Quantifying statistical interdependence by message passing on graphs—Part II: Multidimensional point processes

LETTER **Communicated by Jonathan Victor**

Quantifying Statistical Interdependence by Message Passing on Graphs—Part II: Multidimensional Point Processes

J. Dauwels
jdauwels@mit.edu
Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A., and Amari Research Unit, RIKEN Brain Science Institute, Saitama 351-0198, Japan

F. Vialatte
fvialatte@brain.riken.jp
Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Saitama 351-0198, Japan

T. Weber
theo_w@mit.edu
Operations Research Center, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.

T. Musha
musha@bfl.co.jp
Brain Functions Laboratory, Yokohama 226-8510, Japan

A. Cichocki
cia@brain.riken.jp
Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Saitama 351-0198, Japan

Stochastic event synchrony is a technique to quantify the similarity of pairs of signals. First, events are extracted from the two given time series. Next, one tries to align events from one time series with events from the other. The better the alignment, the more similar the two time series are considered to be. In Part I, the companion letter in this issue, one-dimensional events are considered; this letter concerns multidimensional events. Although the basic idea is similar, the extension to multidimensional point processes involves a significantly more difficult combinatorial problem and therefore is nontrivial.

Also in the multidimensional case, the problem of jointly computing the pairwise alignment and SES parameters is cast as a statistical inference problem. This problem is solved by coordinate descent, more

specifically, by alternating the following two steps: (1) estimate the SES parameters from a given pairwise alignment; (2) with the resulting estimates, refine the pairwise alignment. The SES parameters are computed by maximum a posteriori (MAP) estimation (step 1), in analogy to the one-dimensional case. The pairwise alignment (step 2) can no longer be obtained through dynamic programming, since the state space becomes too large. Instead it is determined by applying the max-product algorithm on a cyclic graphical model.

In order to test the robustness and reliability of the SES method, it is first applied to surrogate data. Next, it is applied to detect anomalies in EEG synchrony of mild cognitive impairment (MCI) patients. Numerical results suggest that SES is significantly more sensitive to perturbations in EEG synchrony than a large variety of classical synchrony measures.

1 Introduction

The problem of detecting correlations between neural signals has attracted quite some attention in the neuroscience community. Several studies have related neural synchrony to attention and cognition (e.g., (Buzsáki, 2006)). Recently it has been demonstrated that patterns of neural synchronization flexibly trigger patterns of neural interactions (Womelsdorf et al., 2006). Moreover, it is has frequently been reported that abnormalities in neural synchrony lie at the heart of a variety of brain disorders such as Alzheimer's and Parkinson's diseases (e.g., Matsuda, 2001; Jeong, 2004; Uhlhaas & Singer, 2006). In response to those findings, efforts have been made to develop novel quantitative methods to detect statistical dependencies in brain signals (see, e.g., Stam, 2005; Quian Quiroga, Kraskov, Kreuz, & Grassberger, 2002; Pereda, Quian Quiroga, & Bhattacharya, 2005).

In this letter, we extend stochastic event synchrony (SES) from one-dimensional point processes (explored in the companion letter) to multi-dimensional processes. The underlying principle is identical, but the inference algorithm to compute the SES parameters is fundamentally different. The basic idea is again the following. First, we extract events from the two given time series. Next, we try to align events from one time series with events from the other. The better the alignment, the more similar the two time series are considered to be. More precisely, the similarity is quantified by the following parameters: time delay, variance of the timing jitter, fraction of "noncoincident" events, and average similarity of the aligned events. In this letter, we mostly focus on point processes in time-frequency domain. The average event similarity is in that case described by two parameters: the average frequency offset between events in the time-frequency plane and the variance of the frequency offset (frequency jitter). SES then consists of five parameters, which quantify the synchrony of oscillatory events and provide an alternative to classical synchrony measures that quantify amplitude or phase synchrony.

The pairwise alignment of point processes is again cast as a statistical inference problem. However, inference in that model cannot be carried out by dynamic programming, since the state space is too large. Instead we apply the max-product algorithm on a cyclic graphical model (Jordan, 1999; Loeliger, 2004; Loeliger et al., 2007); the inference method is now an iterative algorithm. Based on a result in Bayati, Shah, and Sharma (2005; generalized in Sanghavi, 2007, 2008), we show that this algorithm yields the optimal alignment as long as the optimal alignment is unique.

In this letter, we consider only pairs of point processes, but the methods may be extended to multiple point processes. That extension, however, is nontrivial and goes beyond the scope of this letter. It will be described in a future report.

As in the one-dimensional case, the method may be applied to any kind of time series (e.g., finance, oceanography, seismology). However, we here consider only electroencephalogram (EEG) signals. More specifically, we present promising results on the early prediction of Alzheimer's disease based on EEG.

This letter is organized as follows. In the next section, we introduce SES for multidimensional point processes. We describe the underlying statistical model in section 3. Inference in that model is carried out by applying the max-product algorithm on a factor graph of that model. That factor graph is discussed in section 4; the inference method is outlined in section 5 and derived in detail in appendix C. In section 6, we list several extensions of the basic multidimensional SES model. In section 7, we investigate the robustness and reliability of the SES inference method by means of surrogate data. In section 8, we apply that method to detect abnormalities in the EEG synchrony of mild cognitive impairment patients. We offer some concluding remarks in section 9.

2 Principle

Suppose that we are given a pair of continuous-time signals, such as EEG signals recorded from two different channels, and we wish to determine the similarity of those two signals. As a first step, we extract point processes from those signals, which may be achieved in various ways. As an example, we generate point processes in time-frequency domain. First, the time-frequency ("wavelet") transform of each signal is computed in a frequency band $f \in [f_{\min}, f_{\max}]$. Next, those maps are approximated as a sum of half-ellipsoid basis functions, referred to as "bumps" (see Figure 1; we provide more details on bump modeling in section 8.2.3). Each bump is described by five parameters: time t, frequency f, width Δt, height Δf, and amplitude w. The resulting bump models $e = ((t_1, f_1, \Delta t_1, \Delta f_1, w_1), \ldots, (t_n, f_n, \Delta t_n, \Delta f_n, w_n))$ and $e' = ((t'_1, f'_1, \Delta t'_1, \Delta f'_1, w'_1), \ldots, (t'_{n'}, f'_{n'}, \Delta t'_{n'}, \Delta f'_{n'}, w'_{n'}))$ represent the most

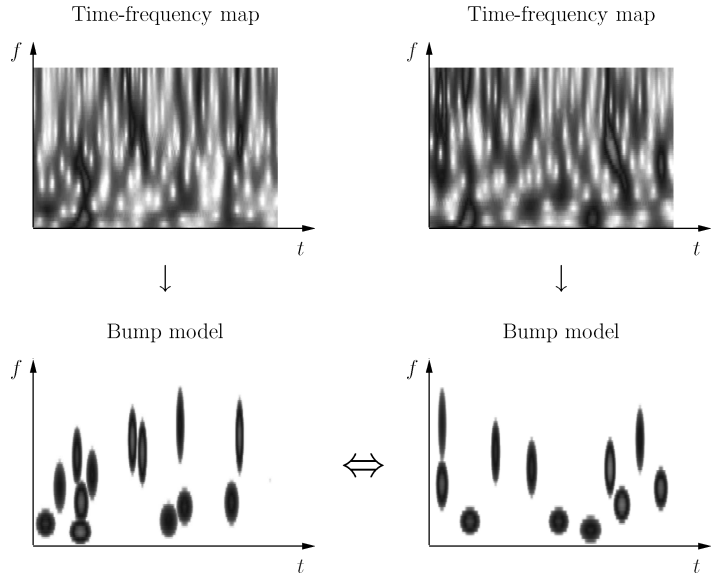


Figure 1: Two-dimensional stochastic event synchrony. (Top) Two given EEG signals in the time-frequency domain. (Bottom) Bump models extracted from those time-frequency maps. Stochastic event synchrony quantifies the similarity of two such bump models.

prominent oscillatory activity in the signals at hand. This activity may correspond to various physical or biological phenomena—for example:

- Oscillatory events in EEG and other brain signals are believed to occur when assemblies of neurons are spiking in synchrony (Buzsáki, 2006; Nunez & Srinivasan, 2006).
- Oscillatory events in calcium imaging data are due to oscillations of intracellular calcium, which are believed to play an important role in signal transduction between cells (see, e.g., Völkers et al., 2007).
- Oscillations and waves are of central interest in several fields beyond neuroscience, such as oceanography (e.g., oceanic "normal modes" caused by convection; Kantha & Clayson, 2000) and seismography (e.g., free earth oscillations and earth oscillations induced by earthquakes, hurricanes, and human activity; Alder, Fernbach, & Rotenberg, 1972).

In the following, we develop SES for bump models. In this setting, SES quantifies the synchronous interplay between oscillatory patterns in two given signals while it ignores the other components in those signals ("background activity"). In contrast, classical synchrony measures such as amplitude or phase synchrony are computed from the entire signal; they make

no distinction between oscillatory components and background activity. As a consequence, SES captures alternative aspects of similarity and hence provides complementary information about synchrony.

Besides bump models, SES may be applied to other sparse representations of signals—for example:

- Matching pursuit (Mallat & Zhang, 1993) and refinements such as orthogonal matching pursuit (Tropp & Gilbert, 2007), stage-wise orthogonal matching pursuit (Donoho, Tsaig, Drori, & Stark, 2006), tree matching pursuit (Duarte, Wakin, & Baraniuk, 2005) and chaining pursuit (Gilbert, Strauss, Tropp, & Vershynin, 2006),
- Chirplets (see, e.g., O'Neill, Flandrin, & Karl, 2002; Cui, Wong, & Mann, 2005; Cui & Wong, 2006),
- Wave atoms (Demanet & Ying, 2007),
- Curvelets (Candès & Donoho, 2002)
- Sparsification by loopy belief propagation (Sarvotham, Baron, & Baraniuk, 2006)
- The Hilbert-Huang transform (Huang et al., 1998),
- Compressed sensing (Candès, Romberg, & Tao, 2006; Donoho, 2006).

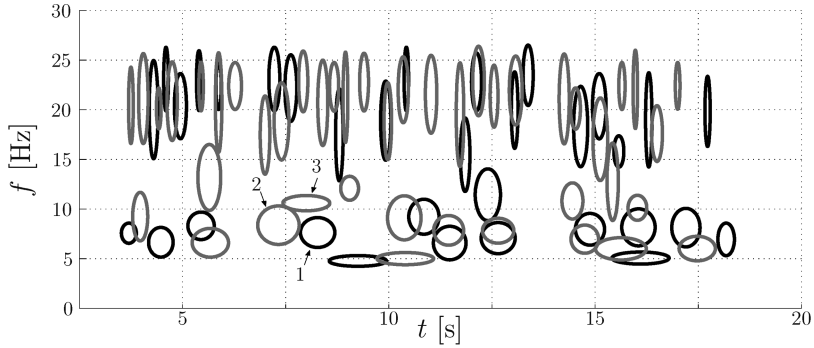
Moreover, the point processes may be defined in spaces other than the time-frequency plane; for example, they may occur in two-dimensional space (e.g., images), space frequency (e.g., wavelet image coding), or space time (e.g., movies), and they may also be defined on more complicated manifolds, such as curves and surfaces. Such extensions may straightforwardly be derived from the example of bump models. We consider several extensions in section 6.

Our extension of stochastic event synchrony to multidimensional point processes (and bump models in particular) is derived from the following observation (see Figure 2a): bumps in one time-frequency map may not be present in the other map ("noncoincident" bumps); other bumps are present in both maps ("coincident bumps") but appear at slightly different positions on the maps. The black lines in Figure 2b connect the centers of coincident bumps and hence visualize the offsets between pairs of coincident bumps.

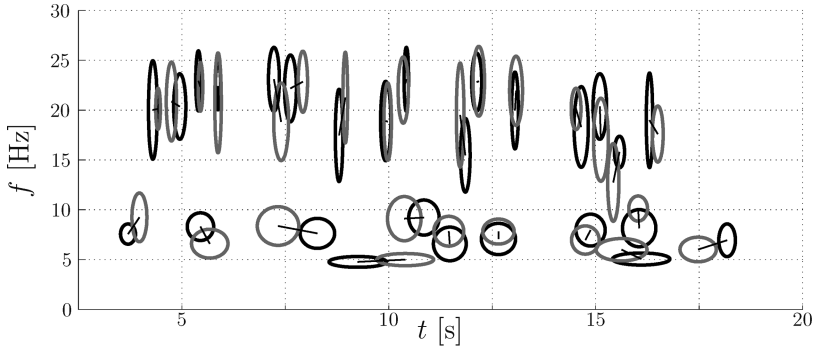
Such offsets jeopardize the suitability of classical similarity measures for time-frequency maps. For example, let us consider the Pearson correlation coefficient r between two time-frequency maps $x_1(t, f)$ and $x_2(t, f)$:

$$r = \frac{\sum_{t,f}(x_1(t, f) - \bar{x}_1)(x_2(t, f) - \bar{x}_2)}{\sqrt{\sum_{t,f}(x_1(t, f) - \bar{x}_1)^2}\sqrt{\sum_{t,f}(x_2(t, f) - \bar{x}_2)^2}}, \tag{2.1}$$

where $\bar{x}_i = \sum_{t,f} x_i(t, f)$ $(i = 1, 2)$. Note that r, like many other classical similarity measures, is based on pointwise comparisons. In other words, it compares the activity at instance (t, f) in map x_1 to the activity in x_2 at the same instance (t, f). Therefore, if the correlated activity in the maps



(a) Bump models of two EEG channels (one in black, the other in gray). One can observe pairs of bumps that are coincident ("matched"); other bumps are not overlapping and cannot be matched to bumps from the other bump model. Under the assumption that large frequency offsets between bumps are not likely to occur, bump 1 ($t = 8.2$ s) should be paired with bump 2 ($t = 7.4$ s) and not with 3 ($t = 8$ s), since the former is much closer in frequency than the latter. Such prior information may be incorporated by means of conjugate priors for s_t and s_f, that is, scaled inverse chi square distributions.



(b) Coincident bumps ($\rho = 27\%$). The black lines connect the centers of coincident bumps.

Figure 2: Coincident and noncoincident activity.

$x_1(t, f)$ and $x_2(t, f)$ is slightly delayed or a little shifted in frequency, the correlation coefficient r will be small, and as a result, it may not be able to capture the correlated activity. Our approach alleviates this shortcoming, since it explicitly handles delays and frequency offsets.

We quantify the interdependence between two bump models by five parameters: the parameters ρ, δ_t, and s_t introduced in the companion letter in this issue:

- ρ: fraction of noncoincident bumps
- δ_t: the average timing offset (delay) between coincident bumps
- s_t: the variance of the timing offset between coincident bumps,

in addition to:

- δ_f: the average frequency offset between coincident bumps
- s_f: the variance of the frequency offset between coincident bumps.

We determine those five parameters and the pairwise alignment of e and e' by statistical inference, as in the one-dimensional case (cf. sections 3 and 4 in the companion letter in this issue). We start by constructing a statistical model that captures the relation between the two bump models e and e'; that model contains the five SES parameters, besides variables related to the pairwise alignment of the bumps of e and e'. Next, we perform inference in that model, resulting in estimates for the SES parameters and the pairwise alignment. More concretely, we apply coordinate descent, as in the case of one-dimensional point processes. In the following section, we outline our statistical model. In section 4, we describe the factor graph of that model. From that factor graph, we derive the inference algorithm for multidimensional SES; in section 5, we outline that inference algorithm. We refer to appendix C for the detailed derivations. In section 6, we suggest various extensions of our statistical model.

3 Statistical Model

In this section, we explain the statistical model that forms the foundation of multidimensional SES. For reasons that we will explain, we represent the model in two different ways (see equations 3.7 and 3.15). For clarity, we list in Table 1 the most important variables and parameters that appear in those representations. We use the notation $\theta = (\delta_t, s_t, \delta_f, s_f)$, $\sigma_t = \sqrt{s_t}$, and $\sigma_f = \sqrt{s_f}$.

Figure 3 illustrates how we extend the generative procedure underlying one-dimensional SES (cf. the companion letter) to the time-frequency domain. As a first step, we generate a hidden point process v (dotted bumps in Figure 3). The number ℓ of bumps v_k is also now described by a geometric prior, in particular,

$$p(\ell) = (1 - \tilde{\lambda})\tilde{\lambda}^{\ell}, \tag{3.1}$$

with $\tilde{\lambda} = \lambda(t_{\max} - t_{\min})(f_{\max} - f_{\min}) \in (0, 1)$. (We motivate this choice of prior in the companion letter.) The centers $(\tilde{t}_k, \tilde{f}_k)$ of those bumps are

Table 1: Variables and Parameters Associated with Models $p(e, e', j, j', \theta)$ (Equation 3.7) and $p(e, e', b, b', c, \theta)$ (Equation 3.15).

Symbol	Explanation
e and e'	The two given bump models
t and t'	Occurrence time of the bumps of e and e'
f and f'	Frequencies of the bumps of e and e'
Δt and $\Delta t'$	Width of the bumps of e and e'
Δf and $\Delta f'$	Height of the bumps of e and e'
v	Hidden bump model from which the observed bump models e and e' are generated
$\tilde{e}$ and $\tilde{e}'$	Bump models obtained by shifting v over $(\delta_t/2, \delta_f/2)$ and $(-\delta_t/2, -\delta_f/2)$, resp., and randomly perturbing the timing and frequency of the resulting sequences (with variance $s_t/2$ and $s_f/2$, resp.)
b and b'	Binary sequences that indicate whether bumps in e and e', resp. are coincident
c	Binary sequence that indicates whether a particular bump in e is coincident with a particular bump in e', more precisely, $c_{kk'} = 1$ iff e_k is coincident with $e'_{k'}$ and is zero otherwise
j and j'	Indices of the coincident bumps in e and e', resp.
n and n'	Length of e and e', resp.
ℓ	Length of v
δ_t and δ_f	Average timing and frequency offset, resp., between e and e'
s_t and s_f	Timing and frequency jitter, resp., between e and e'

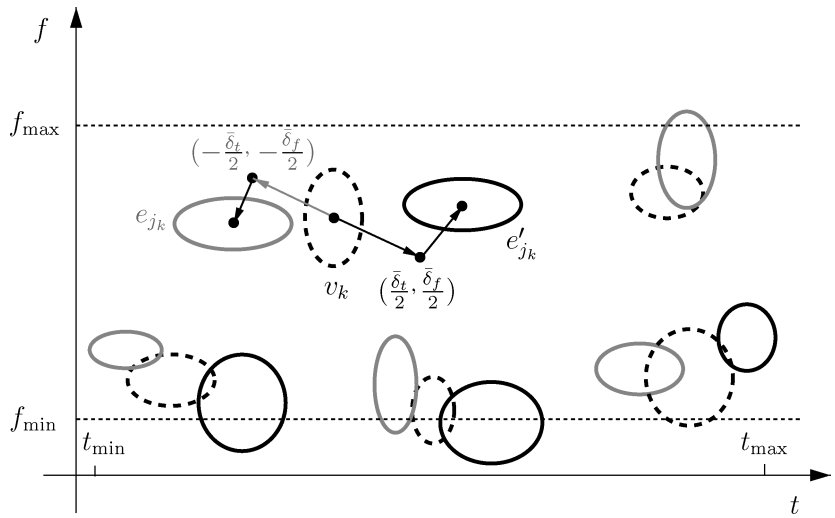


Figure 3: Generative model for e and e'. One first generates a hidden process v and then makes two identical copies of v and shifts those over $(-\delta_t/2, -\delta_f/2)$ and $(\delta_t/2, \delta_f/2)$, respectively. The events of the resulting point process are slightly shifted (with variance (s_t, s_f)), and some of those events are deleted (with probability p_d), resulting in e and e'.

placed uniformly within the rectangle $[t_{\min}, t_{\max}] \times [f_{\min}, f_{\max}]$, and as a consequence,

$$p(\tilde{t}, \tilde{f}\,|\ell) = \frac{1}{(t_{\max} - t_{\min})^{\ell}(f_{\max} - f_{\min})^{\ell}}. \tag{3.2}$$

The amplitudes, widths, and heights of the bumps v_k are independently and identically distributed according to priors p_w, $p_{\Delta t}$, and $p_{\Delta f}$, respectively. Next, from bump model v, we generate the bump models e and e', as follows:

1. Make two copies $\tilde{e}$ and $\tilde{e}'$ of bump model v.
2. Generate new amplitudes w_k, widths Δt_k, and heights Δf_k for the bumps $\tilde{e}_k$ by drawing (independent) samples from the priors p_w, $p_{\Delta t}$, and $p_{\Delta f}$, respectively. Likewise, generate new amplitudes w'_k and widths $\Delta t'_k$ and $\Delta f'_k$ for bumps $\tilde{e}'_k$.
3. Shift the bumps $\tilde{e}_k$ and $\tilde{e}'_k$ over $(-\frac{\bar{\delta}_t}{2}, -\frac{\bar{\delta}_f}{2})$, and $(\frac{\bar{\delta}_t}{2}, \frac{\bar{\delta}_f}{2})$, respectively, with:

$$\bar{\delta}_t = \delta_t\,(\Delta t_k + \Delta t'_k), \tag{3.3}$$

$$\bar{\delta}_f = \delta_f\,(\Delta f_k + \Delta f'_k). \tag{3.4}$$

4. Add small, random perturbations to the position of the bumps $\tilde{e}_k$ and $\tilde{e}'_k$ (cf. Figure 3), modeled as zero-mean gaussian random vectors with diagonal covariance matrix $\mathrm{diag}(\frac{\bar{s}_t}{2}, \frac{\bar{s}_f}{2})$:

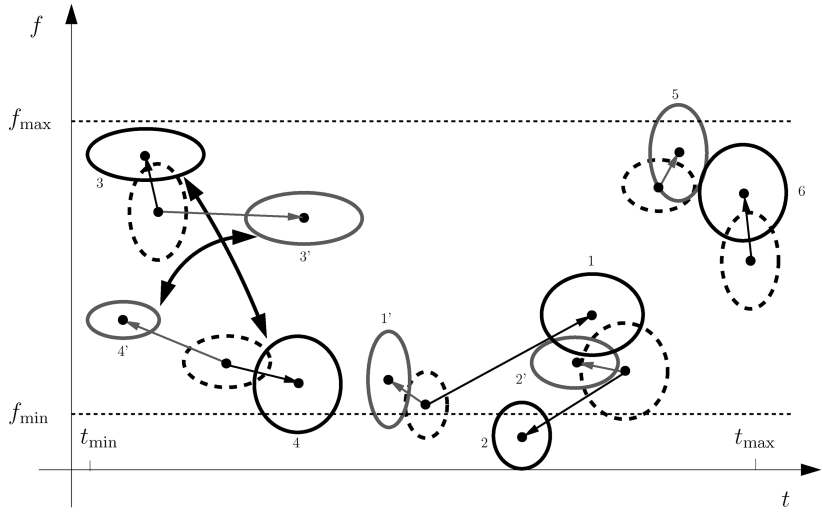
$$\bar{s}_t = s_t\,(\Delta t_k + \Delta t'_k)^2, \tag{3.5}$$

$$\bar{s}_f = s_f\,(\Delta f_k + \Delta f'_k)^2. \tag{3.6}$$

5. Randomly remove bumps from $\tilde{e}$ and $\tilde{e}'$. Each bump is deleted with probability p_d independent of the other bumps, resulting in the bump models e and e'.

As in the one-dimensional case, the above generative procedure (cf. Figure 3) may straightforwardly be extended from a pair of point processes e and e' to a collection of point processes, but inference in the resulting probabilistic model is intractable. We will present approximate inference algorithms in a future report.

