dentritic location. *Nature*, 434:221–225, 2005.

[13] G. Bi and M. Poo. Synaptic modification by correlated activity: Hebb's postulate revisited. *Annual Review of Neuroscience*, 2001.

[14] H. Markram, J. Lübke, M. Frotscher, and B. Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275:213–215, 1997.

[15] W. Gerstner, R. Kempter, JL. van Hemmen, and H. Wagner. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383:76–78, 1996.