Also in the multidimensional case, event synchrony is inherently ambiguous. Figure 4 shows two procedures to generate the same point processes e and e'. If s_t is large, with high probability, events in e and e' will not be ordered in time; for example, events (1,2) and (3',4') in Figure 4a are reversed in time. Ignoring this fact will result in estimates of s_t that are smaller than the true value s_t. The SES algorithm will probably correctly infer the coincident event pairs (3,3') and (4,4'), since those pairs are far apart

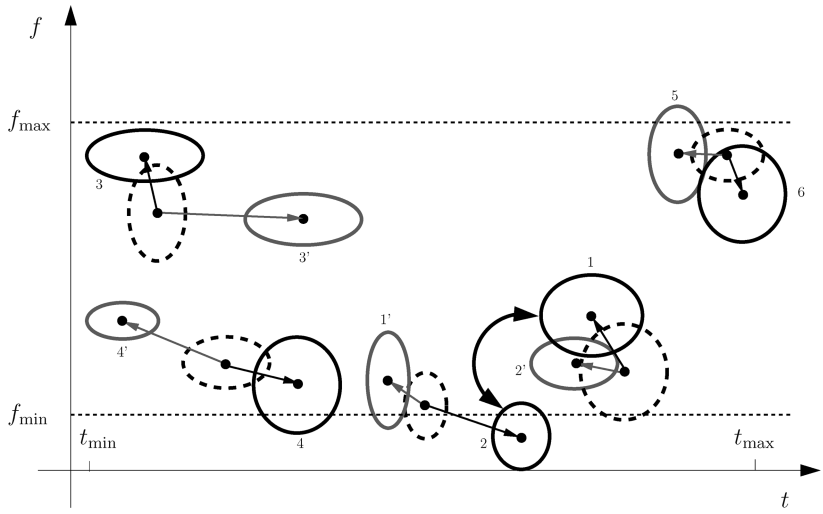


(a) A first procedure to generate e and e'. Interestingly, events 3 and 4' are closer in timing than 3 and 3' (and likewise 3' and 4 versus 3' and 3) but not in frequency, and therefore the multidimensional SES algorithm will treat (3,3') and (4,4') as coincident pairs. In other words, despite the reversal in timing (as indicated by the arrows), the events will be correctly grouped in pairs. Note that events 5 and 6 are noncoincident and that (1,1') and (2,2') are coincident event pairs.

Figure 4: Inherent ambiguity in event synchrony. Two equivalent procedures to generate the multidimensional point processes e and e'.

in frequency and therefore, event 3 is closer to event 3' than it is to event 4'. However, it will probably treat (1',2) and (1,2') as coincident event pairs instead of (1,2) and (1',2') (see Figure 4b), since event 1 is much closer to event 2' than event 2. As a consequence, SES will underestimate s_t. However, the bias will be smaller than in the one-dimensional case. The SES algorithm will incorrectly align pairs of events only if those pairs of events have about the same frequency (as events 1, 1', 2, and 2' in Figure 4); if those events are far apart in frequency (as events 3, 3', 4, and 4' in Figure 4), potential time reversals will be correctly inferred. This observation obviously carries over to any other kind of multidimensional point processes.

Besides timing reversals, some event deletions may be ignored. In Figure 4a, events 5 and 6 are noncoincident; however; in the procedure of Figure 4b, they are both coincident. The latter generative procedure is simpler in the sense that it involves fewer deletions and the perturbations are slightly smaller. As a result, the parameter ρ (and hence also p_d) is



(b) A second procedure to generate the same point processes e and e'. Event pairs (1',2) and (1,2') are now coincident or, equivalently, event 1 now plays the role of event 2 in Figure 4a (as indicated by the arrow). Since event 1 is much closer to event 2' than event 1', the SES inference algorithm will most probably prefer the alignment (1,2') and (1',2) instead of (1,1') and (2,2'). Note that (5,6) is now a coincident event pair. Since events 5 and 6 are close, the SES algorithm would consider them as coincident.

Figure 4: Continued.

generally underestimated. However, this bias will also be smaller than in the one-dimensional case. Indeed, if events 5 and 6 had strongly different frequencies, the SES algorithm would probably not treat them as a coincident pair. Obviously, this observation also extends to any kind of multidimensional point processes.

The generative procedure of Figure 3 leads to the two-dimensional extension of the one-dimensional SES model (cf. equation 3.25 in the companion letter):

$$\begin{aligned} p(e, e', j, j', \theta) = \gamma\, \beta^{n_{\text{non-co}}^{\text{tot}}}\, p(\delta_t)p(s_t)p(\delta_f)p(s_f) \prod_{k=1}^{n_{\text{co}}^{\text{tot}}} p_w(w_{j_k})p_w(w'_{j'_k}) \\ \cdot\, p_{\Delta t}(\Delta t_{j_k})p_{\Delta t}(\Delta t'_{j'_k})p_{\Delta f}(\Delta f_{j_k})p_{\Delta f}(\Delta f'_{j'_k}) \\ \cdot\, \mathcal{N}\big(t'_{j'_k} - t_{j_k}; \bar{\delta}_t, \bar{s}_t\big)\, \mathcal{N}\big(f'_{j'_k} - f_{j_k}; \bar{\delta}_f, \bar{s}_f\big), \end{aligned} \tag{3.7}$$

with $\bar{\delta}_t = \delta_t\,(\Delta t_{j_k} + \Delta t'_{j'_k})$, $\bar{\delta}_f = \delta_f\,(\Delta f_{j_k} + \Delta f'_{j'_k})$, $\bar{s}_t = s_t\,(\Delta t_{j_k} + \Delta t'_{j'_k})^2$, $\bar{s}_f = s_f\,(\Delta f_{j_k} + \Delta f'_{j'_k})^2$, and where the constant β is again given by

$$\beta = p_d\,\sqrt{\lambda}, \tag{3.8}$$

and

$$\gamma = \left(\sqrt{\lambda}\,(1 - p_d)\right)^{n+n'}(1 - \tilde{\lambda})^{\frac{1}{1-p_d^2\,\tilde{\lambda}}}, \tag{3.9}$$

with $\tilde{\lambda} = \lambda(t_{\max} - t_{\min})(f_{\max} - f_{\min})$. In model 3.7, the priors p_w, $p_{\Delta t}$, and $p_{\Delta f}$ for the bump parameters w_{j_k}, $w'_{j'_k}$, Δt_{j_k}, $\Delta t'_{j'_k}$, Δf_{j_k}, and $\Delta f'_{j'_k}$, respectively, are irrelevant for what follows; we therefore discard them now. We also discard the constant γ in equation 3.7, since it does not depend on j and j'; as a consequence, it is not relevant for determining j, j' and the SES parameters. The parameter β, however, clearly affects the inference of j, j' and the SES parameters, since the exponent of β in equation 3.7 does depend on j and j'. We elaborate on the priors of the parameters $\theta = (\delta_t, s_t, \delta_f, s_f)$ later.

As in the one-dimensional case, it is instructive to consider the negative logarithm of equation 3.7:

$$\begin{aligned} -\log p(e, e',\, j,\, j', \theta) = \sum_{k=1}^{n_{\text{co}}^{\text{tot}}} \Bigg[& \frac{(t'_{j'_k} - t_{j_k} - \delta_t)^2}{2s_t(\Delta t_{j_k} + \Delta t'_{j'_k})^2} + \frac{(f'_{j'_k} - f_{j_k} - \delta_f)^2}{2s_f(\Delta f_{j_k} + \Delta f'_{j'_k})^2} \\ & + \frac{1}{4}\log 4\pi^2 s_t(\Delta t_{j_k} + \Delta t'_{j'_k})^2 s_f(\Delta f_{j_k} + \Delta f'_{j'_k})^2 \Bigg] \\ & - n_{\text{non-co}}^{\text{tot}} \log \beta - \log p(\delta_t)p(s_t)p(\delta_f)p(s_f) + \zeta, \end{aligned} \tag{3.10}$$

where ζ is an irrelevant constant. Expression 3.10 may be considered a cost function, along the lines of the one-dimensional case. The unit cost $d(s_t)$ associated with each noncoincident event equals

$$d(s_t) = -\log \beta. \tag{3.11}$$

The unit cost $d(e_{j_k}, e'_{j'_k})$ of each event pair $(e_{j_k}, e'_{j'_k})$ is given by

$$\begin{aligned} d(e_{j_k}, e'_{j'_k}) = & \frac{(t'_{j'_k} - t_{j_k} - \delta_t)^2}{2s_t(\Delta t_{j_k} + \Delta t'_{j'_k})^2} + \frac{(f'_{j'_k} - f_{j_k} - \delta_f)^2}{2s_f(\Delta f_{j_k} + \Delta f'_{j'_k})^2} \\ & + \frac{1}{4}\log\left(4\pi^2 s_t(\Delta t_{j_k} + \Delta t'_{j'_k})^2 s_f(\Delta f_{j_k} + \Delta f'_{j'_k})^2\right). \end{aligned} \tag{3.12}$$

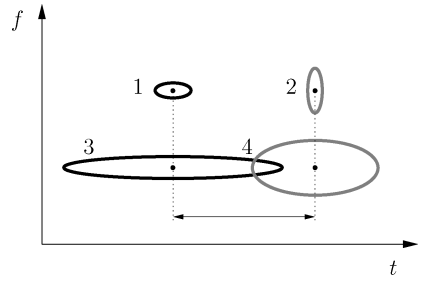


Figure 5: Bumps 1 and 2 are farther apart than 3 and 4, although the distance between their centers is identical. Therefore, in order to quantify the distance between bumps, it is necessary to normalize the distance between the bump centers by the bump widths (cf. equation 3.10).

Interestingly, the first two terms in equation 3.12 may be viewed as a Euclidean distance. Since the point processes e and e' of Figure 1 are defined on the time-frequency plane, the Euclidean distance is indeed a natural metric. Note that the Euclidean distance is normalized: the timing and frequency offsets are normalized by the bump width and height, due to the particular choices 3.3 to 3.7. Figure 5 explains why normalization is crucial.

In the one-dimensional model proposed in the companion letter, the third term in equation 3.12 is absorbed into the unit cost $d(s_t)$. In the multidimensional case, however, that is no longer possible. That term depends on the width and height of the two events e_{j_k} and $e'_{j'_k}$ and cannot be decomposed in a term that depends on only the parameters of e_{j_k} and a second term that depends on only the parameters of $e'_{j'_k}$. In other words, the third term in equation 3.12 cannot be interpreted as unit costs of single events.

In our model of one-dimensional SES, we did not consider the width of events. Instead we incorporated only the occurrence time, since that suffices for most common applications. However, the model could easily be extended to include event widths if necessary.

We wish to point out that the unit costs $d(s_t)$, equation 3.11, and $d(e_{j_k}, e'_{j'_k})$, equation 3.12, are in general not dimensionless; the total cost, equation 3.10, however, is dimensionless in the sense that a change in units will affect the total cost by a constant, irrelevant term. In the one-dimensional case considered in the companion letter, the unit costs are dimensionless, since the third term in equation 3.12 is absorbed into the unit cost $d(s_t)$.

In principle, one may determine the sequences j and j' and the parameters θ by coordinate descent along the lines of the algorithm of one-dimensional SES. In the multidimensional case, however, the alignment cannot be solved by the Viterbi algorithm (or equivalently, the max-product algorithm applied on a cycle-free factor graph of model 3.7). One needs to

allow timing reversals (see Figure 4); therefore, the indices j_k and $j'_{k'}$ are no longer necessarily monotonically increasing, and as a consequence, the state space becomes substantially larger.

Instead of applying the max-product algorithm on a cycle-free factor graph of model 3.7, we apply that algorithm on a cyclic factor graph, which will amount to a practical procedure to obtain pairwise alignments of multidimensional point processes (and bump models in particular). We will show that it finds the optimal solution under very mild conditions. In order to derive this procedure, we introduce a parameterization of model 3.7 that is naturally represented by a cyclic graph. For each pair of events e_k and $e'_{k'}$, we introduce a binary variable $c_{kk'}$ that equals one if e_k and $e'_{k'}$ form a pair of coincident events and is zero otherwise. Since each event in e is associated at most with one event in e', we have the constraints

$$\sum_{k'=1}^{n'} c_{1k'} \triangleq s_1 \in \{0,1\}, \sum_{k'=1}^{n'} c_{2k'} \triangleq s_2 \in \{0,1\}, \ldots, \sum_{k'=1}^{n'} c_{nk'} \triangleq s_n \in \{0,1\}. \tag{3.13}$$

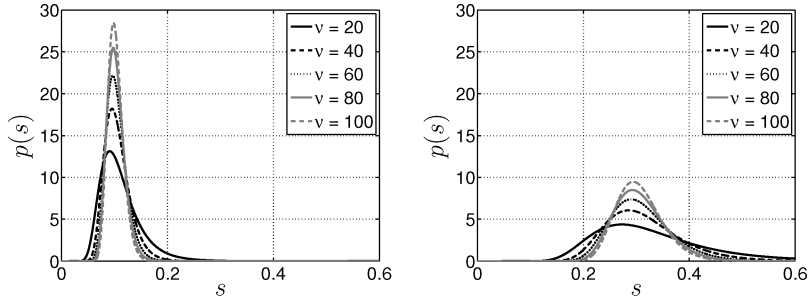
Similarly, each event in e' is associated at most with one event in e, which may be expressed by a similar set of constraints. The sequences s and s' are related to the sequences b and b' (cf. the companion letter) as follows:

$$b_k = 1 - s_k \qquad \text{and} \qquad b'_k = 1 - s'_k. \tag{3.14}$$

From the variables $c_{kk'}$ (with $k = 1, \ldots, n$ and $k' = 1, \ldots, n'$), one can also easily determine the sequences j and j'. Indeed, if $c_{kk'} = 1$, the index k and k' appear in j and j', respectively.

In this representation, the global statistical model 3.7 can be cast as

$$\begin{aligned} p(e, e', b, b', c, \theta) \propto & \prod_{k=1}^{n} (\beta\, \delta[b_k - 1] + \delta[b_k]) \prod_{k'=1}^{n'} (\beta\, \delta[b'_k - 1] + \delta[b'_k]) \\ & \cdot \prod_{k=1}^{n} \prod_{k'=1}^{n'} \Big(\mathcal{N}\big(t'_{k'} - t_k; \bar{\delta}_t, \bar{s}_t\big)\, \mathcal{N}\big(f'_{k'} - f_k; \bar{\delta}_f, \bar{s}_f\big) \Big)^{c_{kk'}} \\ & \cdot p(\delta_t) p(s_t) p(\delta_f) p(s_f) \prod_{k=1}^{n} \left(\delta\left[b_k + \sum_{k'=1}^{n'} c_{kk'} - 1 \right] \right) \\ & \cdot \prod_{k'=1}^{n'} \left(\delta[b'_{k'} + \sum_{k=1}^{n} c_{kk'} - 1] \right), \end{aligned} \tag{3.15}$$



(a) $s_0 = 0.1$. (b) $s_0 = 0.3$.

Figure 6: Scaled inverse chi square distributions for various values of the degrees of freedom ν and widths s_0; $s_0 = 0.1$ (left) and 0.3 (right).

where $\delta[\cdot]$ is the Kronecker delta; the variables $c_{kk'}$, b_k, and $b_{k'}$ are binary; and $\bar{\delta}_t = \delta_t\,(\Delta t_k + \Delta t'_{k'})$, $\bar{\delta}_f = \delta_f\,(\Delta f_k + \Delta f'_{k'})$, $\bar{s}_t = s_t\,(\Delta t_k + \Delta t'_{k'})^2$, $\bar{s}_f = s_f\,(\Delta f_k + \Delta f'_{k'})^2$. The last two factors in equation 3.15 encode the expressions 3.14.

We now comment on the priors of the parameters $\theta = (\delta_t, s_t, \delta_f, s_f)$. Since we (usually) do not need to encode prior information about δ_t and δ_f, we choose improper priors $p(\delta_t) = 1 = p(\delta_f)$. On the other hand, one may have prior knowledge about s_t and s_f. For example, in the case of spontaneous EEG (see section 8), we a priori expect the frequency jitter s_f to be small. Frequency shifts can be caused only by nonlinear transformations, which are hard to justify from a physiological perspective. Therefore, we expect bumps to appear at about the same frequency in both time-frequency maps. The timing jitter s_t may be larger, since signals often propagate over significant distances in the brain, and therefore, timing jitter arises quite naturally. For example, bump 1 ($t = 8.2$ s) should be paired with bump 2 ($t = 7.4$ s) and not with bump 3 ($t = 8$ s), since the former is much closer in frequency than the latter. One may encode such prior information by means of conjugate priors for s_t and s_f, that is, scaled inverse chi square distributions:

$$p(s_t) = \frac{(s_{0,t}\nu_t/2)^{\nu_t/2}}{\Gamma(\nu_t/2)} \frac{e^{-\nu_t s_{0,t}/2s_t}}{s_t^{1+\nu_t/2}} \tag{3.16}$$

$$p(s_f) = \frac{(s_{0,f}\nu_f/2)^{\nu_f/2}}{\Gamma(\nu_f/2)} \frac{e^{-\nu_f s_{0,f}/2s_f}}{s_f^{1+\nu_f/2}}, \tag{3.17}$$

where ν_t and ν_f are the degrees of freedom and $\Gamma(x)$ is the gamma function. In the example of spontaneous EEG, the widths $s_{0,t}$ and $s_{0,f}$ are chosen such that $s_{0,t} > s_{0,f}$, since s_f is expected to be smaller than s_t. Figure 6 shows the scaled inverse chi square distribution with $\nu = 20, 40, \ldots, 100$ and $s_0 = 0.1$ and 0.3.

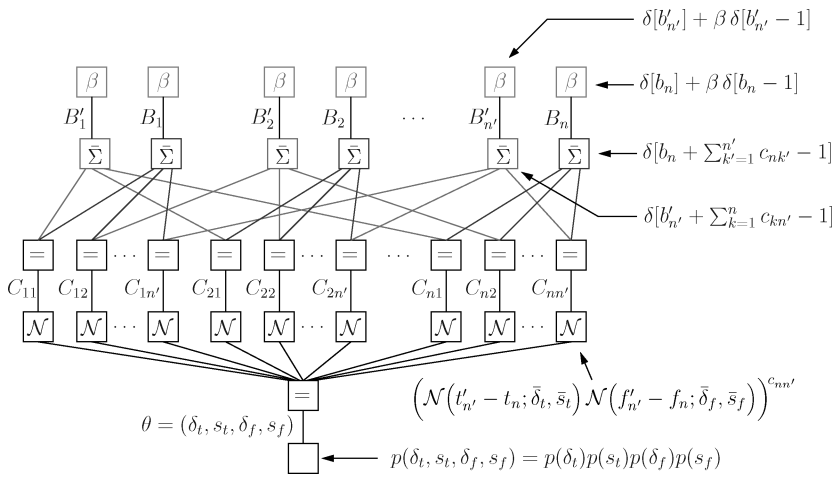


Figure 7: Factor graph of model 3.15. Each edge represents a variable, and each node corresponds to a factor of equation 3.15, as indicated by the arrows at the right-hand side. More details on factor graphs can be found in appendix A.

4 Factor Graph

To perform inference in model 3.15, we use a factor graph of that model (see Figure 7); each edge represents a variable, and each node corresponds to a factor of equation 3.15, as indicated by the arrows at the right-hand side (refer to appendix A for an introduction to factor graphs). We omitted the edges for the (observed) variables t_k, $t'_{k'}$, f_k, $f'_{k'}$, w_k, $w'_{k'}$, Δt_k, $\Delta t'_{k'}$, Δf_k, and $\Delta f'_{k'}$ in order not to clutter the figure. In the following, we discuss the nodes in Figure 7 (from top to bottom):

- The nodes denoted by β correspond to the factors $(\beta\delta[b_k - 1] + \delta[b_k])$ and $(\beta\delta[b'_k - 1] + \delta[b'_k])$.
- The nodes denoted by $\bar{\Sigma}$ represent the factors $\big(\delta[b_k + \sum_{k'=1}^{n'} c_{kk'} - 1]\big)$ and $\big(\delta[b'_{k'} + \sum_{k=1}^{n} c_{kk'} - 1]\big)$. (See also row 4 in Table C.1.)
- The equality constraint nodes (marked by "=") enforce the equality of the incident variables. (See also row 3 in Table C.1.)
- The nodes $\mathcal{N}$ corresponds to the gaussian distributions in equation 3.15. More precisely, they correspond to the factors

$$g_{\mathcal{N}}(c_{kk'}; \theta) = \Big(\mathcal{N}\big(t'_{k'} - t_k; \bar{\delta}_t, \bar{s}_t\big)\,\mathcal{N}\big(f'_{k'} - f_k; \bar{\delta}_f, \bar{s}_f\big)\Big)^{c_{kk'}}, \tag{4.1}$$

where $\bar{\delta}_t = \delta_t\,(\Delta t_k + \Delta t'_{k'})$, $\bar{\delta}_f = \delta_f\,(\Delta f_k + \Delta f'_{k'})$, $\bar{s}_t = s_t\,(\Delta t_k + \Delta t'_{k'})^2$, $\bar{s}_f = s_f\,(\Delta f_k + \Delta f'_{k'})^2$.

- The bottom node stands for the prior:

$$p(\theta) = p(\delta_t, s_t, \delta_f, s_f) = p(\delta_t)p(s_t)p(\delta_f)p(s_f). \tag{4.2}$$

5 Statistical Inference

We determine the alignment $c = (c_{11}, c_{12}, \ldots, c_{nn'})$ and the parameters $\theta = (\delta_t, s_t, \delta_f, s_f)$ by maximum a posteriori (MAP) estimation:

$$(\hat{c}, \hat{\theta}) = \underset{c,\theta}{\operatorname{argmax}}\, p(e, e', c, \theta), \tag{5.1}$$

where $p(e, e', c, \theta)$ is obtained from $p(e, e', b, b', c, \theta)$, equation 3.15, by marginalizing over b and b':

$$\begin{aligned} p(e, e', c, \theta) \propto & \prod_{k=1}^{n} \left(\beta\delta\left[\sum_{k'=1}^{n'} c_{kk'} \right] + \delta\left[\sum_{k'=1}^{n'} c_{kk'} - 1 \right] \right) \\ & \cdot \prod_{k'=1}^{n'} \left(\beta\delta\left[\sum_{k=1}^{n} c_{kk'} \right] + \delta\left[\sum_{k=1}^{n} c_{kk'} - 1 \right] \right) \\ & \cdot \prod_{k=1}^{n} \prod_{k'=1}^{n'} \left(\mathcal{N}\left(t'_{k'} - t_k; \bar{\delta}_t, \bar{s}_t\right) \mathcal{N}\left(f'_{k'} - f_k; \bar{\delta}_f, \bar{s}_f\right) \right)^{c_{kk'}} \\ & \cdot p(\delta_t)p(s_t)p(\delta_f)p(s_f). \end{aligned} \tag{5.2}$$

From $\hat{c}$, we obtain the estimate $\hat{\rho}$ as

$$\hat{\rho} = \frac{n + n' - 2\sum_{k=1}^{n}\sum_{k'=1}^{n'} \hat{c}_{kk'}}{n + n'} = \frac{\sum_{k=1}^{n} \hat{b}_k + \sum_{k=1}^{n'} \hat{b}'_{k'}}{n + n'}. \tag{5.3}$$

The MAP estimate, equation 5.1 is intractable, and we try to obtain it by coordinate descent. First, the parameters θ are initialized (e.g., $\hat{\delta}_t^{(0)} = 0 = \delta_f^{(0)}$, $\hat{s}_t^{(0)} = s_{0,t}$, and $\hat{s}_f^{(0)} = s_{0,f}$); then one alternates the following two update rules until convergence (or until the available time has elapsed):

$$\hat{c}^{(i+1)} = \underset{c}{\operatorname{argmax}}\, p(e, e', c, \hat{\theta}^{(i)}) \tag{5.4}$$

$$\hat{\theta}^{(i+1)} = \underset{\theta}{\operatorname{argmax}}\, p(e, e', \hat{c}^{(i+1)}, \theta). \tag{5.5}$$

The estimate $\hat{\theta}^{(i+1)}$, equation 5.5, is available in closed form, as we show in appendix C. In that appendix, we also show that the alignment 5.4 is equivalent to a classical problem in combinatorial optimization known as max-weight bipartite matching (see, e.g., Gerards, 1995; Pulleyblank, 1995; Bayati et al., 2005; Bayati, Borgs, Chayes, & Zecchina, 2008; Huang & Jebara,

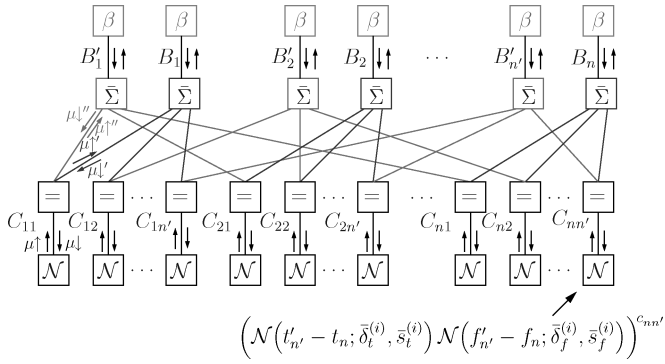


Figure 8: Max-product message passing. The messages indicate the max-product messages, computed according to the max-product update rule (see appendix C).

2007; Sanghavi, 2007, 2008). There are many ways to solve that problem, and we describe one of them, the max-product algorithm, in more detail since it is arguably the simplest approach. That algorithm can be derived by means of the graph of Figure 7. It operates by sending information ("messages") along the edges of that graph, as illustrated in Figure 8. The messages, depicted by arrows, contain (probabilistic) information about which pairs of bumps are coincident and which are not; they are computed according to a generic rule (i.e., the max-product rule). Intuitively, the nodes may be viewed as computing elements that iteratively update their opinion about the bump matching, based on the opinions (messages) they receive from neighboring nodes. When the max-product algorithm eventually has converged and the nodes have found a "consensus," the messages are combined to obtain a decision on c, b, and b' and an estimate of ρ. (In appendix C, we derive the algorithm 5.4 and 5.5, in detail; it is summarized in Table 2. Matlab code of the algorithm is available online at http://www.dauwels.com/SESToolbox.html.)

The computational complexity of this algorithm is in principle proportional to nn', that is, the product of both sequence lengths. If one excludes pairs of events that are too far apart on the time-frequency map, one obtains an algorithm with linear complexity (as in the one-dimensional case).

The fixed points of the algorithm can be characterized as follows: the fixed point $\hat{\theta}$ is a stationary point of equation 5.2, and the alignment $\hat{c}$ is "neighborhood maximum," that is, the posterior probability, equation 5.2, of $\hat{c}$ is guaranteed to be greater than all other assignments in region around that assignment $\hat{c}$ (Freeman & Weiss, 1999).

The algorithm is an instance of coordinate descent and is therefore guaranteed to converge if the conditional maximizations 5.4 and 5.5 have unique solutions (Bezdek & Hathaway, 2002; Bezdek, Hathaway, Howard, Wilson, & Windham, 1987). The conditional maximization 5.5 always has a unique

Table 2: Inference Algorithm for Multidimensional SES.

Input
Models e and e', parameters β, ν_t, ν_f, $s_{0,t}$, $s_{0,f}$, $\hat{\delta}_t^{(0)}$, $\hat{\delta}_f^{(0)}$, $\hat{s}_t^{(0)}$, and $\hat{s}_f^{(0)}$
Algorithm
Iterate the following two steps until convergence:

(1) Update the alignment $\hat{c}$ by max-product message passing
Initialize messages $\mu{\downarrow}'(c_{kk'}) = 1 = \mu{\downarrow}''(c_{kk'})$
Iterate until convergence:
a. Upward messages:

$$\mu{\uparrow}'(c_{kk'}) \propto \mu{\downarrow}''(c_{kk'}) g_{\mathcal{N}}(c_{kk'}; \hat{\theta}^{(i)})$$
$$\mu{\uparrow}''(c_{kk'}) \propto \mu{\downarrow}'(c_{kk'}) g_{\mathcal{N}}(c_{kk'}; \hat{\theta}^{(i)}),$$

where

$$g_{\mathcal{N}}(c_{kk'}; \hat{\theta}^{(i)}) = \left(\mathcal{N}\left(t'_{k'} - t_k; \bar{\delta}_t^{(i)}, \bar{s}_t^{(i)}\right) \mathcal{N}\left(f'_{k'} - f_k; \bar{\delta}_f^{(i)}, \bar{s}_f^{(i)}\right) \right)^{c_{kk'}},$$

with $\bar{\delta}_t^{(i)} = \hat{\delta}_t^{(i)}(\Delta t_k + \Delta t'_{k'})$, $\bar{\delta}_f^{(i)} = \hat{\delta}_f^{(i)}(\Delta f_k + \Delta f'_{k'})$, $\bar{s}_t^{(i)} = \hat{s}_t^{(i)}(\Delta t_k + \Delta t'_{k'})^2$, $\bar{s}_f^{(i)} = \hat{s}_f^{(i)}(\Delta f_k + \Delta f'_{k'})^2$

b. Downward messages:

$$\begin{pmatrix} \mu{\downarrow}'(c_{kk'} = 0) \\ \mu{\downarrow}'(c_{kk'} = 1) \end{pmatrix} \propto \begin{pmatrix} \max\left(\beta, \max_{\ell' \neq k'} \mu{\uparrow}'(c_{k\ell'} = 1)/\mu{\uparrow}'(c_{k\ell'} = 0)\right) \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} \mu{\downarrow}''(c_{kk'} = 0) \\ \mu{\downarrow}''(c_{kk'} = 1) \end{pmatrix} \propto \begin{pmatrix} \max\left(\beta, \max_{\ell \neq k} \mu{\uparrow}''(c_{\ell k'} = 1)/\mu{\uparrow}''(c_{\ell k'} = 0)\right) \\ 1 \end{pmatrix}$$

Compute marginals $p(c_{kk'}) \propto \mu{\downarrow}'(c_{kk'}) \mu{\downarrow}''(c_{kk'}) g_{\mathcal{N}}(c_{kk'}; \hat{\theta}^{(i)})$
Compute decisions $\hat{c}_{kk'} = \operatorname{argmax}_{c_{kk'}} p(c_{kk'})$

(2) Update the SES parameters:

$$\hat{\delta}_t^{(i+1)} = \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \frac{\hat{t}_k'^{(i+1)} - \hat{t}_k^{(i+1)}}{\Delta \hat{t}_k^{(i+1)} + \Delta \hat{t}_k'^{(i+1)}}$$

$$\hat{\delta}_f^{(i+1)} = \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \frac{\hat{f}_k'^{(i+1)} - \hat{f}_k^{(i+1)}}{\Delta \hat{f}_k^{(i+1)} + \Delta \hat{f}_k'^{(i+1)}}$$

$$\hat{s}_t^{(i+1)} = \frac{\nu_t s_{0,t} + n^{(i+1)} \hat{s}_{t,\text{sample}}^{(i+1)}}{\nu_t + n^{(i+1)} + 2}$$

$$\hat{s}_f^{(i+1)} = \frac{\nu_f s_{0,f} + n^{(i+1)} \hat{s}_{f,\text{sample}}^{(i+1)}}{\nu_f + n^{(i+1)} + 2},$$

Output
Alignment $\hat{c}$ and SES parameters $\hat{\rho}$, $\hat{\delta}_t$, $\hat{\delta}_f$, $\hat{s}_t$, $\hat{s}_f$

solution (cf. equation C.1–C.4). If the alignment 5.4 has a unique solution, the max-product algorithm is guaranteed to find that unique optimum (Bayati et al., 2005; Huang & Jebara, 2007; Bayati et al., 2008; Sanghavi, 2007, 2008). Therefore, as long as equation 5.4 has a unique solution, the algorithm of Table 2 is guaranteed to converge. In many applications, the optimum of equation 5.4 is unique with probability one, and as a consequence, the proposed algorithm converges. We provide numerical results in section 8.

6 Extensions

So far, we have developed multidimensional SES for the particular example of bump models in the time-frequency domain. Here we consider alternative SES models in the time-frequency domain and in other domains. Those models are straightforward extensions of equation 3.7. We also outline how the SES inference algorithm can be modified accordingly.

6.1 Bump Parameters. One may incorporate differences in amplitude, width, and height between the bumps of e and e' in model 3.7. In the generative process of Figure 3, those parameters are then no longer drawn independently from certain prior distributions; instead they are obtained by perturbing the parameters of the hidden bump model v. In particular, one may again consider gaussian perturbations, as for the timing t, t' and frequency f, f' of the bumps. This leads to additional parameters δ_w, $\delta_{\Delta t}$, $\delta_{\Delta f}$, s_w, $s_{\Delta t}$, and $s_{\Delta f}$, which stand for the average offset and jitter between the bump amplitudes, widths, and heights of e and e', respectively; it also leads to additional gaussian factors in model 3.7. Moreover, priors for those additional parameters can be included in that model, leading to the expression

$$\begin{aligned} p(e, e', j, j', \theta) = \gamma\, \beta^{n^{\text{tot}}_{\text{non-co}}}\, & p(\delta_t)p(s_t)p(\delta_f)p(s_f)p(\delta_w)p(s_w)p(\delta_{\Delta t})p(s_{\Delta t}) \\ & \cdot p(\delta_{\Delta f})p(s_{\Delta f}) \cdot \mathcal{N}\big(t'_{j'_k} - t_{j_k}; \bar{\delta}_t, \bar{s}_t\big)\, \mathcal{N}\big(f'_{j'_k} - f_{j_k}; \bar{\delta}_f, \bar{s}_f\big) \\ & \cdot \mathcal{N}\big(w'_{j'_k} - w_{j_k}; \delta_w, s_w\big) \cdot \mathcal{N}\big(\Delta t'_{j'_k} - \Delta t_{j_k}; \delta_{\Delta_t}, s_{\Delta_t}\big) \\ & \cdot \mathcal{N}\big(\Delta f'_{j'_k} - \Delta f_{j_k}; \delta_{\Delta_f}, s_{\Delta_f}\big), \end{aligned} \tag{6.1}$$

where the parameters β and γ are again given by equations 3.8 and 3.9, respectively.

The inference algorithm for this model is very similar to the one of model 3.7 (cf. Table 2). The additional parameters δ_w, $\delta_{\Delta t}$, $\delta_{\Delta f}$, s_w, $s_{\Delta t}$, and $s_{\Delta f}$ are updated similarly as δ_t, δ_f, s_t, and s_f. The alignment procedure is almost identical, one merely needs to modify the upward message $g_{\mathcal{N}}$ (see step 1 in table 2). Besides the gaussian factors for the timing and

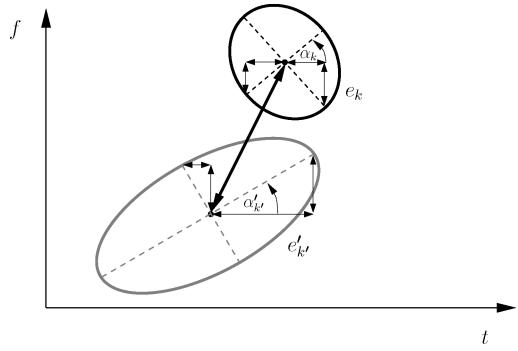


Figure 9: Two oblique bumps e_k and $e'_{k'}$ with rotation angle α_k and $\alpha'_{k'}$.

frequency offsets, that message contains similar factors for the offsets in bump amplitude, width, and height (cf. equation 6.1).

6.2 Oblique Bumps. Alternatively, one may consider oblique bumps—those that are not necessarily parallel to the time and frequency axes (see Figure 9). Such bumps correspond to chirps (see, e.g., O'Neill et al., 2002; Cui et al., 2005; Cui & Wong, 2006). The rotation angle of each bump e_k and $e'_{k'}$ is denoted by α_k and $\alpha'_{k'}$, respectively (with $\alpha_k, \alpha'_{k'} \in [0, \pi/2]$ for all k and k'). Model 3.7 can take those rotations into account. First, the normalization of the timing and frequency offsets needs to be modified accordingly. Second, one may wish to incorporate the difference in rotation angles α_k and $\alpha'_{k'}$ of bump e_k and $e'_{k'}$, respectively. In the generative process of Figure 3, one may include gaussian perturbations for the rotation angles. This leads to the following extension of model 3.7:

$$\begin{aligned} p(e, e', j, j', \theta) &= \gamma\, \beta^{n_{\text{non-co}}^{\text{tot}}}\, p(\delta_t) p(s_t) p(\delta_f) p(s_f) p(\delta_\alpha) p(s_\alpha) \\ &\cdot \prod_{k=1}^{n_{\text{co}}^{\text{tot}}} p_w(w_{j_k}) p_w(w'_{j'_k}) p_{\Delta t}(\Delta t_{j_k}) p_{\Delta t}(\Delta t'_{j'_k}) p_{\Delta f}(\Delta f_{j_k}) p_{\Delta f}(\Delta f'_{j'_k}) \\ &\cdot \mathcal{N}\big(t'_{j'_k} - t_{j_k}; \bar{\delta}_t, \bar{s}_t\big)\, \mathcal{N}\big(f'_{j'_k} - f_{j_k}; \bar{\delta}_f, \bar{s}_f\big)\, \mathcal{N}\big(\alpha'_{j'_k} - \alpha_{j_k}; \delta_\alpha, s_\alpha\big), \end{aligned} \tag{6.2}$$

where δ_α and s_α are the average offset and jitter, respectively, between the rotation angles. The parameters β and γ are again given by equations 3.8 and 3.9, respectively, $\bar{\delta}_t = \delta_t\,(\tilde{\Delta} t_{j_k} + \tilde{\Delta} t'_{j'_k})$, $\bar{\delta}_f = \delta_f\,(\tilde{\Delta} f_{j_k} + \tilde{\Delta} f'_{j'_k})$, $\bar{s}_t = s_t\,(\tilde{\Delta} t_{j_k} + \tilde{\Delta} t'_{j'_k})^2$, $\bar{s}_f = s_f\,(\tilde{\Delta} f_{j_k} + \tilde{\Delta} f'_{j'_k})^2$, with

$$\tilde{\Delta} t_{j_k} = \cos\alpha_k\, \Delta t_{j_k} + \sin\alpha_k\, \Delta f_{j_k} \tag{6.3}$$

$$\tilde{\Delta} f_{j_k} = \sin\alpha_k\, \Delta t_{j_k} + \cos\alpha_k\, \Delta f_{j_k} \tag{6.4}$$

$$\tilde{\Delta} t'_{j'_k} = \cos \alpha'_{k'} \, \Delta t'_{j'_k} + \sin \alpha'_{k'} \, \Delta f'_{j'_k} \tag{6.5}$$

$$\tilde{\Delta} f'_{j'_k} = \sin \alpha'_{k'} \, \Delta t'_{j'_k} + \cos \alpha'_{k'} \, \Delta f'_{j'_k}. \tag{6.6}$$

Note that model 6.2 reduces to model 3.7 if $\alpha_k = 0 = \alpha'_{k'}$ for all k and k'. Model 6.2 does not incorporate differences in the amplitude, width, and height of the bumps, but it could easily be extended if necessary.

In order to extend the SES algorithm of Table 2 to oblique bumps, three modifications are needed:

- In the upward message $g_{\mathcal{N}}$ (step 1 in Table 2) and in the update of the SES parameters (step 2), the parameters Δt_k, $\Delta t'_{k'}$, Δf_k, and $\Delta f'_{k'}$ are replaced by $\tilde{\Delta} t_k$, $\tilde{\Delta} t'_{k'}$, $\tilde{\Delta} f_k$, and $\tilde{\Delta} f'_{k'}$.
- If the prior on the parameters δ_α and s_α is the improper prior $p(\delta_\alpha) = 1 = p(s_\alpha)$, those parameters are updated similarly as δ_t and s_t:

$$\hat{\delta}_\alpha^{(i+1)} = \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \hat{\alpha}_k'^{(i+1)} - \hat{\alpha}_k^{(i+1)} \tag{6.7}$$

$$\hat{s}_\alpha^{(i+1)} = \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \left(\hat{\alpha}_k'^{(i+1)} - \hat{\alpha}_k^{(i+1)} - \hat{\delta}_\alpha^{(i+1)} \right)^2. \tag{6.8}$$

  Since $\alpha_k, \alpha'_k \in [0, \pi/2]$, one can simply add the angle differences.
- The alignment procedure is again almost identical: the upward message $g_{\mathcal{N}}$ (step 1 in Table 2) contains, in addition to the gaussian factors for the timing and frequency offsets, a similar factor for the offsets in bump rotation angle (cf. equation 6.2).

6.3 Point Processes in Other Spaces.

6.3.1 Euclidean Spaces. Until now we have considered bump models in the time-frequency domain. However, the statistical model 3.7 directly applies to point processes in other domains, for example, three-dimensional space. Indeed, one can easily verify that the generative procedure depicted in Figure 3 is not restricted to time-frequency domain, since at no point does the procedure rely on particularities of time and frequency. In general, the constants β and γ in equation 3.7 are still defined by equations 3.8 and 3.9, respectively. The constant $\tilde{\lambda}$ in equation 3.9 is in general defined as $\tilde{\lambda} = \lambda \, \mathrm{vol}(S)$, where S is the space in which the point processes are defined and $\mathrm{vol}(S)$ is the volume of that space. The SES algorithm can straightforwardly be extended to more general models, along the lines of the extensions we considered in sections 6.1 and 6.2.

For example, in time and three-dimensional space, SES may be described by the following statistical model:

$$\begin{aligned} p(e, e',\, j,\, j', \theta) = \gamma\, \beta^{n^{\text{tot}}_{\text{non-co}}}\, & p(\delta_t)\, p(s_t)\, p(\delta_x)\, p(s_x)\, p(\delta_y)\, p(s_y)\, p(\delta_z)\, p(s_z) \\ & \cdot \prod_{k=1}^{n^{\text{tot}}_{\text{co}}} p_w(w_{j_k})\, p_w(w'_{j'_k})\, p_{\Delta t}(\Delta t_{j_k})\, p_{\Delta t}(\Delta t'_{j'_k})\, p_{\Delta x}(\Delta x_{j_k})\, p_{\Delta x}(\Delta x'_{j'_k}) \\ & \cdot p_{\Delta y}(\Delta y_{j_k})\, p_{\Delta y}(\Delta y'_{j'_k})\, p_{\Delta z}(\Delta z_{j_k})\, p_{\Delta z}(\Delta z'_{j'_k})\, \mathcal{N}\big(t'_{j'_k} - t_{j_k};\, \bar{\delta}_t, \bar{s}_t\big) \\ & \cdot \mathcal{N}\big(x'_{j'_k} - x_{j_k};\, \bar{\delta}_x, \bar{s}_x\big)\, \mathcal{N}\big(y'_{j'_k} - y_{j_k};\, \bar{\delta}_y, \bar{s}_y\big)\, \mathcal{N}\big(z'_{j'_k} - z_{j_k};\, \bar{\delta}_z, \bar{s}_z\big), \end{aligned} \tag{6.9}$$

where $\bar{\delta}_t = \delta_t\,(\Delta t_{j_k} + \Delta t'_{j'_k})$, $\bar{\delta}_x = \delta_x\,(\Delta x_{j_k} + \Delta x'_{j'_k})$, $\bar{\delta}_y = \delta_y\,(\Delta y_{j_k} + \Delta y'_{j'_k})$, $\bar{\delta}_z = \delta_z\,(\Delta z_{j_k} + \Delta z'_{j'_k})$; the parameters $\bar{s}_t$, $\bar{s}_x$, $\bar{s}_y$, and $\bar{s}_z$ are defined similarly; and the parameters β and γ are again given by equations 3.8 and 3.9, respectively. The parameter $\tilde{\lambda}$ in equation 3.9 is now defined as

$$\tilde{\lambda} = \lambda(t_{\max} - t_{\min})(x_{\max} - x_{\min})(y_{\max} - y_{\min})(z_{\max} - z_{\min}). \tag{6.10}$$

In model 6.9, the bumps have dispersion in time and space. In some applications, however, the bumps may have dispersion only in space, not in time. In that case, one would need to replace $\bar{\delta}_t$ and $\bar{s}_t$ by δ_t and s_t, respectively, and there would be no factors $p_{\Delta t}(\Delta t_{j_k})$ and $p_{\Delta t}(\Delta t'_{j'_k})$.

Note that an SES model for point processes in three-dimensional space may be directly obtained from model 6.9. One simply needs to remove the factors $p(\delta_t)$, $p(s_t)$, $p_{\Delta t}(\Delta t_{j_k})$, $p_{\Delta t}(\Delta t'_{j'_k})$, and $\mathcal{N}(t'_{j'_k} - t_{j_k};\, \bar{\delta}_t, \bar{s}_t)$.

The inference algorithm for model 6.9 can be readily obtained from the algorithm of Table 2. The parameter updates are very similar, and the same holds for the pairwise alignment procedure. The upward message $g_{\mathcal{N}}$ (step 1 in Table 2) contains a gaussian factor for the timing offsets and similar factors for the offsets in the three spatial dimensions (cf. equation 6.9).

Interestingly, one can easily combine the above extensions. For example, one may consider oblique bumps in time and three-dimensional space; that model may take changes in bump orientation, amplitude, and width into account.

6.3.2 Non-Euclidean Spaces. So far, we have considered gaussian perturbations or, equivalently, Euclidean distances. In some applications, however, the point processes may be defined on curved manifolds, and non-Euclidean distances are then more natural. For instance, the two point processes may be defined on a planar closed curve. We consider such as example in Dauwels et al. (in press), which concerns the synchrony of morphological and molecular events in cell migration. More specifically, those events are extracted from time-lapse fluorescence resonance energy transfer

(FRET) images of Rac1 activity. The protein Rac1 is well known to induce filamentous structures that enable cells to migrate. The morphological and molecular events take place along the cell boundary, and since we consider images, that boundary is a closed planar curve. We do not take the dispersion of the events into account, since it is not relevant for the application at hand. The morphological and molecular events are denoted by $e = ((t_1, u_1, w_1), \ldots, (t_n, u_n, w_n))$ and $e' = ((t'_1, u'_1, w'_1), \ldots, (t'_{n'}, u'_{n'}, w'_{n'}))$, respectively, where t_k and $t'_{k'}$, u_k and $u'_{k'}$, and w_k and $w'_{k'}$ denote the occurrence time, position along the boundary, and the amplitude, respectively, of the morphological and molecular events. The distance between morphological and molecular events is non-Euclidean. We adopt the following statistical model for morphological and molecular events (Dauwels et al., in press):

$$p(e, e', j, j', \theta) = \gamma\, \beta^{n^{\text{tot}}_{\text{non-co}}}\, p(\delta_t)\, p(s_t)\, p(\delta_u)\, p(s_u) \prod_{k=1}^{n^{\text{tot}}_{\text{co}}} p_w(w_{j_k})\, p_w(w'_{j'_k}) \cdot \mathcal{N}\big(g_t(t_{j_k}, t'_{j'_k}); \delta_t, s_t\big)\, \mathcal{N}\big(g_u(u_{j_k}, u'_{j'_k}); \delta_u, s_u\big), \quad (6.11)$$

where g_t and g_u are nonlinear functions that take the shape of the cell boundary into account. Due to those nonlinearities, the factors $\mathcal{N}(g_t(t_{j_k}, t'_{j'_k}); \delta_t, s_t)$ and $\mathcal{N}(g_u(u_{j_k}, u'_{j'_k}); \delta_u, s_u)$ are not gaussian distributions, and the distance between events is non-Euclidean. The parameters β and γ are again given by equations 3.8 and 3.9, respectively. The parameter $\tilde{\lambda}$ in equation 3.9 is now defined as

$$\tilde{\lambda} = \lambda(t_{\max} - t_{\min})L, \quad (6.12)$$

where L is the length of the cell boundary. Extending the algorithm of Table 2 to model 6.11 is straightforward. The parameter updates (step 2 in Table 2) are now given by:

$$\hat{\delta}_t^{(i+1)} = \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} g_t\big(\hat{t}_k^{(i+1)}, \hat{t}'^{(i+1)}_k\big) \quad (6.13)$$

$$\hat{s}_t^{(i+1)} = \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \Big(g_t\big(\hat{t}_k^{(i+1)}, \hat{t}'^{(i+1)}_k\big) - \hat{\delta}_t^{(i+1)}\Big)^2 \quad (6.14)$$

$$\hat{\delta}_u^{(i+1)} = \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} g_u\big(\hat{u}_k^{(i+1)}, \hat{u}'^{(i+1)}_k\big) \quad (6.15)$$

$$\hat{s}_u^{(i+1)} = \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \Big(g_u\big(\hat{u}_k^{(i+1)}, \hat{u}'^{(i+1)}_k\big) - \hat{\delta}_u^{(i+1)}\Big)^2. \quad (6.16)$$

The pairwise alignment procedure is almost identical (step 1 in Table 2). We again need only to modify the upward message $g_{\mathcal{N}}$:

$$g_{\mathcal{N}}(c_{kk'}; \hat{\theta}^{(i)}) = \left(\mathcal{N}\left(g_t(t_k, t'_{k'}); \hat{\delta}_t^{(i)}, \hat{s}_t^{(i)}\right) \mathcal{N}\left(g_u(u_k, u'_{k'}); \hat{\delta}_u^{(i)}, \hat{s}_u^{(i)}\right)\right)^{c_{kk'}}.$$

In the next sections, we use the basic model 3.7, since that suffices for our purposes.

7 Analysis of Surrogate Data

As in the one-dimensional case (see section 6 in the companion letter), we investigate the robustness and reliability of multidimensional SES by means of surrogate data. We randomly generated 1000 pairs of two-dimensional point processes (e, e') according to the symmetric procedure depicted in Figure 3.

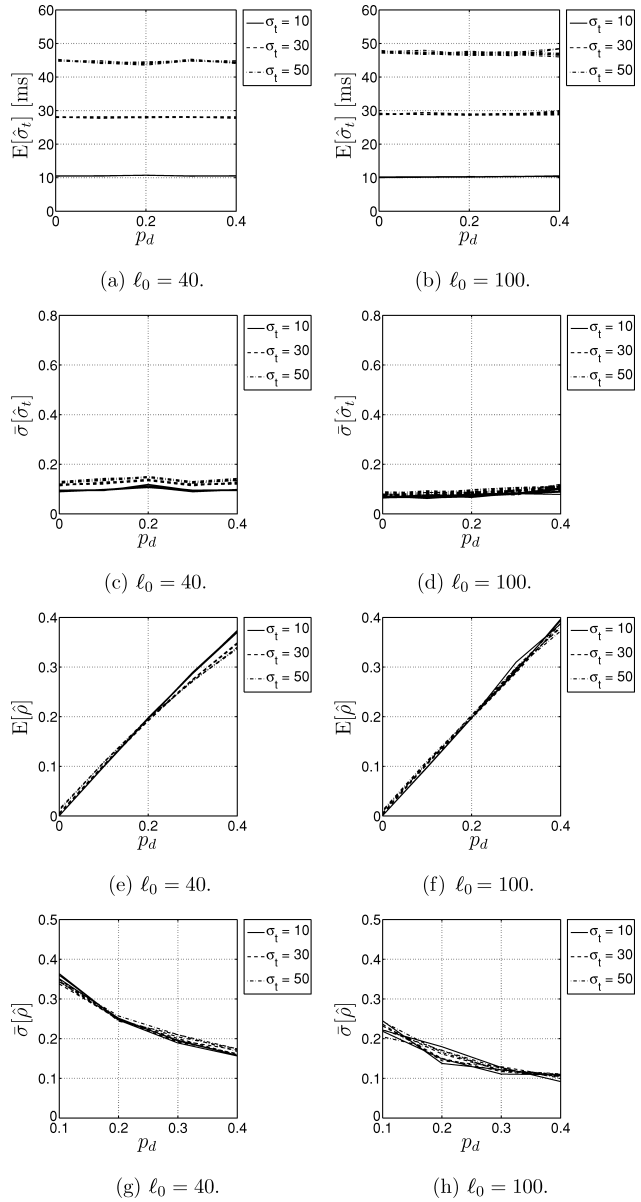
We considered several values of the parameters ℓ, p_d, δ_t, δ_f, s_t (σ_t), and s_f (σ_f). More specifically, the length ℓ was chosen as $\ell = \ell_0/(1 - p_d)$, where $\ell_0 \in \mathbb{N}_0$ is a constant. With this choice, the expected length of e and e' is ℓ_0, independent of p_d. We considered the values $\ell_0 = 40$ and 100, $p_d = 0$, 0.1, ..., 0.4, $\delta_t = 0$, 25, 50, $\sigma_t = 10$, 30, and 50, $\delta_f = 0$, 2.5, 5, $\sigma_f = 1$, 2.5, and 5, $t_{\min} = 0\,\mathrm{s}$, $f_{\min} = 0\,\mathrm{Hz}$, $t_{\max} = \ell_0 \cdot 100\,\mathrm{ms}$ and $f_{\max} = \ell_0 \cdot 1\,\mathrm{Hz}$. With this choice, the average event occurrence rate is about 10 Hz for all ℓ_0 and p_d. The width Δt_k and height Δf_k of all bumps is set equal to 0.5 ms and 0.5 Hz, respectively, so that $(\Delta t_k + \Delta t'_{k'}) = 1\,\mathrm{ms}$ and $(\Delta f_k + \Delta f'_{k'}) = 1\,\mathrm{Hz}$, for all k and k', and hence $\bar{\delta}_t = \delta_t$ ms, $\bar{\delta}_f = \delta_f$ Hz, $\bar{s}_t = s_t$ ms^2, and $\bar{s}_f = s_f$ Hz^2 (cf. equations 3.3–3.6, and Table 2).

We used the initial values $\hat{\delta}_t^{(0)} = 0$, 30, and 70, $\hat{\delta}_f^{(0)} = 0$, $\hat{s}_t^{(0)} = 30^2$, and $\hat{s}_f^{(0)} = 3^2$. The parameter β was identical for all parameter settings—$\beta = 0.005$. It was optimized to yield the best overall results. We used an uninformative prior for δ_t, δ_f, s_t, and s_f: $p(\delta_t) = p(\delta_f) = p(s_t) = p(s_f) = 1$.

In order to assess the SES measures $S = s_t, \rho$, we compute for each parameter setting the expectation $\mathrm{E}[S]$ and normalized standard deviation $\overline{\sigma}[S] = \sigma[S]/\mathrm{E}[S]$. Those statistics are computed by averaging over 1000 pairs of point processes (e,e'), randomly generated according to the symmetric procedure depicted in Figure 3.

The results are summarized in Figures 10 to 12. From those figures, we can make the following observations:

- The estimates of s_t and ρ are slightly biased, especially for small ℓ_0 (in particular, $\ell_0 = 40$), $s_t \geq 30^2$, and $p_d > 0.2$. More specifically, the expected value of those estimates is slightly smaller than the true value, which is due to the ambiguity inherent in event synchrony (cf. Figure 4). However, the bias is significantly smaller than in the one-dimensional case (cf. section 6 in the companion letter). The bias increases with s_f, which is in agreement with our expectations: the



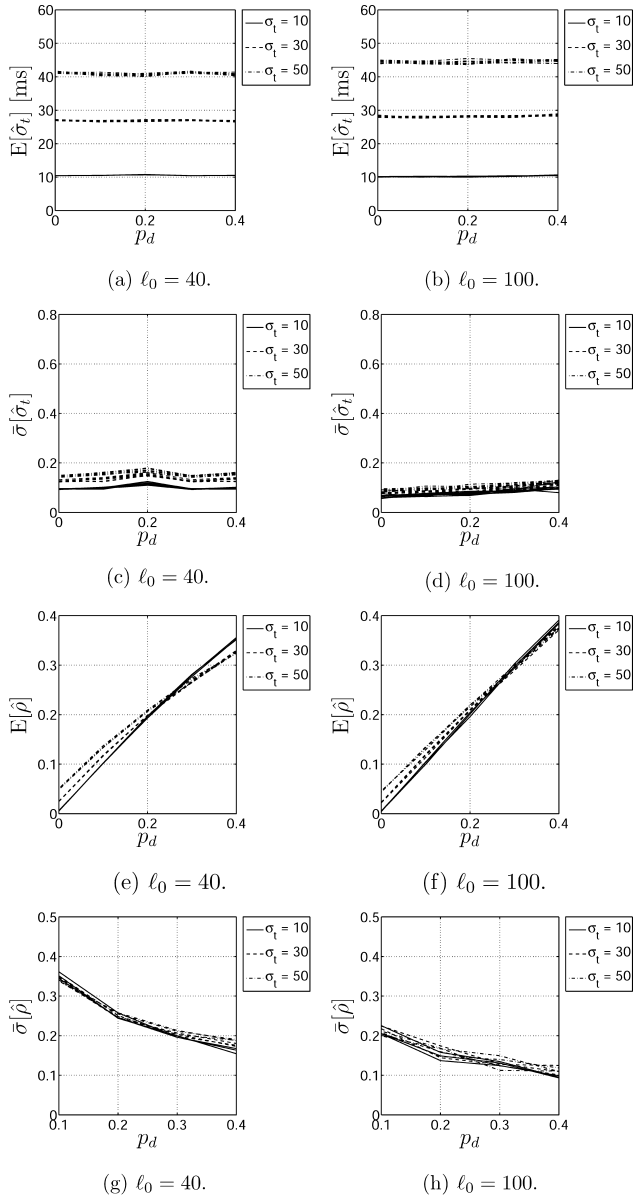
(a) $\ell_0 = 40$. (b) $\ell_0 = 100$.

(c) $\ell_0 = 40$. (d) $\ell_0 = 100$.

(e) $\ell_0 = 40$. (f) $\ell_0 = 100$.

(g) $\ell_0 = 40$. (h) $\ell_0 = 100$.

Figure 10: Results for surrogate data. The figure shows the expected value $\mathrm{E}[\hat{\sigma}_t]$ and $\mathrm{E}[\hat{\rho}]$ and the normalized standard deviation $\bar{\sigma}[\hat{\sigma}_t]$ and $\bar{\sigma}[\hat{\rho}]$ for the parameter settings $\ell_0 = 40$ and 100, $\delta_t = 0, 25, 50$, $\delta_f = 0, 2.5, 5$, $\sigma_t = 10, 30, 50$, $\sigma_f = 1$ and $p_d = 0, 0.1, \ldots, 0.4$. The curves for different δ_t and δ_f practically coincide.



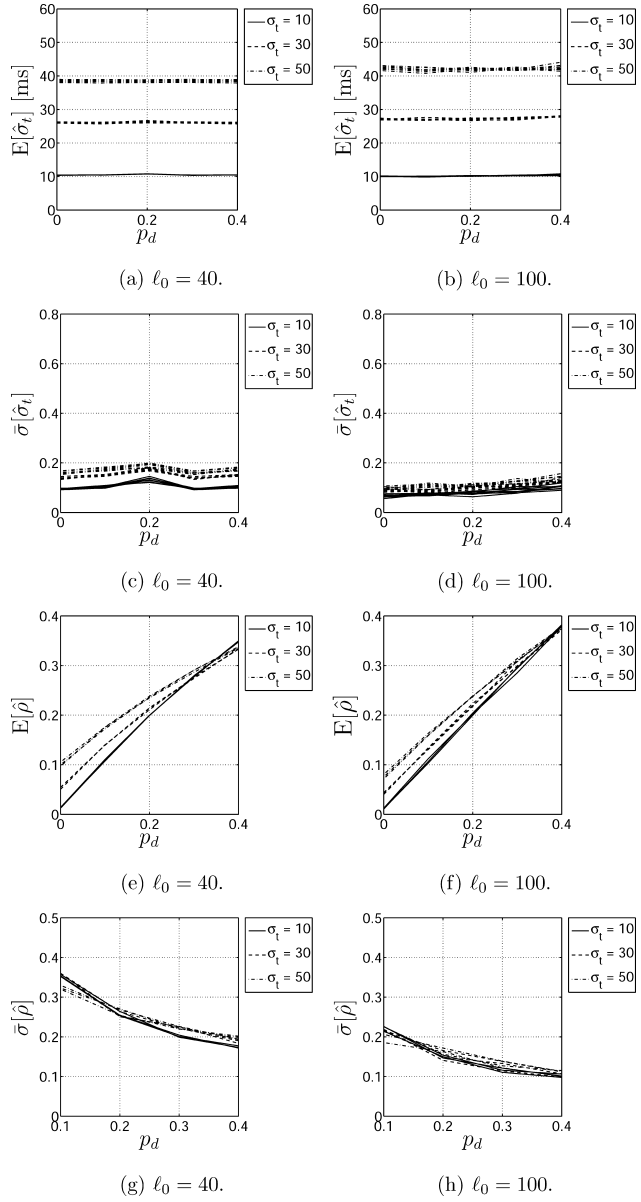
(a) $\ell_0 = 40$. (b) $\ell_0 = 100$.

(c) $\ell_0 = 40$. (d) $\ell_0 = 100$.

(e) $\ell_0 = 40$. (f) $\ell_0 = 100$.

(g) $\ell_0 = 40$. (h) $\ell_0 = 100$.

Figure 11: Results for surrogate data. The figure shows the expected value E[$\hat{\sigma}_t$] and E[$\hat{\rho}$] and the normalized standard deviation $\bar{\sigma}[\hat{\sigma}_t]$ and $\bar{\sigma}[\hat{\rho}]$ for the same parameter settings as in Figure 10, but now with $\sigma_f = 2.5$. Again, the curves for different δ_t and δ_f practically coincide.



(a) $\ell_0 = 40$. (b) $\ell_0 = 100$.

(c) $\ell_0 = 40$. (d) $\ell_0 = 100$.

(e) $\ell_0 = 40$. (f) $\ell_0 = 100$.

(g) $\ell_0 = 40$. (h) $\ell_0 = 100$.

Figure 12: Results for surrogate data. The figure shows the expected value $\mathrm{E}[\hat{\sigma}_t]$ and $\mathrm{E}[\hat{\rho}]$ and the normalized standard deviation $\bar{\sigma}[\hat{\sigma}_t]$ and $\bar{\sigma}[\hat{\rho}]$ for the same parameter settings as in Figure 10, but now with $\sigma_f = 5$. Again, the curves for different δ_t and δ_f practically coincide.

more frequency jitter there is, the more likely that some events are reversed in frequency, and hence are aligned incorrectly.

- As in the one-dimensional case, the estimates of δ_t are unbiased for all considered values of δ_t, δ_f, s_t, s_f, and p_d, as are the estimates of δ_f (not shown here).
- The estimates of s_t only weakly depend on p_d, and vice versa.
- The estimates of s_t and ρ do not depend on δ_t and δ_f. They are robust to lags δ_t and frequency offsets δ_f, since the latter can be estimated reliably.
- The normalized standard deviation of the estimates of δ_t, s_t, and ρ grows with s_t and p_d, but it remains below 30%. Those estimates are therefore reliable.
- The expected value of s_t and ρ weakly depends on the length ℓ_0: The estimates of s_t and ρ are less biased for larger ℓ_0. The normalized standard deviation of the SES parameters decreases as the length ℓ_0 increases, as expected.

In summary, by means of the SES inference method, one may reliably and robustly determine the timing dispersion s_t and event reliability ρ of pairs of multidimensional point processes. We wish to reiterate, however, that it slightly underestimates the timing dispersion and the number of event deletions due to the ambiguity inherent in event synchrony (cf. Figure 4). Moreover, as in the one-dimensional case, it is critical to choose an appropriate set of initial values $\hat{\delta}_t^{(0)}$, $\hat{\delta}_f^{(0)}$, $\hat{s}_t^{(0)}$, and $\hat{s}_f^{(0)}$.

8 Application: Diagnosis of Mild Cognitive Impairment from EEG

Several clinical studies have shown that the EEG of Alzheimer's disease (AD) patients is generally less coherent than of age-matched control subjects. This is also the case for patients suffering from mild cognitive impairment (see MCI; Jeong, 2004), for a review). In this section, we apply SES to detect subtle perturbations in EEG synchrony of MCI patients. First we describe the EEG data at hand (section 8.1); describe how we preprocess the EEG, extract bump models, and apply SES (section 8.2); and present our results (section 8.3).

8.1 EEG Data. The EEG data used here have been analyzed in previous studies concerning early diagnosis of AD (Chapman et al., 2007; Cichocki et al., 2005; Hogan, Swanwick, Kaiser, Rowan, & Lawlor, 2003; Musha et al., 2002; Vialatte et al., 2005).

Ag/AgCl electrodes (disks of diameter 8 mm) were placed on 21 sites according to the 10–20 international system, with the reference electrode on the right earlobe. EEG was recorded with Biotop 6R12 (NEC San-ei, Tokyo, Japan) using analog bandpass filtering in the frequency range 0.5 to 250 Hz at a sampling rate of 200 Hz. As in Chapman et al. (2007), Cichocki et al.

(2005), Hogan et al. (2003), Musha et al. (2002), and Vialatte et al. (2005), the signals were then digitally bandpass filtered between 4 Hz and 30 Hz using a third-order Butterworth filter.

The subjects were in two study groups. The first consisted of a group of 25 patients who had complained of memory problems. These subjects were then diagnosed as suffering from MCI and subsequently developed mild AD. The criteria for inclusion into the MCI group were a mini-mental-state exam (MMSE) score of 24 (max score, 30), though the average score in the MCI group was 26 (SD of 1.8). The other group was a control set consisting of 56 age-matched, healthy subjects who had no memory or other cognitive impairments. The average MMSE of this control group was 28.5 (SD of 1.6). The ages of the two groups were 71.9 ± 10.2 and 71.7 ± 8.3, respectively. Finally, it should be noted that the MMSE scores of the MCI subjects studied here are quite high compared to a number of other studies. For example, in Hogan et al. (2003) the inclusion criterion was MMSE $= 20$, with a mean value of 23.7, while in Chapman et al. (2007), the criterion was MMSE $= 22$ (the mean value was not provided). The disparity in cognitive ability between the MCI and control subjects was thus comparatively small, making the classification task relatively difficult.

All recording sessions were conducted with the subjects in an awake but resting state with eyes closed. The EEG technicians prevented the subjects from falling asleep (vigilance control). After recording, the EEG data were carefully inspected. Indeed, EEG recordings are prone to a variety of artifacts, for example, due to electronic smog, head movements, and muscular activity. The EEG data were investigated by an EEG expert, blinded from the results of this analysis. In particular, only subjects were retained in the analysis whose EEG recordings contained at least 20 s of artifact-free data. Based on this requirement, the number of subjects in the two groups described was reduced to 22 and 38, respectively. From each subject, one EEG segment of 20 s was analyzed (for each of the 21 channels).

8.2 Methods. We successively apply the following transformations to the EEG signals:

1. Wavelet transform
2. Normalization of the wavelet coefficients
3. Bump modeling of the normalized wavelet representation
4. Aggregation of the resulting bump models in several regions.

Eventually, we compute the SES parameters for each pair of aggregated bump models. In the following, we detail each of those five operations.

8.2.1 Wavelet Transform. In order to extract the oscillatory patterns in the EEG, we apply a wavelet transform. More specifically, we use the complex Morlet wavelets (Goupillaud, Grossman, & Morlet, 1984; Delprat et al.,

1992):

$$\psi(t) = A \exp\left(-t^2/2\sigma_0^2\right) \exp(2i\pi f_0 t), \tag{8.1}$$

where t is time, f_0 is frequency, σ_0 is a (positive) real parameter, and A is a (positive) normalization factor. The Morlet wavelet, equation 8.1, has proven to be well suited for the time-frequency analysis of EEG (see Tallon-Baudry, Bertrand, Delpuech, & Pernier, 1996; Herrmann, Grigutsch, & Busch, 2005). The product $w_0 = 2\pi f_0 \cdot \sigma_0$ determines the number of periods in the wavelet ("wavenumber"). This number should be sufficiently large ($\geq$5); otherwise the wavelet $\psi(t)$ does not fulfill the admissibility condition:

$$\int \frac{|\psi(t)|^2}{t} dt < \infty, \tag{8.2}$$

and as a result, the temporal localization of the wavelet becomes unsatisfactory (Goupillaud et al., 1984; Delprat et al., 1992). We choose a wavenumber $w_0 = 7$, as in the earlier studies (Tallon-Baudry et al., 1996; Vialatte, Martin, et al., 2007). This choice yields good temporal resolution in the frequency range we consider in this study.

The wavelet transform $x(t, s)$ of an EEG signal $x(t)$ is obtained as

$$x(t, s) \triangleq \sum_{t'=1}^{K} x(t')\, \psi^*\left(\frac{t' - t}{s}\right), \tag{8.3}$$

where $\psi(t)$ is the Morlet "mother" wavelet (see equation 8.1), s is a scaling factor, and $K = f_s T$, with f_s the sampling frequency and T the length of the signal. For the EEG data at hand, we have $T = 20s$ and $f_s = 200\,\text{Hz}$, and hence $K = 4000$. The scaled and shifted "daughter" wavelet in equation 8.3 has center frequency $f \triangleq f_0/s$. In the following, we use the notation $x(t, f)$ instead of $x(t, s)$.

Next we compute the squared magnitude $s(t, f)$ of the coefficients $x(t, f)$:

$$s(t, f) \triangleq |x(t, f)|^2. \tag{8.4}$$

Intuitively the time-frequency coefficients $s(t, f)$ represent the energy of oscillatory components with frequency f at time instances t. It is noteworthy that $s(t, f)$ contains no information about the phase of that component.

It is well known that EEG signals have a very nonflat spectrum with an overall 1/f shape, besides state-dependent peaks at specific frequencies. Therefore, the map $s(t, f)$ contains most energy at low frequencies f. If we directly apply bump modeling to the map $s(t, f)$, most bumps would be located in the low-frequency range; in other words, the high-frequency

range would be under-represented. Since relevant information might be contained at high frequency, we normalize the map $s(t, f)$ before extracting the bump models.

We point out that the time-frequency map $s(t, f)$ may be determined by alternative methods. For example, one may compute $s(t, f)$ by the multitaper method (Thomson, 1982) or by filter banks (Harris, 2004). We decided to use the Morlet wavelet transformation for two reasons:

- Morlet wavelets have the optimal joint time-frequency resolution, which is fundamentally limited by the uncertainty principle: the resolution in both time and frequency cannot be arbitrarily high simultaneously. It is well known that the Morlet wavelets achieve the uncertainty relation with equality (Goupillaud et al., 1984; Delprat et al., 1992; Mallat, 1999).
- EEG signals are typically highly nonstationary. The wavelet transform is ideally suited for nonstationary signals (Mallat, 1999), in contrast to approaches based on multitapers and filter banks.

However, it may also be meaningful to use a multitaper method or filter banks. For example, the multitaper method too is optimal in some sense: it minimizes out-of-band leakage, and all voxels of the time-frequency domain have the same size and shape. In addition, in the multitaper method, all estimates are independent due to orthogonality, a property not shared by wavelets (Mitra & Pesaran, 1999).

8.2.2 Normalization. The coefficients $s(t, f)$ are centered and normalized, resulting in the coefficients $\tilde{z}(t, f)$:

$$\tilde{z}(t, f) \stackrel{\triangle}{=} \frac{s(t, f) - m_s(f)}{\sigma_s(f)}, \tag{8.5}$$

where $m_s(f)$ is obtained by averaging $s(t, f)$ over the whole length of the EEG signal:

$$m_s(f) = \frac{1}{K} \sum_{t=1}^{K} s(t, f). \tag{8.6}$$

Likewise, $\sigma_s^2(f)$ is the variance of $s(t, f)$:

$$\sigma_s^2(f) = \frac{1}{K} \sum_{t=1}^{K} \big(s(t, f) - m_s(f)\big)^2. \tag{8.7}$$

In words, the coefficients $\tilde{z}(t, f)$ encode fluctuations from the baseline EEG power at time t and frequency f. The normalization 8.5 is known as z-score

(see, e.g., Buzsáki, 2006), and is commonly applied (Matthew & Cutmore, 2002; Martin, Gervais, Hugues, Messaoudi, & Ravel, 2004; Ohara, Crone, Weiss, and Lenz, 2004; Vialatte, Martin, et al., 2007; Chen et al., 2007). The coefficients $\tilde{z}(t, f)$ are positive when the activity at t and f is stronger than the baseline $m_s(f)$ and negative otherwise.

There are various approaches to apply bump modeling to the z-score $\tilde{z}(t, f)$. One may first set the negative coefficients to zero and next apply bump modeling. The bump models in that case represent peak activity. Alternatively, one may first set the positive coefficients equal to zero, reverse the sign of the negative coefficients, and then apply bump modeling. In that case, the bump models represent dips in the energy maps $s(t, f)$.

In the application of diagnosing AD (see section 8), we will follow yet another approach. In order to extract bump models, we wish to exploit as much information as possible from the $\tilde{z}$ maps. Therefore we set only a small fraction of the coefficients $\tilde{z}(t, f)$ equal to zero: the 1% smallest coefficients. This approach was also followed in Vialatte, Martin, et al. (2007), and is equivalent to the following transformation: we shift the coefficients 8.5 in the positive direction by adding a constant α, and the remaining negative coefficients are set to zero:

$$z(t, f) \triangleq \left\lceil \tilde{z}(t, f) + \alpha \right\rceil^{+} = \left\lceil \frac{s(t, f) - m_s(f)}{\sigma_s(f)} + \alpha \right\rceil^{+}, \tag{8.8}$$

where $\lceil x \rceil^{+} = x$ if $x \geq 0$ and $\lceil x \rceil^{+} = 0$ otherwise. The constant α is chosen such that only 1% of the coefficients remain negative after addition with α; this corresponds to $\alpha = 3.5$ in the application at hand. (In Vialatte, Martin, et al., 2007, it corresponds to $\alpha = 2$.) The top row of Figure 1 shows the normalized wavelet map z, equation 8.8, of two EEG signals.

8.2.3 Bump Modeling. Next, bump models are extracted from the coefficient maps z (see Figure 1 and Vialatte, Martin, et al., 2007). We approximate the map $z(t, f)$ as a sum $z_{\text{bump}}(t, f, \theta)$ of a "small" number of smooth basis functions or "bumps" (denoted by f_{bump}):

$$z(t, f) \approx z_{\text{bump}}(t, f, \theta) \triangleq \sum_{k=1}^{N_b} f_{\text{bump}}(t, f, \theta_k), \tag{8.9}$$

where θ_k are vectors of bump parameters and $\theta \triangleq (\theta_1, \theta_2, \ldots, \theta_{N_b})$. The sparse bump approximation $z_{\text{bump}}(t, f, \theta)$ represents regions in the time-frequency plane where the EEG contains more power than the baseline. In other words, it captures the most significant oscillatory activities in the EEG signal.

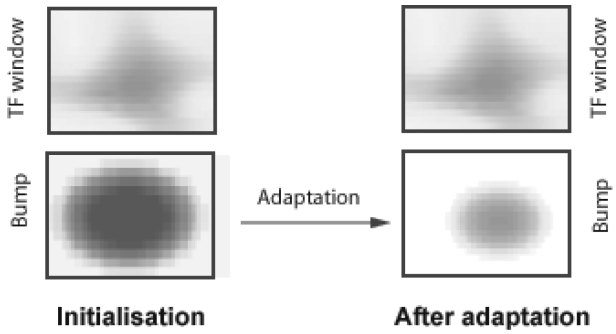


Figure 13: Learning the bump parameters by minimizing the quadratic cost function. A given patch of the time-frequency map (Top left and right); Initial bump (Bottom left); Bump obtained after adaptation (Bottom right).

We choose half-ellipsoid bumps since they are well suited for our purposes (Vialatte, 2005; Vialatte, Martin, et al., 2007) (see Figure 13). Since we wish to keep the number of bump parameters as low as possible, the principal axes of the half-ellipsoid bumps are restricted to be parallel to the time-frequency axes. As a result, each bump is described by five parameters: the coordinates of its center (i.e., time t_k and frequency f_k), its amplitude $w_k > 0$, and the extension Δt_k and Δf_k in time and frequency, respectively, in other words, $\theta_k = (t_k, f_k, w_k, \Delta t_k, \Delta f_k)$. More precisely, the ellipsoid bump function $f_{\text{bump}}(t, f, \theta_k)$ is defined as

$$f_{\text{bump}}(t, f, \theta_k) = \begin{cases} w_k \sqrt{1 - \kappa(t, f, \theta_k)} & \text{for } 0 \leq \kappa(t, f, \theta_k) \leq 1 \\ 0 & \text{for } \kappa(t, f, \theta_k) > 1, \end{cases} \tag{8.10}$$

where

$$\kappa(t, f, \theta_k) = \frac{(t-t_k)^2}{(\Delta t_k)^2} + \frac{(f-f_k)^2}{(\Delta f_k)^2}. \tag{8.11}$$

For the EEG data described in section 8.1, the number of bumps N_b (cf. equation 8.9) is typically between 50 and 100, and therefore, $z_{\text{bump}}(t, f, \theta)$ is fully specified by a few hundred parameters. On the other hand, the time-frequency map $z(t, f)$ consists of between 10^4 and 10^5 coefficients; the bump model $z_{\text{bump}}(t, f, \theta)$ is thus a sparse (but approximate) representation of $z(t, f)$.

The bump model $z_{\text{bump}}(t, f, \theta)$ is extracted from $z(t, f)$ by the following algorithm (Vialatte, 2005; Vialatte, Martin, et al., 2007):

1. Define appropriate boundaries for the map $z(t, f)$ in order to avoid finite-size effects.

2. Partition the map $z(t, f)$ into small zones. The size of these zones depends on the time-frequency ratio of the wavelets and is optimized to model oscillatory activities lasting four to five oscillation periods. Larger oscillatory patterns are modeled by multiple bumps.
3. Find the zone $\mathcal{Z}$ that contains the most energy.
4. Adapt a bump to that zone. The bump parameters are determined by minimizing the quadratic cost function (see Figure 13):

$$\mathcal{E}(\theta_k) \stackrel{\triangle}{=} \sum_{t,f\in\mathcal{Z}} \big(z(t, f) - f_{\text{bump}}(t, f, \theta_k)\big)^2. \tag{8.12}$$

   Withdraw the bump from the original map.
5. The fraction of total intensity contained in that bump is computed:

$$F = \frac{\sum_{t,f\in\mathcal{Z}} f_{\text{bump}}(t,f,\theta_k)}{\sum_{t,f\in\mathcal{Z}} z(t,f)}. \tag{8.13}$$

   If $F < G$ for three consecutive bumps (and hence those bumps contain only a small fraction of the energy of map $z(t, f)$), stop modeling and proceed to step 6; otherwise iterate step 3.
6. After all signals have been modeled, define a threshold $T \geq G$, and remove the bumps for which $F < T$. This allows us to trade off the information loss and modeling of background noise. When too few bumps are generated, information about the oscillatory activity of the brain is lost. But if too many bumps are generated, the bump model also contains low-amplitude oscillatory components. Since the measurement process typically introduces a substantial amount of noise, it is likely that the low-amplitude oscillatory components do not stem from organized brain oscillations but are instead due to measurement noise. By adjusting the threshold T, we try to find an appropriate number of bumps.

In our application, we used a threshold $G = 0.05$. With this threshold, each bump model contains many bumps. Some of those bumps may actually model background noise. Therefore, we further pruned the bump models (cf. step 6). We tested various values of the threshold $T \in [0.2, 0.25]$; as we will show, the results depend on the specific choice of T. The optimal separation between MCI and age-matched control subjects is obtained for $T = 0.22$; the separation gradually diminishes for increasing and decreasing values of T.

We refer to Vialatte (2005) and Vialatte, Martin, et al. (2007) for more information on bump modeling. In particular, we used the same choice of boundaries (step 1) and partitions (step 2) as in those references. Matlab Code of the bump extraction is available online at http://www.bsp.brain.riken.jp/%7Efvialatte/bumptoolbox/toolbox_home.html.

Eventually, we obtain 21 bump models: one per EEG channel. In the following, we describe how those models are further processed.

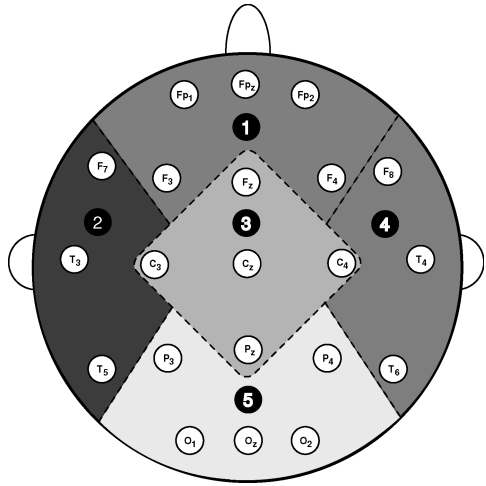


Figure 14: The 21 electrodes used for EEG recording, distributed according to the 10–20 international placement system (Nunez & Srinivasan, 2006). The clustering into $N_R = 5$ zones is indicated by the shading and dashed lines (1 = frontal, 2 = left temporal, 3 = central, 4 = right temporal, and 5 = occipital).

8.2.4 Aggregation. As a next step, we group the 21 electrodes into a small number N_R of regions, as illustrated in Figure 14 for $N_R = 5$. We report results for N_R = 3, 5, and 7. From the 21 bump models obtained by sparsification (cf. section 8.2.3), we extract a single bump model for each of the zones by means of the aggregation algorithm described in Vialatte, Martin, et al. (2007).

8.2.5 Stochastic Event Synchrony. Aggregation vastly reduces the computational complexity. Instead of computing the SES parameters among all possible pairs of 21 electrodes (210 in total), we compute those parameters for all pairs of regions— $N_R(N_R - 1)/2$ pairs in total. In addition, in order to obtain measures for average synchrony, we average the SES parameters over all region pairs, resulting in one set of average SES parameters per subject. It is noteworthy that in this setting, the SES parameters quantify large-scale synchrony, since each region spans several tens of millimeters. In the following, we consider only ρ and s_t, since those two parameters are the most relevant.

We choose the parameters of the SES algorithm as follows. Since we are dealing with spontaneous EEG, it is unlikely that the EEG signals from certain channels are delayed with regard to other channels; moreover, systematic frequency offsets are unrealistic. Therefore, we choose the initialization $\hat{\delta}_t^{(0)} = 0 = \hat{\delta}_f^{(0)}$. We used the parameter settings

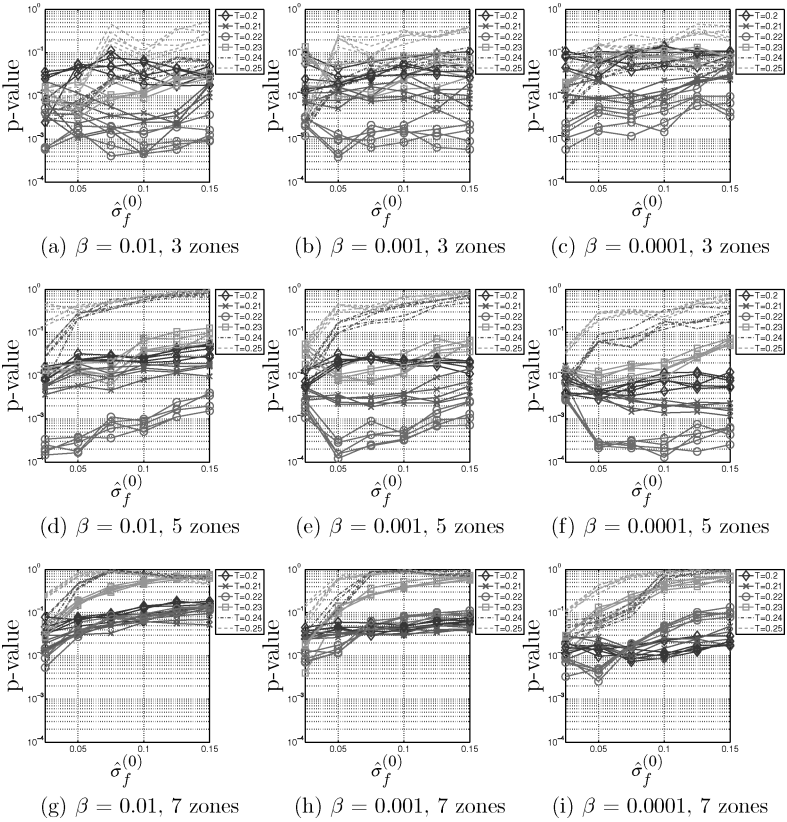
$\hat{s}_t^{(0)} = s_{0,t} = 0.15^2, 0.175^2, \ldots, 0.25^2$, and $\hat{s}_f^{(0)} = s_{0,f} = 0.025^2, 0.050^2, \ldots, 0.15^2$. We will show results for all those parameter values. The parameters ν_t and ν_f are set equal to 100, which corresponds to priors for s_t and s_f that have a sufficiently wide support (cf. Figure 6). We have observed that smaller values of ν_t and ν_f are not satisfactory (e.g., $\nu_t = 50 = \nu_f$): the prior takes nonnegligible values for large values of s_t and s_f, which leads to prohibitively large and unrealistic offsets in time and frequency. Larger values of ν_t and ν_f are not satisfactory either, since the priors for s_t and s_f then become too informative and would strongly bias the parameter estimates.

8.3 Results. The main results are summarized in Figures 15 and 16. They contain p-values obtained by the Mann-Whitney test for the parameters ρ and s_t, respectively. This test indicates whether the parameters take different values for the two subject populations. More precisely, low p-values indicate large difference in the medians of the two populations. The p-values are shown for $\hat{s}_t^{(0)} = s_{0,t} = 0.15^2, 0.175^2, \ldots, 0.25^2$, $\hat{s}_f^{(0)} = s_{0,f} = 0.025^2, 0.050^2, \ldots, 0.15^2$, $\beta = 0.01, 0.001, 0.0001$, $T = 0.2, 0.21, \ldots, 0.25$, and the number of zones $N_R = 3, 5$, and 7, with $\nu_t = 100 = \nu_f$.

The lowest p-values for ρ are obtained for $T = 0.22$ and $N_R = 5$ (see Figure 15). In particular, the smallest value is p $=1.2 \cdot 10^{-4}$, which occurs for $\beta = 0.001$, $\hat{s}_t^{(0)} = s_{0,t} = 0.225^2$, and $\hat{s}_f^{(0)} = s_{0,f} = 0.05^2$ (see Figure 19e).

It is interesting that the results depend on T (cf. section 8.2.3). That parameter allows us to balance the information loss and modeling of background noise: when too few bumps are generated, information about the oscillatory activity of the brain is lost. But if too many bumps are generated, the bump model also contains low-amplitude oscillatory components. The p-values are the lowest for $T = 0.22$ and become gradually larger as T decreases from $T = 0.22$ to 0.2 and as T increases from $T = 0.22$ to 0.25. One explanation could be that the number of bumps in each bump model is significantly smaller for MCI patients than in control subjects, with the maximum difference at $T = 0.22$. If the bump models of MCI patients contained fewer bumps, it would be intrinsically harder to align those models. However, as Figure 17 shows, this is not the case. On average, the bump models of MCI patients contain fewer bumps than the models of control subjects, but the difference is only weakly significant at best. Moreover, the largest difference does not consistently occur at $T = 0.22$. In other words, the difference in number of bumps between both subject populations cannot explain the dependency of the p-values on T.

This seems to suggest an alternative explanation: at $T = 0.22$, the optimal trade-off between information loss and modeling of background noise occurs. At lower values of T, the bump models contain more background noise—components that are unrelated to oscillatory events in the brain signals—and therefore the statistical differences between both populations



(a) $\beta = 0.01$, 3 zones (b) $\beta = 0.001$, 3 zones (c) $\beta = 0.0001$, 3 zones

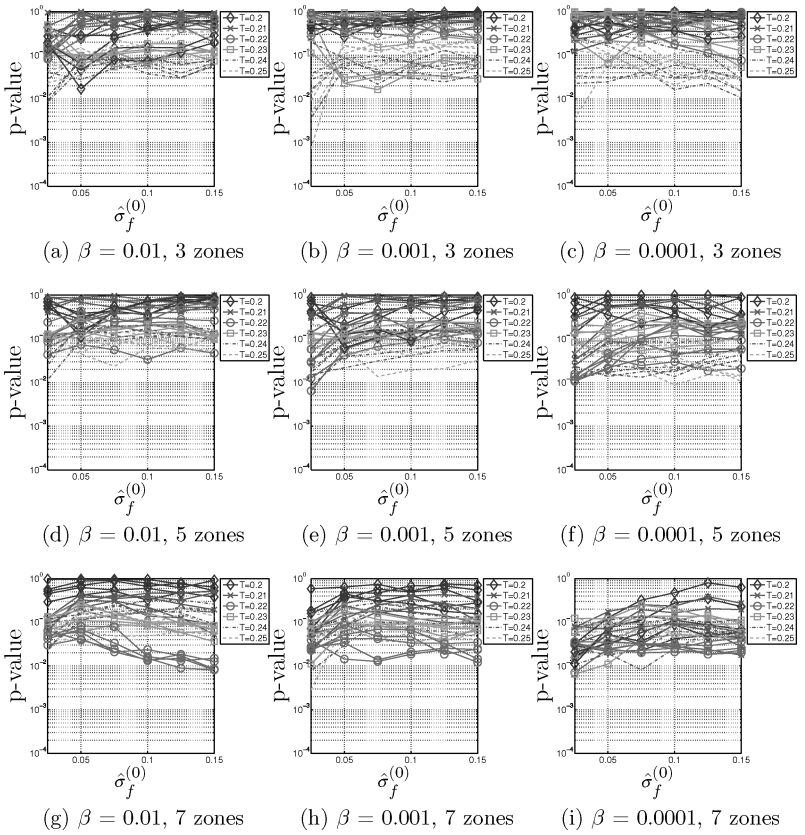
(d) $\beta = 0.01$, 5 zones (e) $\beta = 0.001$, 5 zones (f) $\beta = 0.0001$, 5 zones

(g) $\beta = 0.01$, 7 zones (h) $\beta = 0.001$, 7 zones (i) $\beta = 0.0001$, 7 zones

Figure 15: p-values obtained by the Mann-Whitney test for the parameter ρ for $(\hat{\sigma}_t^{(0)})^2 = \hat{s}_t^{(0)} = s_{0,t} = 0.15^2, 0.175^2, \ldots, 0.25^2$, $(\hat{\sigma}_f^{(0)})^2 = \hat{s}_f^{(0)} = s_{0,f} = 0.025^2, 0.050^2, \ldots, 0.15^2$, $\beta = 0.01$, 0.001, 0.0001, $T = 0.2$, 0.21, ..., 0.25 and the number of zones $N_R = 3$, 5, and 7, with $\nu_t = 100 = \nu_f$. The p-values seem to vary little with $s_t^{(0)}$, $s_f^{(0)}$, and β, but are more dependent on T and the number of zones. The lowest p-values are obtained for $T = 0.22$ and five zones; the corresponding statistical differences are highly significant.

decrease. At higher values of T, the models capture fewer oscillatory events in the brain signals, and therefore important information to distinguish both populations is discarded. The estimated parameters become less reliable.

From Figure 15, we can conclude that the statistical differences in ρ are highly significant, especially for $T = 0.22$ and $N_R = 5$. There is a significantly higher degree of noncorrelated activity in MCI patients, more specifically, a high number of noncoincident, nonsynchronous oscillatory events. Interestingly, we did not observe a significant effect on the timing jitter s_t of the coincident events (see Figure 16): very few p-values for s_t



(a) $\beta = 0.01$, 3 zones (b) $\beta = 0.001$, 3 zones (c) $\beta = 0.0001$, 3 zones

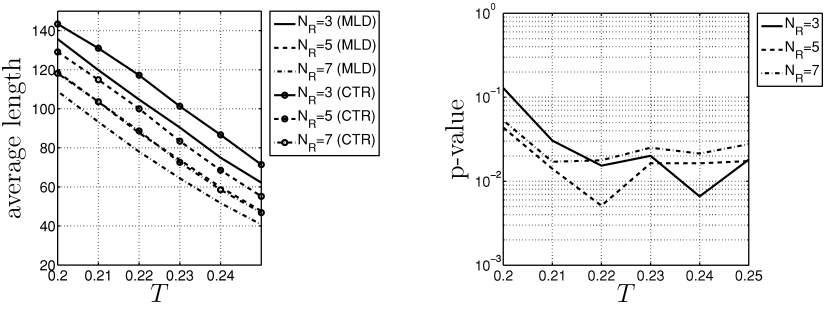
(d) $\beta = 0.01$, 5 zones (e) $\beta = 0.001$, 5 zones (f) $\beta = 0.0001$, 5 zones

(g) $\beta = 0.01$, 7 zones (h) $\beta = 0.001$, 7 zones (i) $\beta = 0.0001$, 7 zones

Figure 16: p-values obtained by the Mann-Whitney test for the parameter s_t for $(\hat{\sigma}_t^{(0)})^2 = \hat{s}_t^{(0)} = s_{0,t} = 0.15^2, 0.175^2, \ldots, 0.25^2$, $(\hat{\sigma}_f^{(0)})^2 = \hat{s}_f^{(0)} = s_{0,f} = 0.025^2, 0.050^2, \ldots, 0.15^2$, $\beta = 0.01$, 0.001, 0.0001, $T = 0.2$, 0.21, ..., 0.25, and the number of zones $N_R = 3$, 5, and 7, with $\nu_t = 100 = \nu_f$. Very few p-values are smaller than 0.01, which suggests there are no significant differences in s_t.

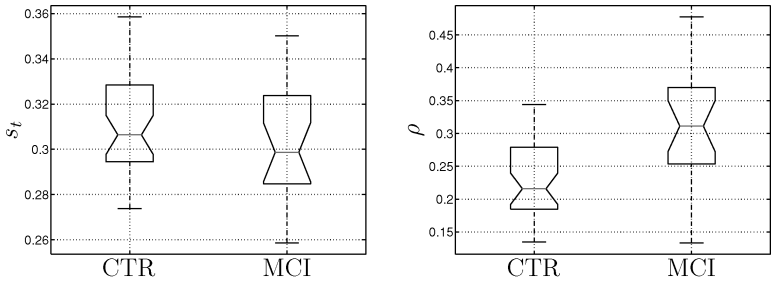
are smaller than 0.01, which suggests there are no significant differences in s_t. In other words, MCI seems to be associated with a significant increase of noncoincident background activity, while the coincident activity remains well synchronized. Figure 18 shows box plots for ρ and s_t, for the parameter setting that leads to the lowest p-values for ρ: $T = 0.22$, $N_R = 5$, $\beta = 0.001$, $\hat{s}_t^{(0)} = s_{0,t} = 0.225^2$, and $\hat{s}_f^{(0)} = s_{0,f} = 0.05^2$.

We now discuss how the p-values for ρ depend on $\hat{s}_t^{(0)} = s_{0,t}$ and $\hat{s}_f^{(0)} = s_{0,f}$. Figure 19 shows those p-values for $\hat{s}_t^{(0)} = s_{0,t} = 0.025^2, 0.050^2, \ldots, 0.25^2$, $\hat{s}_f^{(0)} = s_{0,f} = 0.025^2, 0.050^2, \ldots, 0.15^2$, $\beta = 0.01, 0.001, 0.0001$, with $T = 0.22$,



(a) Average number of bumps (b) p-values

Figure 17: Average number of bumps in each bump model for MCI and control subjects; average number (left) and p-values obtained by the Mann-Whitney test (right).



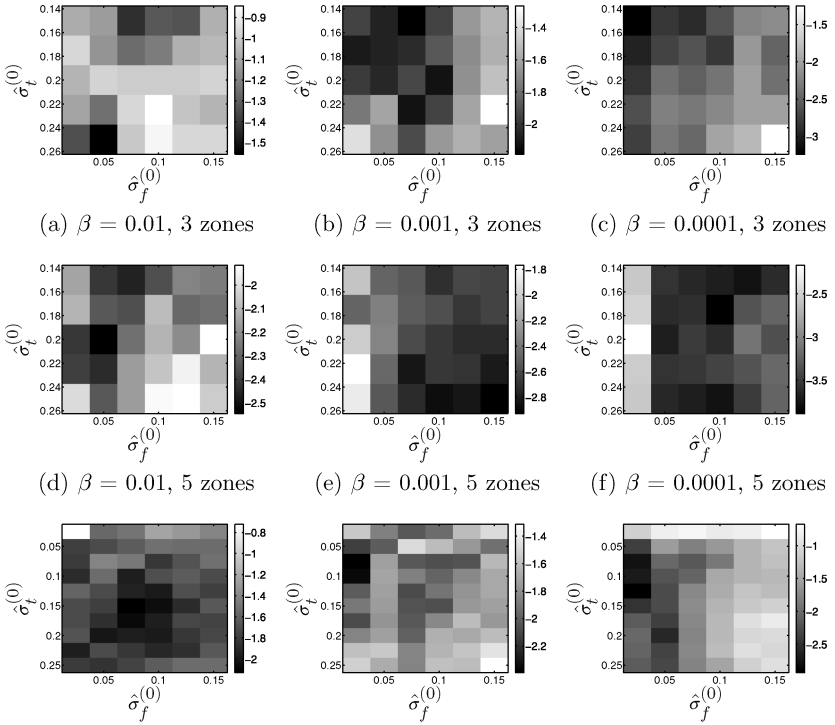
(a) s_t (p = 0.19) (b) ρ (p = 0.00012)

Figure 18: Box plots of s_t and ρ, for MCI and control subjects, with $T = 0.22$, $N_R = 5$, $\beta = 0.001$, $\hat{s}_t^{(0)} = s_{0,t} = 0.225^2$, and $\hat{s}_f^{(0)} = s_{0,f} = 0.05^2$. Interestingly, the parameter ρ leads to highly significant differences (p = 0.00012), in contrast to the parameter s_t (p = 0.19).

$N_R = 5$, and $\nu_t = 100 = \nu_f$. Note that, in order not to clutter the figures, we show results only for $\hat{s}_t^{(0)}$=$s_{0,t} = 0.15^2, 0.175^2, \ldots, 0.25^2$ in Figures 15 and 16; Figure 19 shows in addition results for $\hat{s}_t^{(0)} = s_{0,t} = 0.025^2, 0.050^2, \ldots, 0.125^2$.

When both $\hat{s}_t^{(0)} = s_{0,t}$ and $\hat{s}_f^{(0)} = s_{0,f}$ are smaller than or equal 0.75^2, the fraction of nonmatched events is usually about 70% to 80% (not shown here), and pairs of events that are close in time and frequency are not always matched. In other words, the obtained solutions are not satisfactory for those values of $\hat{s}_t^{(0)} = s_{0,t}$ and $\hat{s}_f^{(0)} = s_{0,f}$.

The smallest p-values occur typically for $\hat{s}_t^{(0)} > 0.15^2$ and $\hat{s}_f^{(0)} < 0.1^2$. This is in agreement with our expectations. As we argued in section 3, we expect bumps to appear at about the same frequency in both time-frequency maps,



(a) $\beta = 0.01$, 3 zones (b) $\beta = 0.001$, 3 zones (c) $\beta = 0.0001$, 3 zones

(d) $\beta = 0.01$, 5 zones (e) $\beta = 0.001$, 5 zones (f) $\beta = 0.0001$, 5 zones

(g) $\beta = 0.01$, 7 zones (h) $\beta = 0.001$, 7 zones (i) $\beta = 0.0001$, 7 zones

Figure 19: p-values (Mann-Whitney test) for the parameter ρ for $(\hat{\sigma}_t^{(0)})^2 = \hat{s}_t^{(0)} = s_{0,t} = 0.025^2, 0.050^2, \ldots, 0.25^2$, $(\hat{\sigma}_f^{(0)})^2 = \hat{s}_f^{(0)} = s_{0,f} = 0.025^2, 0.050^2, \ldots, 0.15^2$ $\beta = 0.01, 0.001, 0.0001$, with $T = 0.22$, $N_R = 3, 5, 7, \ldots$, and $\nu_t = 100 = \nu_f$.

since frequency shifts are hard to justify from a physiological perspective, whereas timing jitter arises quite naturally.

We verified that the SES measures ρ and s_t are not correlated with other synchrony measures, for example, Pearson correlation coefficient, magnitude and phase coherence, and phase synchrony (Pearson r, $p > 0.10$; see Dauwels, Vialatte, & Cichocki, 2008, for more details). In contrast to the classical measures, SES quantifies the synchrony of oscillatory events instead of more conventional amplitude or phase synchrony; therefore, it provides complementary information about EEG synchrony.

We applied a variety of classical synchrony measures to the same EEG data set (Dauwels et al., 2008). Most measures yield (weakly) significantly different values for the MCI and control subjects. Some differences are highly significant; the most significant results were obtained with the

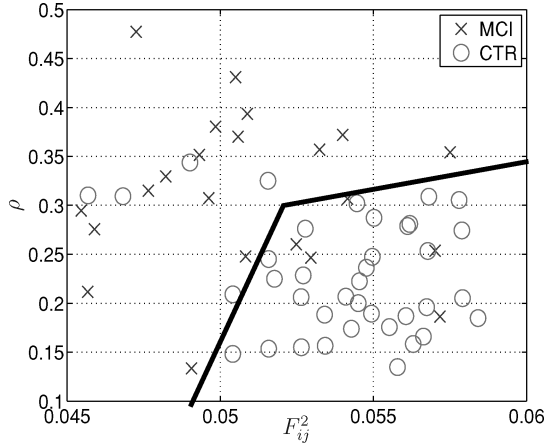
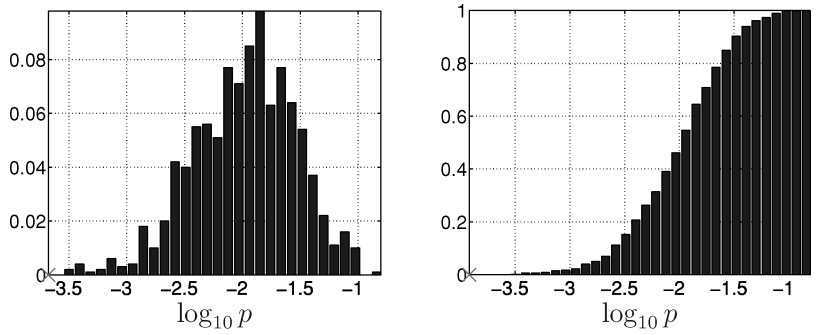


Figure 20: Combining ρ with ffDTF as features to distinguish MCI from age-matched control subjects. Note that ffDTF is a similarity measure, whereas ρ is a dissimilarity measure. The (ffDTF, ρ) pairs of the MCI and control subjects tend toward the left top corner and bottom right corner, respectively. The piecewise-linear decision boundary (solid) yields a classification rate of about 85%.

full-frequency direct transfer function (ffDTF), which is a Granger measure (Pereda et al., 2005), resulting in a p-value of about 10^{-3} (Mann-Whitney test). We combined ρ with ffDTF as features to distinguish MCI from control subjects (see Figure 20). We used the parameter setting of the SES algorithm that leads to the smallest p-value for ρ ($p = 1.2 \cdot 10^{-4}$); we verified that all parameter settings with $T = 0.22$ and $N_R = 5$ yield about the same classification results. About 85% of the subjects are correctly classified, which is a promising result. However, it is too weak to allow us to predict AD reliably. To this end, we would need to combine those two synchrony measures with complementary features, for example, derived from the slowing effect of MCI on EEG, or perhaps from different modalities such as PET, MRI, or biochemical indicators. We wish to point out, however, that in the data set at hand, patients did not carry out any specific task. In addition, we considered recordings of 20 s, which are rather short. It is plausible that the sensitivity of EEG synchrony could be further improved by increasing the length of the recordings and recording the EEG before, while, and after patients carry out specific tasks, such as, working memory tasks. As such, the classifier displayed in Figure 20 might be applied to screen a population for MCI, since it requires only an EEG recording system, a relatively simple and low-cost technology, available in most hospitals.

We tried to verify whether the small p-value of ρ is due to a decrease in coincident oscillatory events or whether it can be attributed to an effect not related to synchrony or perhaps to an artifact. To this end, we generated



(a) Histogram (b) Cumulative

Figure 21: Distribution of the p-value of parameter ρ for 1000 surrogate EEG data sets. The p-value of ρ for the actual EEG data set ($p = 0.00012$) is indicated by a cross. All surrogates yielded p-values larger than 0.00012.

and investigated surrogate data. From a given bump model, we obtain a surrogate bump model by shuffling the bumps over time. The center t_k of the bumps is chosen randomly; more precisely, it is drawn uniformly over the support of the bump model, and the other bump parameters are kept fixed. We created 1000 such bump models for each subject and obtained as a result 1000 surrogate EEG data sets. The distribution of the p-values of ρ for those 1000 surrogates is shown in Figure 21. The p-value of ρ for the actual EEG data set ($p = 0.00012$) is indicated by a cross. All the surrogates yielded p-values larger than 0.00012. We interpret this result as follows. If the p-values of the surrogate data were on average about 0.00012, we would be able to conclude that synchrony alone cannot explain the observed significant decrease in ρ. Since the p-values of the surrogates are on average much larger than 0.00012, it is less likely that other effects besides decrease of coincident neural activity result in the lower ρ in MCI patients.

We analyzed the convergence of the proposed inference algorithm (cf. Table 2). A histogram of the number of iterations (steps 1 and 2 in Table 2) required for convergence is shown in Figure 22, computed over all subjects, all pairs of regions, and all parameter settings. The algorithm converged after at most 23 iterations, and on average, after about 4 iterations. We allowed a maximum number of 50 iterations, and therefore, Figure 22 indicates that the algorithm always converged for the EEG data set at hand, as suggested by the theory of section 5.

Besides the algorithm of Table 2, we also implemented an algorithm in which the alignment 5.4 is carried out by a linear programming relaxation instead of the max-product algorithm. Since that algorithm is more complicated, we will not describe it here. We observed that both algorithms always converged to the same results. Moreover, since the max-product

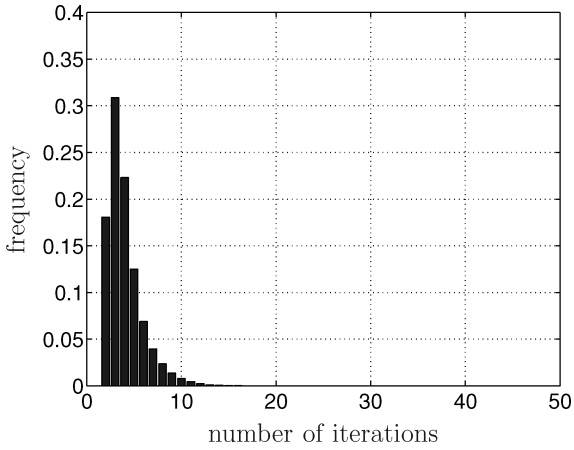


Figure 22: Histogram of the number of iterations (step 1 and 2 in Table 2) required for convergence, computed over all subjects and all pairs of regions. The algorithm converged after at most 23 iterations, and on average, after about 4 iterations. We allowed a maximum number of 50 iterations, and the histogram shows that the algorithm always converged for the EEG data set at hand.

algorithm always converged in our experiments, we can deduce that the optimal solution of the linear programming relaxation of equation 5.4 was every time unique (Bayati et al., 2005, 2008; Huang & Jebara, 2007; Sanghavi, 2007, 2008). Since it is well known that the linear programming relaxation is tight for bipartite max-weight matching (Gerards, 1995; Pulleyblank, 1995), we can conclude that in our experiments, both the max-product algorithm and linear programming relaxation of equation 5.4 resulted in the unique optimal alignment.

9 Conclusion

We have presented an alternative method to quantify the similarity of two time series, referred to as stochastic event synchrony (SES). The first step is to extract events from both time series, resulting in two point processes. The events in those point processes are then aligned. The better the alignment, the more similar the original time series are considered to be. We focused on multidimensional point processes in this letter.

Through the analysis of surrogate data, we verified that SES can also distinguish timing dispersion from event reliability in the multidimensional case. However, it typically underestimates the timing dispersion and overestimates event reliability; this is due to the ambiguous nature of the synchrony of point processes. The bias tends to be smaller for multidimensional point processes than for one-dimensional point processes.

Also in the multidimensional case, it is crucial to extract suitable events from the given time series. Only if those events are characteristic for the time series may SES yield meaningful results. As we have shown, for spontaneous EEG signals, it is natural to consider oscillatory events from the time-frequency representation; in particular, we considered bump models extracted from time-frequency maps of the EEG. However, depending on the nature of the EEG, there might be interesting alternatives, for example, based on matching pursuit or chirplets.

Since the proposed similarity measure does not take the entire time series into account but focuses exclusively on certain events, it provides complementary information about synchrony. Therefore, we believe that it may prove to be useful to blend our similarity measure with classical measures such as the Pearson correlation coefficient, Granger causality, or phase synchrony indices. We have shown that such a combined approach yields interesting results for the concrete application of diagnosing MCI from EEG: we computed ρ, the fraction of nonmatched oscillatory events, and full-frequency directed transfer function (ffDTF) from spontaneous EEG and used those two (dis)similarity measures as features to distinguish MCI from control subjects, resulting in a classification rate of about 85%. Moreover, we observed that there are significantly more nonmatched oscillatory events in the EEG of MCI subjects than in control subjects. The timing jitter s_t of the matched oscillatory events, however, is not different in the two subject groups. In future work, we will analyze additional data sets and incorporate other modalities such as fMRI and DTI into the analysis.

We wish to underline that the SES measures proposed in this letter are applicable only to pairs of signals. However, extensions to an arbitrary number of signals are feasible. Moreover, the SES parameters here are assumed to be constant; SES may be extended to time-varying parameters. Such extensions will be the subject of future reports.

Finally, we wish to outline another potential extension. In the generative process of SES, the events of the hidden point process are sampled independently and uniformly in the space at hand. However, in some applications, those events may naturally occur in clusters. More generally, the events may be statistically dependent. For example, it has been shown that specific frequency bands in EEG are sometimes coupled. Such couplings lead to correlations between the bumps in time-frequency domain. Our current analysis ignores such correlations. By taking those dependencies into account, we may be able to further improve our classification results and gain further insights about MCI and AD.

Appendix A: Factor Graphs

In this appendix, we provide some basic information on graphical models, in particular, factor graphs. We closely follow Loeliger (2004) and Loeliger et al. (2007). Graphical models are graphical representations of

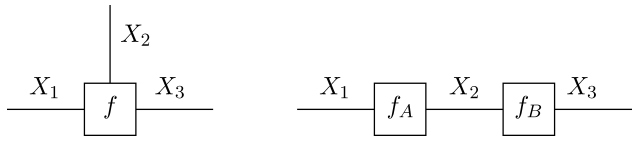


Figure 23: Factor graph of function without structure—$f(x_1, x_2, x_3)$ (left)—and a function with structure—$f(x_1, x_2, x_3) \triangleq f_A(x_1, x_2) f_B(x_2, x_3)$ (right).

multivariate functions. Examples are Markov random fields (or Markov networks), Bayesian networks (or belief networks), and factor graphs (Loeliger, 2004; Jordan, 1999; Loeliger et al., 2007). We use factor graphs in this letter—more specifically, Forney-style factor graphs or "normal" graphs, since they are more flexible than other types of graphical models. Moreover, the sum-product and max-product algorithm can be formulated most easily in the factor graph notation (see Loeliger, 2004, for a more detailed argumentation).

Graphical models (and factor graphs in particular) represent functions. Let us look at some examples.

Example 1: Factor Graph of a Function without Structure. The factor graph of the function $f(x_1, x_2, x_3)$ is shown in Figure 23 (left): edges represent variables, and nodes represent factors. An edge is connected to a node if and only if the corresponding variable is an argument of the corresponding function.

The concept of factor graphs becomes interesting as soon as the function has structure, that is, when it factors.

Example 2: Factor Graph of a Function with Structure. Let us assume that the function $f(x_1, x_2, x_3)$ of example 1 factors as $f(x_1, x_2, x_3) \triangleq f_A(x_1, x_2) f_B(x_2, x_3)$. The factor graph of Figure 23 (right) represents this factorization. We call f the global function and f_A and f_B local functions.

Example 3: The (global) function

$$f(x_1, x_2, x_3, x_4, x_5, x_6) \triangleq f_A(x_1, x_2) f_B(x_3, x_4) f_C(x_2, x_4, x_5) f_D(x_5, x_6) \tag{A.1}$$

is represented by the factor graph in Figure 24.

More formally, a Forney-style factor graph (FFG) is defined as follows:

- *Factor graph.* An FFG represents a function f and consists of nodes and edges. We assume that f can be written as a product of factors.

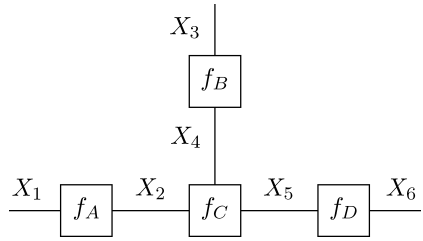


Figure 24: An example factor graph, representing the function A.1. Each node corresponds to a factor in that function— f_A, f_B, f_C, and f_D—each edge corresponds to a variable—X_1, X_2, ..., X_6.

- *Global functions.* The function f is called the global function.
- *Nodes/local functions.* There is a node for every factor, also called *local function*.
- *Edges/variables.* There is an edge or half-edge for every variable.
- *Connections.* An edge (or half-edge) representing some variable X is connected to a node representing some factor f if and only if f is a function of X.
- *Configuration.* A configuration is a particular assignment of values to all variables. We use capital letters for unknown variables and lowercase letters for particular values. This imitates the notation used in probability theory to denote chance or random variables and realizations thereof.
- *Configuration space.* The configuration space Ω is the set of all configurations: it is the domain of the global function f. One may regard the variables as functions of the configuration ω, just as we would with random or chance variables.
- *Valid configuration.* A configuration $\omega \in \Omega$ will be called valid if $f(\omega) \neq 0$.

Implicit in the previous definition is the assumption that no more than two edges are connected to one node. This restriction is easily circumvented by introducing variable replication nodes (also referred to as "equality constraint nodes"). An equality constraint node represents the factorization $\delta(x - x')\delta(x' - x'')$, and is depicted in Figure 25 (left). It enforces the equality of the variables X, X', and X''. The (single) equality constraint node generates two replicas of X: X' and X''. If more replicas are required, one can concatenate single nodes as shown in Figure 25 (middle); Combining those single nodes leads to a compound equality constraint node (see Figure 25 (right).

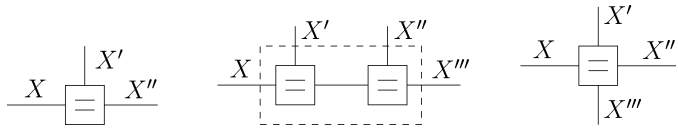


Figure 25: Equality constraint node used for variable replication. (Left) Single node. (Right) Compound node. (Middle) The compound node as concatenation of single nodes.

Appendix B: Summary Propagation Algorithm

This appendix aims at giving a brief review of the summary-propagation algorithm on a generic level. Also here we closely follow Loeliger (2004) and Loeliger et al. (2007). One of the most important operations that can be performed on factor graphs is marginalization: the computation of marginals of probability functions. Marginalization lies at the heart of many algorithms in signal processing, coding, and machine learning. As we will show, computing marginals amounts to passing messages ("summaries") along the edges in the factor graph of the system at hand. We describe this generic message-passing algorithm, called the sum(mary)-product algorithm (SPA).

B.1 Summary Propagation on Factor Trees.

Example 4: Marginalization of a Factored Function. Let us consider again the global function $f(x_1, x_2, x_3, x_4, x_5, x_6)$ of example 3. Suppose we are interested in the marginal function

$$f(x_5) \triangleq \sum_{x_1,x_2,x_3,x_4,x_6} f(x_1, x_2, x_3, x_4, x_5, x_6). \tag{B.1}$$

With the factorization (see equation A.1), we have:

$$\begin{aligned} f(x_5) &= \sum_{x_1,x_2,x_3,x_4,x_6} f_A(x_1, x_2) \cdot f_B(x_3, x_4) \cdot f_C(x_2, x_4, x_5) \cdot f_D(x_5, x_6) \\ &= \underbrace{\sum_{x_2,x_4} f_C(x_2, x_4, x_5) \underbrace{\left(\sum_{x_1} f_A(x_1, x_2) \right)}_{\mu_{f_A \to x_2}(x_2)} \cdot \underbrace{\left(\sum_{x_3} f_B(x_3, x_4) \right)}_{\mu_{f_B \to x_4}(x_4)}}_{\mu_{f_C \to x_5}(x_5)} \\ &\quad \cdot \underbrace{\left(\sum_{x_6} f_D(x_5, x_6) \right)}_{\mu_{f_D \to x_5}(x_5)}. \end{aligned} \tag{B.2}$$

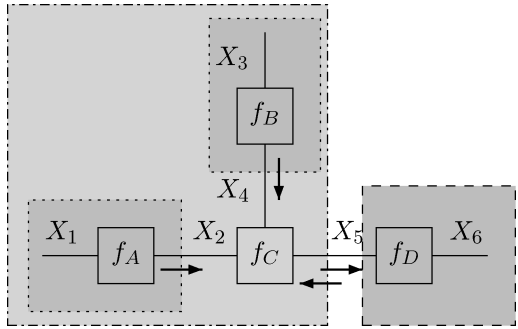


Figure 26: Summary propagation for computing $f(x_5)$. The arrows correspond to intermediate results, referred to as "messages" (see equation B.2).

The idea behind equation B.2 is to "push" the summations as much right as possible. For example, when summing with regard to X_6, we can push the summation sign to the right side of every factor except $f_D(x_5, x_6)$, since this factor depends on X_6. As a result, instead of carrying out a high-dimensional sum, it suffices to carry out simpler ones (one- and two-dimensional in our example). The intermediate terms $\mu_{f_j \to x_i}(x_i)$ are functions of X_i. The domain of such a function is the alphabet of X_i. Their meaning becomes obvious when looking at Figure 26.

The intermediate results can be interpreted as messages flowing along the edges of the graph. For example, the message $\mu_{f_A \to x_2}(x_2)$, which is the sum $\sum_{x_1} f_A(x_1, x_2)$, can be interpreted as a message leaving node f_A along edge X_2. If both $\mu_{f_A \to x_2}(x_2)$ and $\mu_{f_B \to x_4}(x_4)$ are available, the message $\mu_{f_C \to x_5}(x_5)$ can be computed as the output message of node f_C toward edge X_5. The final result of equation B.2 is

$$f(x_5) = \mu_{f_C \to x_5}(x_5) \cdot \mu_{f_D \to x_5}(x_5). \tag{B.3}$$

It is the product of the two messages along the same edge.

Each message can be regarded as a "summary" of what lies "behind" it, as illustrated by the boxes in Figure 26. Computing a message means "closing" a part of the graph ("box"). The details inside such a box are "summed out"; only a summary is propagated (hence the name *summary-propagation*). In the first step, the dark-shaded areas in Figure 26 are summarized (resulting in $\mu_{f_A \to x_2}(x_2)$ and $\mu_{f_D \to x_5}(x_5)$). Afterward, the lighter-shaded box is closed (amounting to $\mu_{f_C \to x_2}(x_2)$), until we arrive at equation B.3.

Half-edges (such as X_1) do not carry a message toward the connected node; alternatively, the edge may be thought of as carrying a message representing a neutral factor 1. With this in mind, we notice that every

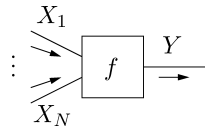


Figure 27: Message along a generic edge.

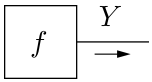


Figure 28: Message out of a leaf node.

message (i.e., every intermediate result) of equation B.2 is computed in the same way. Consider the generic node depicted in Figure 27 with messages arriving along its edges $X_1, \ldots, X_N$.

The message toward edge Y is computed by the following sum-product rule:

$$\mu_{f\to y}(y) \triangleq \sum_{x_1,\ldots,x_N} f(y, x_1, \ldots, x_N)\mu_{x_1\to f}(x_1)\cdots\mu_{x_N\to f}(x_N). \tag{B.4}$$

In words, the message out of a node f along the edge Y is the product of the function f and all messages toward f along all other edges, summarized over all variables except Y. This is the sum-product rule. In general, messages are computed out of any edge; there is no preferential direction. The message out of a leaf node f along edge Y is the function f itself, as illustrated in Figure 28.

Example 5: Maximization of a Factored Function. Let us consider again the global function $f(x_1, x_2, x_3, x_4, x_5, x_6)$ of example 4. Assume we are now interested in the function ("max-marginal")

$$f(x_5) \triangleq \max_{x_1,x_2,x_3,x_4,x_6} f(x_1, x_2, x_3, x_4, x_5, x_6). \tag{B.5}$$

With the factorization A.1, we have:

$$\begin{aligned} f(x_5) &= \max_{x_1,x_2,x_3,x_4,x_6} f_A(x_1,x_2)\cdot f_B(x_3,x_4)\cdot f_C(x_2,x_4,x_5)\cdot f_D(x_5,x_6) \\ &= \underbrace{\max_{x_2,x_4} f_C(x_2,x_4,x_5)\underbrace{\left(\max_{x_1} f_A(x_1,x_2)\right)}_{\mu_{f_A\to x_2}(x_2)}\cdot\underbrace{\left(\max_{x_3} f_B(x_3,x_4)\right)}_{\mu_{f_B\to x_4}(x_4)}}_{\mu_{f_C\to x_5}(x_5)} \\ &\quad \cdot\underbrace{\left(\max_{x_6} f_D(x_5,x_6)\right)}_{\mu_{f_D\to x_5}(x_5)}. \end{aligned} \tag{B.6}$$

It is noteworthy that every message of equation B.6 is computed according to the same rule. The max-product rule is as follows:

$$\mu_{f\to y}(y) \stackrel{\triangle}{=} \max_{x_1,\ldots,x_N} f(y,x_1,\ldots,x_N)\mu_{x_1\to f}(x_1)\cdots\mu_{x_N\to f}(x_N). \tag{B.7}$$

The sum-product and max-product rules can be considered as instances of the following single rule:

Summary-product rule: The message $\mu_{f\to y}(y)$ out of a factor node $f(y,\ldots)$ along the edge Y is the product of $f(y,\ldots)$ and all messages toward f along all edges except Y, summarized over all variables except Y.

The following example shows how several marginals can be obtained simultaneously in an efficient manner.

Example 6: Recycling Messages. Suppose we are also interested in the max-marginal function $f(x_2)$ of the global function $f(x_1,x_2,x_3,x_4,x_5,x_6)$ of example 4:

$$f(x_2) \stackrel{\triangle}{=} \max_{x_1,x_3,x_4,x_5,x_6} f(x_1,x_3,x_4,x_5,x_6).$$

This max-marginal can be computed by the max-product algorithm depicted in Figure 29. Note that we have already computed the messages $\mu_{f_A\to x_2}(x_2)$, $\mu_{f_B\to x_4}(x_4)$, and $\mu_{f_D\to x_5}(x_5)$ in equation B.6. They can be "reused" for computing $f(x_2)$. Eventually $f(x_2)$ is obtained as

$$f(x_2) = \mu_{f_A\to x_2}(x_2)\mu_{f_C\to x_2}(x_2). \tag{B.8}$$

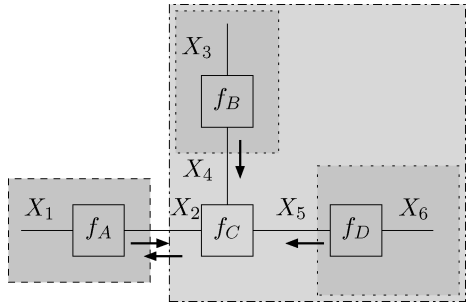


Figure 29: Summary propagation for computing $f(x_2)$. Note that we have already computed the messages $\mu_{f_A \to x_2}(x_2)$, $\mu_{f_B \to x_4}(x_4)$, and $\mu_{f_D \to x_5}(x_5)$ for computing $f(x_5)$ (see Figure 26). They can be "reused" for computing $f(x_2)$.

From example 6, we learn that the two messages associated with an edge are the same for the computation of each (max-)marginal. It is therefore sufficient to compute each message once. The (max-)marginal $f(y)$ of a certain variable Y is the product of the two messages on the corresponding edge, such as equations B.3 and B.8. In general, it is

$$f(y) = \mu_{f_A \to y}(y) \cdot \mu_{f_B \to y}(y), \tag{B.9}$$

where f_A and f_B are the two nodes attached to edge Y. For half-edges, the message coming from the open end carries a neutral factor 1. Therefore, the message from the node toward the edge is already the marginal of the corresponding variable.

In its general form, the summary-propagation algorithm (SPA) computes two messages on every edge. For factor graphs without loops (factor trees), the marginals can obtained in an optimal number of computations as follows.[1] One starts the message computation from the leaves and proceeds with nodes whose input messages become available. In this way, each message is computed exactly once, as illustrated in Figure 30. When the algorithm stops, exact marginals, such as equation B.9, are available for all variables simultaneously.

In summary:

- Marginals such as equation B.5 can be computed as the product of two messages as in equation B.9.
- Such messages are summaries of the subgraph behind them.
- All messages (except those out of terminal nodes) are computed from other messages according to the summary-product rule.

[1]The number of computations may be reduced by additional information about the structure of the local node functions. This is the case when the factor nodes themselves may be expressed by (nontrivial) factor trees.

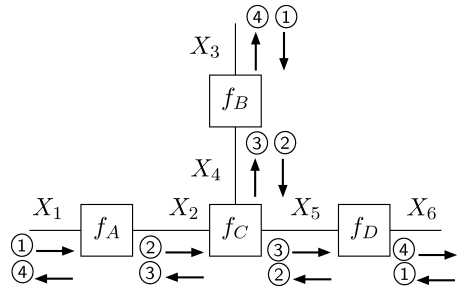


Figure 30: The SPA computes two messages along each edge. Those messages are required for calculating the marginal functions $f(x_1)$, $f(x_2)$, $f(x_3)$, $f(x_4)$, $f(x_5)$, and $f(x_6)$. The circled numbers indicate the order of the message computations.

If the summaries are computed by the sum-product rule, the above algorithm is referred to as sum-product algorithm or belief propagation. And if the summaries are computed according to the max-product rule, it is known as the max-product algorithm.

If rule B.4 or B.7 is applied, the values of the messages often quickly tend to zero, and the algorithm becomes instable. Therefore, it is advisable to scale the message. Instead of the message $\mu(.)$, a modified message $\tilde{\mu}(.) \triangleq \gamma\mu(.)$ is computed, where the scale factor γ may be chosen as one wishes. The final result, equation B.9, will then be known up to a scaling factor, which is often not a problem.

A message update schedule says when one has to calculate what message. For factor trees, there is an optimal message update schedule, as we explained previously; for cyclic factor graphs, this is not the case.

B.2 Summary Propagation on Cyclic Factor Graphs. The situation becomes quite different when the graph has cycles. In this case, the summary-propagation algorithm becomes iterative: a new output message at some node can influence the inputs of the same node through another path in the graph. The algorithm does not amount to the exact marginal functions. In fact, there is no guarantee that the algorithm converges. Astonishingly, applying the summary-product algorithm on cyclic graphs works excellently in the context of coding and signal processing, and machine learning. In many practical cases, the algorithm reaches a stable point, and the obtained marginal functions are satisfactory: decisions based on those marginals are often close enough to the "optimal" decisions.

Summary propagation on cyclic graphs consists of the following steps:

1. All edges are initialized with a neutral message—a factor $\mu(.) = 1$.
2. All messages are recursively updated according to some schedule. This schedule may vary from step to step.

3. After each step, the marginal functions are computed according to equation B.9.
4. One makes decisions based on the current marginal functions.
5. The algorithm is halted when the available time is over or when some stopping criterion is satisfied (e.g., when all messages varied less than some small ε over the last iterations).

Appendix C: Derivation of the SES Inference Algorithm

In this appendix, we derive the inference algorithm for multidimensional SES, summarized in Table 2.

The estimate $\hat{\theta}^{(i+1)}$, equation 5.5, is available in closed form; indeed, it is easily verified that the point estimates $\hat{\delta}_t^{(i+1)}$ and $\hat{\delta}_f^{(i+1)}$ are the (sample) mean of the timing and frequency offset, respectively, computed over all pairs of coincident events:

$$\hat{\delta}_t^{(i+1)} \triangleq \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \frac{\hat{t}_k'^{(i+1)} - \hat{t}_k^{(i+1)}}{(\Delta\hat{t}_k^{(i+1)} + \Delta\hat{t}_k'^{(i+1)})^2} \tag{C.1}$$

$$\hat{\delta}_f^{(i+1)} \triangleq \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \frac{\hat{f}_k'^{(i+1)} - \hat{f}_k^{(i+1)}}{(\Delta\hat{f}_k^{(i+1)} + \Delta\hat{f}_k'^{(i+1)})^2}, \tag{C.2}$$

where $n^{(i+1)}$ is the number of coincident bump pairs in alignment $\hat{c}^{(i+1)}$, and where we used the shorthand notation $\hat{t}_k^{(i+1)} = t_{\hat{j}_k^{(i+1)}}$, $\hat{f}_k^{(i+1)} = f_{\hat{j}_k^{(i+1)}}$, $\Delta\hat{t}_k^{(i+1)} = \Delta t_{\hat{j}_k^{(i+1)}}$, $\Delta\hat{f}_k^{(i+1)} = \Delta f_{\hat{j}_k^{(i+1)}}$, and likewise $\hat{t}'^{(i+1)}_k$, $\hat{f}'^{(i+1)}_k$, $\Delta\hat{t}'^{(i+1)}_k$, $\Delta\hat{f}'^{(i+1)}_k$.

The estimates $\hat{s}_t^{(i+1)}$ and $\hat{s}_f^{(i+1)}$ are obtained as

$$\hat{s}_t^{(i+1)} = \frac{\nu_t s_{0,t} + n^{(i+1)}\hat{s}_{t,\text{sample}}^{(i+1)}}{\nu_t + n^{(i+1)} + 2} \tag{C.3}$$

$$\hat{s}_f^{(i+1)} = \frac{\nu_f s_{0,f} + n^{(i+1)}\hat{s}_{f,\text{sample}}^{(i+1)}}{\nu_f + n^{(i+1)} + 2}, \tag{C.4}$$

where ν_t, ν_f, $s_{0,t}$, and $s_{0,f}$ are the parameters of the conjugate priors 3.16 and 3.17, and $s_{t,\text{sample}}$ and $s_{f,\text{sample}}$ are the (sample) variance of the timing and frequency offset, respectively, computed over all pairs of coincident events:

$$\hat{s}_{t,\text{sample}}^{(i+1)} \triangleq \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \frac{(\hat{t}_k'^{(i+1)} - \hat{t}_k^{(i+1)} - \hat{\delta}_t^{(i+1)})^2}{(\Delta\hat{t}_k^{(i+1)} + \Delta\hat{t}_k'^{(i+1)})^2} \tag{C.5}$$

$$\hat{s}_{f,\text{sample}}^{(i+1)} \triangleq \frac{1}{n^{(i+1)}} \sum_{k=1}^{n^{(i+1)}} \frac{(\hat{f}_k'^{(i+1)} - \hat{f}_k^{(i+1)} - \hat{\delta}_f^{(i+1)})^2}{(\Delta \hat{f}_k^{(i+1)} + \Delta \hat{f}_k'^{(i+1)})^2}. \tag{C.6}$$

Now we address the update, equation 5.4, i.e., finding the optimal pairwise alignment c for given values $\hat{\theta}^{(i)}$ of the parameters θ. In the following, we will show that it is equivalent to a standard problem in combinatorial optimization, that is, max-weight bipartite matching see, e.g., (Gerards, 1995; Pulleyblank, 1995; Bayati et al., 2005, 2008; Huang & Jebara, 2007; Sanghavi, 2007, 2008). First, let us point out that in equation 5.2, there is a factor β for every noncoincident bump; the total number of factors β is hence $n_{\text{non-co}} = n + n' - 2n_{\text{co}}$, where n_{co} is the number of coincident bump pairs. For each pair of coincident bumps, there is a factor $\mathcal{N}(\cdot; \bar{\delta}_t, \bar{s}_t)\mathcal{N}(\cdot; \bar{\delta}_f, \bar{s}_f)$. In total there are n_{co} such factors. Therefore, we can rewrite equation 5.2 as

$$p(e, e', c, \theta) \propto \prod_{k=1}^{n} \prod_{k'=1}^{n'} \left(\mathcal{N}(t_{k'}' - t_k; \bar{\delta}_t, \bar{s}_t)\, \mathcal{N}(f_{k'}' - f_k; \bar{\delta}_f, \bar{s}_f) \beta^{-2} \right)^{c_{kk'}} \cdot I(c) p(\delta_t) p(s_t) p(\delta_f) p(s_f), \tag{C.7}$$

where we omitted the factor $\beta^{n+n'}$ since it is an irrelevant constant, and

$$I(c) = \prod_{k=1}^{n} \left(\delta\left[\sum_{k'=1}^{n'} c_{kk'} \right] + \delta\left[\sum_{k'=1}^{n'} c_{kk'} - 1 \right] \right) \cdot \prod_{k'=1}^{n'} \left(\delta[\sum_{k=1}^{n} c_{kk'} \right] + \delta\left[\sum_{k=1}^{n} c_{kk'} - 1 \right] \right). \tag{C.8}$$

The factor $I(c)$ encodes the constraints 3.13. The maximization 5.4 is equivalent to

$$\hat{c}^{(i+1)} = \underset{c}{\text{argmax}} \log p(e, e', c, \hat{\theta}^{(i)}). \tag{C.9}$$

Using equation C.7, we can rewrite equation C.9 as

$$\hat{c}^{(i+1)} = \underset{c}{\text{argmax}} \sum_{kk'} w_{kk'} c_{kk'} + \log I(c) + \zeta, \tag{C.10}$$

where ζ is an irrelevant constant and

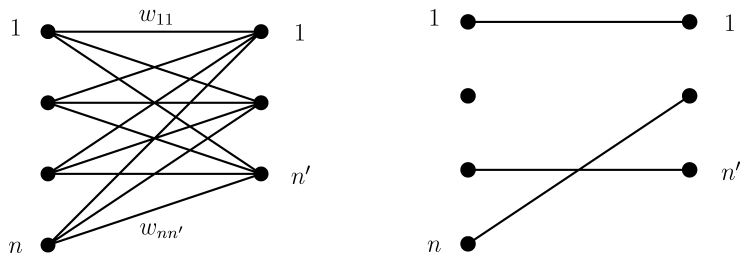
$$w_{kk'} = -\frac{\left(t_{k'}' - t_k - \hat{\delta}_t^{(i)}\right)^2}{2s_t(\Delta t_k + \Delta t_{k'}')^2} - \frac{\left(f_{k'}' - f_k - \hat{\delta}_f^{(i)}\right)^2}{2s_f(\Delta f_k + \Delta f_{k'}')^2} - 2\log\beta - 1/2\log 2\pi s_t(\Delta t_k + \Delta t_{k'}')^2 - 1/2\log 2\pi s_f(\Delta f_k + \Delta f_{k'}')^2, \tag{C.11}$$

where the weights $w_{kk'}$ can be positive or negative. If weight $w_{kk'}$ is negative, then $c_{kk'} = 0$. Indeed, setting $c_{kk'}$ equal to one would decrease $\log p(e, e', c, \hat{\theta}^{(i)})$. Bump pairs $(e_k, e'_{k'})$ with large weights $w_{kk'}$ are likely to be coincident. The closer the bumps $(e_k, e'_{k'})$ on the time-frequency plane, the larger their weight $w_{kk'}$. From the definition of β, equation 3.8, we can also see that the weights increase as the prior for a deletion p_d decreases. Indeed, the fewer deletions, the more likely that a bump e_k is coincident with a bump e'_k. In addition, the weights $w_{kk'}$ grow as the concentration λ of bumps on the time-frequency plane decreases. Indeed, if there are few bumps in each model (per square unit) and a bump e_k of e happens to be close to a bump $e'_{k'}$ of e', they are probably a coincident bump pair, since most likely, there are only a few other bumps in e' that are close to e_k.

One can naturally associate a bipartite graph with the optimization problem, equation C.10. The latter is a graph whose nodes can be divided into two disjoint sets $\mathcal{V}_1$ and $\mathcal{V}_2$ such that every edge connects a node in $\mathcal{V}_1$ and one in $\mathcal{V}_2$, that is, there is no edge between two vertices in the same set. As a first step, one associates a node to each bump in e, resulting in the set of nodes $\mathcal{V}_1$, and one associates a node to each bump in e', resulting in the set of nodes $\mathcal{V}_2$. Next, one draws edges between each node of $\mathcal{V}_1$ and $\mathcal{V}_2$, resulting in the bipartite graph depicted in Figure 31a. At last, one assigns a weight to every edge. More precisely, the edge between node k of $\mathcal{V}_1$ and node k' of $\mathcal{V}_2$ has weight $w_{kk'}$. Let us now look back at problem C.10: one maximizes a sum of weights $w_{kk'}$ subject to the constraints 3.3. This problem is equivalent to finding the heaviest disjoint set of edges in the bipartite graph of Figure 31a. This set of edges does not need to be connected to every node; some nodes may not be matched. For example, in Figure 31b, the second node of $\mathcal{V}_1$ is not matched. The latter problem is known as imperfect max-weight bipartite matching and can be solved in at least three different ways:

- By the Edmonds-Karp (Edmonds & Karp, 1972) or auction algorithm (Bertsekas & Tsitsiklis, 1989)
- By using the tight LP relaxation to the integer programming formulation of bipartite max-weight matching (Gerards, 1995; Pulleyblank, 1995)
- By applying the max-product algorithm (Bayati et al., 2005, 2008; Huang & Jebara, 2007; Sanghavi, 2007, 2008).

The Edmonds-Karp (Edmonds & Karp, 1972) and auction algorithm (Bertsekas & Tsitsiklis, 1989) both result in the optimum solution of equation C.10. The same holds for the linear programming relaxation approach and the max-product algorithm as long as the optimum solution is unique. If the latter is not unique, the linear programming relaxation method may result in noninteger solutions, and the max-product algorithm will not converge, as shown in (Sanghavi, 2007, 2008). Note that in many practical problems, the optimum matching C.10 is unique with probability one. This is in particular the case for bump models (see section 8.2). Since the



(a) Weighted bipartite graph. (b) Non-perfect matching.

Figure 31: Bipartite max-weight matching. The weighted bipartite graph (left) is obtained as follows. First, one associates a node to each bump in e, resulting in the set of nodes $\mathcal{V}_1$ (nodes at the left), and one associates a node to each bump in e', resulting in the set of nodes $\mathcal{V}_2$ (nodes at the right). Next, one draws edges between each node of $\mathcal{V}_1$ and $\mathcal{V}_2$ and associates a weight $w_{kk'}$ to each edge. Problem C.10 is equivalent to finding the heaviest disjoint set of edges in that weighted bipartite graph. Note that some nodes may not be connected to edges of that subset, that is, the matching may be nonperfect (right).

max-product algorithm is arguably the simplest algorithm in the list, we will describe only that algorithm.

Before we can apply the max-product algorithm on the graph of Figure 7 in order to find the optimal alignment, equation 5.4, we first need to slightly modify that graph. Indeed, the alignment c is computed for given $\theta = \hat{\theta}^{(i)}$, that is, one computes c conditioned on $\theta = \hat{\theta}^{(i)}$. Generally if one performs inference conditioned on a variable X, the edge(s) X need to be removed from the factor graph of the statistical model at hand. Therefore, for the purpose of computing equation 5.4, one needs to remove the θ edges (and the two bottom nodes in Figure 7), resulting in the factor graph depicted in Figure 8. It is noteworthy that the $\mathcal{N}$-nodes have become leaf nodes and that θ in $g_{\mathcal{N}}$, equation 4.1, is replaced by the estimate $\hat{\theta}^{(i)}$.

Before applying the max-product algorithm to Figure 8, we briefly describe it in general terms. The max-product algorithm is an optimization procedure that operates on a factor graph (or any other kind of graphical model; Jordan, 1999; Loeliger, 2004; Loeliger et al., 2007); local information (referred to as "messages") propagates along the edges in the graph and is computed at each node according to the generic max-product computation rule. After convergence or after a fixed number of iterations, one combines the messages in order to obtain decisions (Loeliger, 2004; Loeliger et al., 2007)). If the graph is cycle free, one obtains an optimal solution of the optimization problem; if the graph is cyclic, the max-product algorithm may not converge, and if it converges, the resulting decisions are not necessarily optimal (Loeliger, 2004; Loeliger et al., 2007). However, for certain problems

that involve cyclic graphs, it has been shown that the max-product algorithm is guaranteed to find the optimum solution. As we pointed out earlier, this is in particular the case for the max-weight matching problem (Bayati et al., 2005, 2008; Huang & Jebara, 2007; Sanghavi, 2007, 2008). We refer to appendix B for more information on the max-product algorithm.

The messages in the graph of Figure 8 are iteratively updated according to the max-product update rule, which is stated in row 1 of Table 3 for a generic node g. We now apply that generic rule to the nodes in Figure 8. Let us first consider the β- and $\mathcal{N}$-nodes, which are leaf nodes. The max-product message leaving a leaf node is nothing but the node function itself (see row 2 of Table 3). Therefore, the messages $\mu\downarrow(b_k)$ and $\mu\downarrow(b'_k)$, propagating downward along the edges B_k and B'_k, respectively, are given by

$$\mu\downarrow(b_k) = g_\beta(b_k) = \beta\delta[b_k - 1] + \delta[b_k] \tag{C.12}$$

$$\mu\downarrow(b'_k) = g_\beta(b'_k) = \beta\delta[b'_k - 1] + \delta[b'_k], \tag{C.13}$$

and similarly, the messages $\mu\uparrow(c_{kk'})$, propagating upward along the edges $C_{kk'}$:

$$\mu\uparrow(c_{kk'}) = g_{\mathcal{N}}(c_{kk'}; \hat{\theta}^{(i)}) \tag{C.14}$$

$$= \Big(\mathcal{N}\big(t'_{k'} - t_k; \bar{\delta}_t^{(i)}, \bar{s}_t^{(i)}\big)\, \mathcal{N}\big(f'_{k'} - f_k; \bar{\delta}_f^{(i)}, \bar{s}_f^{(i)}\big)\Big)^{c_{kk'}}, \tag{C.15}$$

where $\bar{\delta}_t^{(i)} = \hat{\delta}_t^{(i)}(\Delta t_k + \Delta t'_{k'})$, $\bar{\delta}_f^{(i)} = \hat{\delta}_f^{(i)}(\Delta f_k + \Delta f'_{k'})$, $\bar{s}_t^{(i)} = \hat{s}_t^{(i)}(\Delta t_k + \Delta t'_{k'})^2$, and $\bar{s}_f^{(i)} = \hat{s}_f^{(i)}(\Delta f_k + \Delta f'_{k'})^2$. Note that the messages $\mu\downarrow(b_k)$ and $\mu\downarrow(b'_k)$ never change; they do not need to be recomputed in the course of the max-product algorithm.

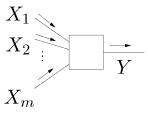
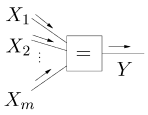
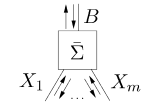
We now turn to the messages at the $\bar{\Sigma}$-nodes. Row 4 of Table 3 considers a generic $\bar{\Sigma}$-node. Its (generic) incident edges $X_1, \ldots, X_m$ are replaced by $C_{1k'}, \ldots, C_{nk'}$ or $C_{k1}, \ldots, C_{kn'}$ in Figure 8. For convenience, we now compute the messages in terms of $X_1, \ldots, X_m$; later we will formulate them in terms of $C_{kk'}$ (more specifically, in equations C.22, C.23, C.30, and C.32). The message $\mu\uparrow(b)$, propagating upward along the edge B, is computed as

$$\mu\uparrow(b) = \max_{x_1,\ldots,x_m} \delta\Big[b + \sum_{k=1}^{m} x_k - 1\Big]\, \mu\uparrow(x_1) \ldots \mu\uparrow(x_m) \tag{C.16}$$

$$= \mu\uparrow(x_1 = 0) \ldots \mu\uparrow(x_m = 0) \cdot \Big(\delta[b-1] + \delta[b] \max_k \mu\uparrow(x_k = 1)/\mu\uparrow(x_k = 0)\Big), \tag{C.17}$$

where $X_1, \ldots, X_m, B \in \{0, 1\}$ and $\mu\uparrow(x_k)$ are the messages propagating upward along the edges X_k. In row 4 of Table 3, we have written the message

Table 3: Max-Product Computation Rules for the Nodes in the Factor Graph of Figure 8.

1	 $g(x_1, x_2, \dots x_m, y)$	$\mu(y) \propto \max_{x_1,x_2,\dots,x_m} g(x_1, \dots, x_m, y)\mu(x_1) \dots \mu(x_m)$
2	 $g(y)$	$\mu(y) \propto g(y)$
3	 $\delta[x_1 - x_2] \dots \delta[x_m - y]$ $X_1, \dots, X_m, Y \in \{0, 1\}$	$\begin{pmatrix} \mu(y=0) \\ \mu(y=1) \end{pmatrix} \propto \begin{pmatrix} \mu(x_1=0)\mu(x_2=0) \dots \mu(x_m=0) \\ \mu(x_1=1)\mu(x_2=1) \dots \mu(x_m=1) \end{pmatrix}$
4	 $\delta\big[b + \sum_{k=1}^{m} x_k - 1\big]$ $B, X_1, \dots, X_m \in \{0, 1\}$	$\begin{pmatrix} \mu{\downarrow}(x_k=0) \\ \mu{\downarrow}(x_k=1) \end{pmatrix} \propto \begin{pmatrix} \max\left(\frac{\mu\downarrow(b=1)}{\mu\downarrow(b=0)}, \max_{\ell\neq k} \frac{\mu\uparrow(x_\ell=1)}{\mu\uparrow(x_\ell=0)}\right) \\ 1 \end{pmatrix}$ $\begin{pmatrix} \mu{\uparrow}(b=0) \\ \mu{\uparrow}(b=1) \end{pmatrix} \propto \begin{pmatrix} \max_k \frac{\mu\uparrow(x_k=1)}{\mu\uparrow(x_k=0)} \\ 1 \end{pmatrix}$

$\mu\uparrow(b)$ componentwise. The messages $\mu\downarrow(x_k)$, propagating downward along the edges X_k, are computed similarly:

$$\mu\downarrow(x_k) = \max_{b,x_1,\ldots,x_{k-1},x_{k+1}\ldots,x_m} \delta\left[b + \sum_{k=1}^{m} x_k - 1\right] \mu\downarrow(b)\mu\uparrow(x_1)\ldots\mu\uparrow(x_{k-1}) \cdot \mu\uparrow(x_{k+1})\ldots\mu\uparrow(x_m) \tag{C.18}$$

$$= \mu\uparrow(b=0)\mu\uparrow(x_1=0)\ldots\mu\uparrow(x_{k-1}=0)\mu\uparrow(x_{k+1}=0) \ldots\mu\uparrow(x_m=0)\cdot\Big[\delta[x_k-1]+\delta[x_k]\max\big(\mu\downarrow(b=1)/\mu\downarrow(b=0), \max_{\ell\neq k}\mu\uparrow(x_\ell=1)/\mu\uparrow(x_\ell=0)\big)\Big], \tag{C.19}$$

where $\mu\downarrow(b)$ is the message propagating downward along the edge B. The componentwise formulation of $\mu\downarrow(x_k)$ is also listed in row 4 of Table 3.

At last, we turn to the messages computed at the equality constraint nodes in Figure 8. The (generic) equality constraint node is considered in row 3 of Table 3. The message $\mu(y)$, leaving this node along the edge Y, is computed as follows:

$$\mu(y) = \max_{x_1,\ldots,x_m} \delta[x_1-x_2]\ldots\delta[x_{m-1}-x_m]\delta[x_m-y]\,\mu(x_1)\ldots\mu(x_m) \tag{C.20}$$

$$= \mu(x_1=y)\ldots\mu(x_m=y), \tag{C.21}$$

where $X_1,\ldots,X_m, Y \in \{0,1\}$. Since the equality constraint node is symmetric, the other messages leaving the equality constraint node (along the edges $X_1,\ldots,X_m$) are computed analogously.

We now use equations C.12 to C.21 to derive the update rules for the messages $\mu\uparrow(b_k)$, $\mu\uparrow(b'_k)$, $\mu\downarrow(c_{kk'})$, $\mu\uparrow'(c_{kk'})$, $\mu\downarrow'(c_{kk'})$, $\mu\uparrow''(c_{kk'})$, and $\mu\downarrow''(c_{kk'})$ in Figure 8:

- The messages $\mu\uparrow(b_k)$ and $\mu\uparrow(b'_k)$ propagate upward along the edges b_k and b'_k respectively, toward the β-nodes.
- The messages $\mu\downarrow(c_{kk'})$ propagate downward along the edges $C_{kk'}$, leaving the equality constraint nodes.
- The messages $\mu\uparrow'(c_{kk'})$ and $\mu\uparrow''(c_{kk'})$ propagate upward along the edges $C_{kk'}$, toward the $\bar{\Sigma}$-nodes connected to the edges B_k and $B'_{k'}$, respectively (see Figure 8, left-hand side).
- The messages $\mu\downarrow'(c_{kk'})$ and $\mu\downarrow''(c_{kk'})$ propagate downward along the edges $C_{kk'}$, leaving the $\bar{\Sigma}$-nodes connected to the edges B_k and $B'_{k'}$, respectively.

We start with the messages $\mu\uparrow(b_k)$:

$$\mu\uparrow(b_k) = \mu\uparrow'(c_{k1}=0)\ldots\mu\uparrow'(c_{kn'}=0) \cdot\Big(\delta[b_k-1]+\delta[b_k]\max_{k'}\mu\uparrow'(c_{kk'}=1)/\mu\uparrow'(c_{kk'}=0)\Big), \quad \text{(C.22)}$$

where we used equation C.17, and

$$\mu\uparrow(b'_{k'}) = \mu\uparrow''(c_{1k'}=0)\ldots\mu\uparrow''(c_{nk'}=0) \cdot\Big(\delta[b'_{k'}-1]+\delta[b'_{k'}]\max_{k}\mu\uparrow''(c_{kk'}=1)/\mu\uparrow''(c_{kk'}=0)\Big). \quad \text{(C.23)}$$

The messages $\mu\downarrow(c_{kk'})$ are derived as follows:

$$\mu\downarrow(c_{kk'}) = \mu\downarrow'(c_{kk'})\mu\downarrow''(c_{kk'}), \quad \text{(C.24)}$$

where we used equation C.21.

The messages $\mu\uparrow'(c_{kk'})$ are derived as follows:

$$\mu\uparrow'(c_{kk'}) \propto \mu\downarrow''(c_{kk'})\mu\uparrow(c_{kk'}) = \mu\downarrow''(c_{kk'})g_{\mathcal{N}}(c_{kk'};\hat{\theta}^{(i)}) \quad \text{(C.25)}$$

$$= \mu\downarrow''(c_{kk'})\Big(\mathcal{N}\big(t'_{k'}-t_k;\bar{\delta}_t^{(i)},\bar{s}_t^{(i)}\big)\,\mathcal{N}\big(f'_{k'}-f_k;\bar{\delta}_f^{(i)},\bar{s}_f^{(i)}\big)\Big)^{c_{kk'}}, \quad \text{(C.26)}$$

where we used equations C.15 and C.21 and where $\bar{\delta}_t^{(i)} = \hat{\delta}_t^{(i)}(\Delta t_k + \Delta t'_{k'})$, $\bar{\delta}_f^{(i)} = \hat{\delta}_f^{(i)}(\Delta f_k + \Delta f'_{k'})$, $\bar{s}_t^{(i)} = \hat{s}_t^{(i)}(\Delta t_k + \Delta t'_{k'})^2$, and $\bar{s}_f^{(i)} = \hat{s}_f^{(i)}(\Delta f_k + \Delta f'_{k'})^2$. Similarly, we have:

$$\mu\uparrow''(c_{kk'}) \propto \mu\downarrow'(c_{kk'})\mu\uparrow(c_{kk'}) = \mu\downarrow'(c_{kk'})g_{\mathcal{N}}(c_{kk'};\hat{\theta}^{(i)}) \quad \text{(C.27)}$$

$$= \mu\downarrow'(c_{kk'})\Big(\mathcal{N}\big(t'_{k'}-t_k;\bar{\delta}_t^{(i)},\bar{s}_t^{(i)}\big)\,\mathcal{N}\big(f'_{k'}-f_k;\bar{\delta}_f^{(i)},\bar{s}_f^{(i)}\big)\Big)^{c_{kk'}}. \quad \text{(C.28)}$$

The messages $\mu\uparrow'(c_{kk'})$ and $\mu\uparrow''(c_{kk'})$ depend on the messages $\mu\downarrow''(c_{kk'})$ and $\mu\downarrow'(c_{kk'})$, respectively. The latter are computed as follows:

$$\mu\downarrow'(c_{kk'}) \propto \Big(\delta[c_{kk'}-1]+\delta[c_{kk'}]\max\big(\mu\downarrow(b_k=1)/\mu\downarrow(b_k=0), \max_{\ell'\neq k'}\mu\uparrow'(c_{k\ell'}=1)/\mu\uparrow'(c_{k\ell'}=0)\big)\Big) \quad \text{(C.29)}$$

$$= \Big(\delta[c_{kk'}-1]+\delta[c_{kk'}]\max\big(\beta, \max_{\ell'\neq k'}\mu\uparrow'(c_{k\ell'}=1)/\mu\uparrow'(c_{k\ell'}=0)\big)\Big), \quad \text{(C.30)}$$

where we used equations C.12 and C.19, and

$$\mu\downarrow''(c_{kk'}) \propto \Big(\delta[c_{kk'}-1] + \delta[c_{kk'}] \max\big(\mu\downarrow(b'_k=1)/\mu\downarrow(b'_k=0), \max_{\ell\neq k} \mu\uparrow''(c_{\ell k'}=1)/\mu\uparrow''(c_{\ell k'}=0)\big)\Big) \tag{C.31}$$

$$= \Big(\delta[c_{kk'}-1] + \delta[c_{kk'}] \max\big(\beta, \max_{\ell\neq k} \mu\uparrow''(c_{\ell k'}=1)/\mu\uparrow''(c_{\ell k'}=0)\big)\Big). \tag{C.32}$$

The messages $\mu\downarrow''(c_{kk'})$ and $\mu\downarrow'(c_{kk'})$ depend on $\mu\uparrow'(c_{kk'})$ and $\mu\uparrow''(c_{kk'})$, and vice versa (as we pointed out earlier). Therefore, a natural way to determine all of those messages is to first initialize $\mu\downarrow'(c_{kk'}) = 1 = \mu\downarrow''(c_{kk'})$ and then iterate the updates, equations C.26 to C.30, until convergence. This can also be understood from Figure 8: since the graph is cyclic, the max-product algorithm becomes an iterative procedure.

After convergence or after a fixed number of iterations, we compute the marginals $p(c_{kk'})$ as follows:

$$p(c_{kk'}) \propto \mu\downarrow(c_{kk'})\mu\uparrow(c_{kk'}) \tag{C.33}$$

$$= \mu\downarrow'(c_{kk'})\mu\downarrow''(c_{kk'}) \cdot \Big(\mathcal{N}\big(t'_{k'} - t_k; \bar{\delta}_t^{(i)}, \bar{s}_t^{(i)}\big)\, \mathcal{N}\big(f'_{k'} - f_k; \bar{\delta}_f^{(i)}, \bar{s}_f^{(i)}\big)\Big)^{c_{kk'}}, \tag{C.34}$$

where we used equations C.15 and C.24. The decisions $\hat{c}_{kk'}$ are then obtained as

$$\hat{c}_{kk'} = \mathrm{argmax}_{c_{kk'}}\, p(c_{kk'}). \tag{C.35}$$

Acknowledgments

Results of this work were in part reported in Dauwels, Vialatte, Rutkowski, & Cichocki (2007). We wish to thank Zhe ("Sage") Chen (Harvard University), Kenji Morita (RIKEN Brain Science Institute), Yuichi Sakumura (NAIST), Carlos Rodriguez (University at Albany), Sujay Sanghavi (MIT), and Yuki Tsukada (NAIST) for helpful discussions. We are grateful to participants of the retreat of the MIT Picower Center for Learning and Memory (May 2007, Cape Cod, MA, U.S.A.), the RIKEN Symposium "Brain Activity and Information Integration" (September 2007, RIKEN, Saitama, Japan), and the NIPS Workshop "Large-Scale Brain Dynamics" (December 2007, Whistler, Canada) for numerous inspiring questions and comments. We also acknowledge the anonymous reviewers for helpful comments and

Monique Maurice (RIKEN) for designing several figures in this letter. J.D. is deeply indebted to Shun-ichi Amari (RIKEN Brain Science Institute) and Andi Loeliger (ETH Zurich) for continuing support and encouragement.

J.D. was in part supported by postdoctoral fellowships from the Japanese Society for the Promotion of Science, the King Baudouin Foundation, and the Belgian American Educational Foundation. T.W. was in part supported by NSF grant DMI-0447766. Part of this work was carried out while J.D. and T.W. were at the RIKEN Brain Science Institute, Saitama, Japan.

References

Alder, B., Fernbach, S., & Rotenberg, M. (Eds.). (1972). *Seismology: Surface waves and earth oscillations.* New York: Academic Press.

Bayati, M., Borgs, C., Chayes, J., & Zecchina, R. (2008). On the exactness of the cavity method for weighted b-matchings on arbitrary graphs and its relation to linear programs. *Journal of Statistical Physics*, L06001.1–L06001.10.

Bayati, M., Shah, D., & Sharma, M. (2005). Maximum weight matching via max-product belief propagation. In *Proc. 2005 IEEE International Symposium on Information Theory (ISIT 2005)* (pp. 1763–1767). Piscataway, NJ: IEEE.

Bertsekas, D., & Tsitsiklis, J. (1989). *Parallel and distributed computation: Numerical methods*. Englewood Cliffs NJ: Prentice Hall.

Bezdek, J., & Hathaway, R. (2002). Some notes on alternating optimization. In *Proc. AFSS Int. Conference on Fuzzy Systems* (pp. 187–195). Berlin: Springer.

Bezdek, J., Hathaway, R., Howard, R., Wilson, C., & Windham, M. (1987). Local convergence analysis of a grouped variable version of coordinate descent. *Journal of Optimization Theory and Applications, 54*(3), 471–477.

Buzsáki, G. (2006). *Rhythms of the brain*. New York: Oxford University Press.

Candès, E. J., & Donoho, D. L. (2002). New tight frames of curvelets and optimal representations of objects with piecewise-C2 singularities. *Comm. Pure Appl. Math., 57*, 219–266.

Candès, E., Romberg, J., & Tao, T. (2006). Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency estimation. *IEEE Trans. Information Theory*, *52*, 489–509.

Chapman, R., Nowlis, G., McCrary, J., Chapman, J., Sandoval, T., Guillily, M., et al. (2007). Brain event-related potentials: Diagnosing early-stage Alzheimer's disease. *Neurobiol. Aging, 28*, 194–201.

Chen, Z., Ohara, S., Cao, J., Vialatte, F., Lenz, F., & Cichocki, A. (2007). Statistical modeling and analysis of laser-evoked potentials of electrocorticogram recordings from awake humans. *Computational Intelligence and Neuroscience*, 10479. Available online at http://www.hindawi.com/GetArticle.aspx?doi=10.1155/2007.104.

Cichocki, A., Shishkin, S., Musha, T., Leonowicz, Z., Asada, T., & Kurachi, T. (2005). EEG filtering based on blind source separation (BSS) for early diagnosis of Alzheimer's disease. *Clin. Neurophys., 116*, 729–737.

Cui, J., & Wong, W. (2006). The adaptive chirplet transform and visual evoked potentials. *IEEE Transactions on Biomedical Engineering*, *53*(7), 1378–1384.

Cui, J., Wong, W., & Mann, S. (2005). Time-frequency analysis of visual evoked potentials using chirplet transform. *Electronic Letters, 41*(4), 217–218.

Dauwels, J., Tsukada, Y., Sakumura, Y., Ishii, S., Aoki, K., Nakamura, T., et al. (in press). *On the synchrony of morphological and molecular signaling events in cell migration*. In *Proc. International Conference on Neural Information Processing*. N.p.

Dauwels, J., Vialatte, F., & Cichocki, A. (2008). *A comparative study of synchrony measures for the early diagnosis of Alzheimer's disease based on EEG*. Manuscript submitted for publication.

Dauwels, J., Vialatte, F., Rutkowski, T., & Cichocki, A. (2007). Measuring neural synchrony by message passing. In J. C. Platt, D. Koller, Y. Singer, & S. Rowels. *Advances in neural information processing systems, 20*. Cambridge, MA: MIT Press.

Delprat, N., Escudié, B., Guillemain, P., Kronland-Martinet, R., Tchamitchian, P., & Torrésani, B. (1992). Asymptotic wavelet and Gabor analysis: Extraction of instantaneous frequencies. *IEEE Trans. Information Theory*, *38*, 644–664.

Demanet, L., & Ying, L. (2007). Wave atoms and sparsity of oscillatory patterns. In *Appl. Comput. Harmon. Anal., 23*, 368–387.

Donoho, D. (2006). Compressed sensing. *IEEE Trans. Information Theory*, *52*(4), 1289–1306.

Donoho, D., Tsaig, I., Drori, I., & Stark, J-C. (2006). *Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit* (Tech. Rep. 2006.2). Palo Alto, CA: Department of Statistical, Stanford University.

Duarte, M. F., Wakin, M. B., & Baraniuk, R. G. (2005). Fast reconstruction of piecewise smooth signals from random projections. In *Proc. of the Workshop on Signal Processing with Adaptive Sparse Structured Representations*. Piscataway, NJ: IEEE Press.

Edmonds, J., & Karp, R. (1972). Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM, 19*(2), 248–264.

Freeman, W. T., & Weiss, Y. (1999). *On the fixed points of the max-product algorithm*. (Tech. Rep. 1999-039). Cambridge, MA: Mitsubishi Electric Research Laboratories.

Gerards, A. M. H. (1995). *Matching: Handbooks in operations research and management science*. 7, Dordrecht: North-Holland.

Gilbert, A. C., Strauss, M. J., Tropp, J., & Vershynin, R. (2006). *Algorithmic linear dimension reduction in the ℓ_1 norm for sparse vectors*. In *Proc. 44th Annual Allerton Conf. Communication, Control, and Computing.* N.p.

Goupillaud, P., Grossman, A., & Morlet, J. (1984). Cycle-octave and related transforms in seismic signal analysis. *Geoexploration, 23*, 85–102.

Harris, F. J. (2004). *Multirate signal processing for communication systems.* Upper Saddle River, NJ: Prentice Hall.

Herrmann, C. S., Grigutsch, M., & Busch, N. A. (2005). EEG oscillations and wavelet analysis. In T. Handy (Ed.), *Event-related potentials: A methods handbook* (pp. 229–259). Cambridge, MA: MIT Press.

Hogan, M. J., Swanwick, G. R. J., Kaiser, J., Rowan, M., & Lawlor, B. (2003). Memory-related EEG power and coherence reductions in mild Alzheimer's disease. *Int. J. Psychophysiol. 49*, 147–163.

Huang, B., & Jebara, T. (2007). Loopy belief propagation for bipartite maximum weight b-matching. In *Proc. Eleventh International Conference*

on Artificial Intelligence and Statistics (AISTATS). Available online at http://www.stat.umn.edu/aistat/proceedings/start.htm.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., et al. (1998). The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 454*(1971), 903–995.

Jeong, J. (2004). EEG dynamics in patients with Alzheimer's disease. *Clinical Neurophysiology, 115*, 1490–1505.

Jordan, M. I. (Ed.). (1999). *Learning in graphical models*. Cambridge, MA: MIT Press.

Kantha, L., & Clayson, C. (2000). *Numerical models of oceans and oceanic processes*. Orlando, FL: Academic Press.

Loeliger, H.-A. (2004). An introduction to factor graphs. *IEEE Signal Processing Magazine, 21*, 28–41.

Loeliger, H.-A., Dauwels, J., Hu, J., Korl, S., Li, Ping, & Kschischang, F. (2007). The factor graph approach to model-based signal processing. *Proceedings of the IEEE, 95*(6), 1295–1322.

Mallat, S. (1999). *A wavelet tour of signal processing*. Orlando, FL: Academic Press.

Mallat, S., & Zhang, Z. (1993). Matching pursuit with time-frequency dictionaries. *IEEE Transactions on Signal Processing, 41*(12), 3397–3415.

Martin, C., Gervais, R., Hugues, E., Messaoudi, B., & Ravel, N. (2004). Learning modulation of odor-induced oscillatory responses in the rat olfactory bulb: A correlate of odor recognition? *Journal of Neuroscience, 24*(2), 389–397.

Matsuda, H. (2001). Cerebral blood flow and metabolic abnormalities in Alzheimer's disease. *Ann. Nucl. Med., 15*, 85–92.

Matthew, B., & Cutmore, T. (2002). Low-probability event-detection and separation via statistical wavelet thresholding: An application to psychophysiological denoising. *Clinical Neurophysiology, 113*, 1403–1411.

Mitra, P., & Pesaran, B. (1999). Analysis of dynamic brain imaging data. *Biophysical Journal, 76*, 2, 691–708.

Musha, T., Asada, T., Yamashita, F., Kinoshita, T., Chen, Z., Matsuda, H., et al. (2002). A new EEG method for estimating cortical neuronal impairment that is sensitive to early stage Alzheimer's disease. *Clin. Neurophysiol, 113*, 1052–1058.

Nunez, P., & Srinivasan, R. (2006). *Electric fields of the brain: The neurophysics of EEG*. New York: Oxford University Press.

Ohara, S., Crone, N. E., Weiss, N., & Lenz, F. A. (2004). Attention to a painful cutaneous laser stimulus modulates electrocorticographic event-related desynchronization in humans. *Clinical Neurophysiology, 115*, 1641–1652.

O'Neill, J. C., Flandrin, P., & Karl, W. C. (2002). Sparse representations with chirplets via maximum likelihood estimation. *Physical Review E, 65*, 041903.1–041903.14.

Pereda, E., Quian Quiroga, R. Q., & Bhattacharya, J. (2005). Nonlinear multivariate analysis of neurophsyiological signals. *Progress in Neurobiology, 77*, 1–37.

Pulleyblank, W. (1995). *Matchings and extension: Handbook of combinatorics*. Dordrecht: North-Holland.

Quian Quiroga, R. Q., Kraskov, A., Kreuz, T., & Grassberger, P. (2002). Performance of different synchronization measures in real data: A case study on EEG signals. *Physical Review E, 65*, 041903.1–041903.14.

Sanghavi, S. (2007). Equivalence of LP relaxation and max-product for weighted matching in general graphs. In *Proc. IEEE Information Theory Workshop*. Piscataway, NJ: IEEE Press.

Sanghavi, S. (2008). Linear programming analysis of loopy belief propagation for weighted matching. In J. C. Platt, D. Koller, Y. Singer, & S. Roweis (Eds.), *Advances in neural information processing systems, 20*. Cambridge, MA: MIT Press.

Sarvotham, S., Baron, D., & Baraniuk, R. G. (2006). *Compressed sensing reconstruction via belief propagation*. (Tech. Rep. ECE-06-01). Houston, TX: Electrical and Computer Engineering Department, Rice University.

Stam, C. J. (2005). Nonlinear dynamical analysis of EEG and MEG: Review of an emerging field. *Clinical Neurophysiology, 116*, 2266–2301.

Tallon-Baudry, C., Bertrand, O., Delpuech, C., & Pernier, J. (1996). Stimulus specificity of phase-locked and non-phase-locked 40Hz visual responses in human. *Journal of Neuroscience, 16*, 4240–4249.

Thomson, D. J. (1982). Spectrum estimation and harmonic analysis. *Proceedings of the IEEE, 70*, 1055–1096.

Tropp, J., & Gilbert, A. C. (2007). Signal recovery from partial information via orthogonal matching pursuit. *IEEE Transactions on Information Theory, 53*, 4655–4666.

Uhlhaas, P., & Singer, W. (2006). Neural synchrony in brain disorders: Relevance for cognitive dysfunctions and pathophysiology. *Neuron, 52*, 155–168.

Vialatte, F., Cichocki, A., Dreyfus, G., Musha, T., Rutkowski, T. M., & Gervais, R. (2005). Blind source separation and sparse bump modelling of time-frequency representation of EEG signals: New tools for early detection of Alzheimer's disease. In *Proc. IEEE Workshop on Machine Learning for Signal Processing* (pp. 27–32). Piscataway, NJ: IEEE Press.

Vialatte, F. (2005). *Modélisation en bosses pour l'analyse des motifs oscillatoires reproductibles dans l'activité de populations neuronales: Applications à l'apprentissage olfactif chez l'animal et à la détection précoce de la maladie d'Alzheimer*. Unpublished doctoral dissertation, Paris VI University.

Vialatte, F., Martin, C., Dubois, R., Haddad, J., Quenet, B., Gervais, R. et al. (2007). A machine learning approach to the analysis of time-frequency maps, and its application to neural dynamics. *Neural Networks, 20*, 194–209.

Völkers, M., Loughrey, C. M., MacQuaide, N., Remppis, A., de George, B., Koch, W. J., et al. (2007). S100A1 decreases calcium spark frequency and alters their spatial characteristics in permeabilized adult ventricular cardiomyocytes. *Cell Calcium, 41*, 135–143.

Womelsdorf, T., Schoffelen, J. M., Oostenveld, R., Singer, W., Desimone, R., Engel, A. K., et al. (2006). Modulation of neuronal interactions through neuronal synchronization. *Science, 316*, 1609–1612.

Received November 4, 2008; accepted February 3, 2009.