Automatic Learning of Synchrony in Neuronal Electrode Recordings

David Picado Muiño and Christian Borgelt

European Centre for Soft Computing, Gonzalo Gutiérrez Quirós s/n, 33600 Mieres (Asturias), Spain

Abstract

Synchrony among neuronal impulses (or spikes) plays, according to some of the most prominent neural coding hypotheses, a central role in information processing in biological neural networks. When dealing with multiple electrode recordings (i.e., spike trains) modelers generally characterize synchrony by means of a maximal time span (since *exact* spike-time coincidences cannot be expected): two or more spikes are regarded as synchronous if they lie from each other within a distance at most this maximal time span. Such time span is determined by the modeler and there is no agreement about how long it should be. In this paper we present methodology to *learn* this time span automatically from spike-train data that involves the assessment of the amount of synchrony in the database (relative to that expected if spike trains in it were uncorrelated) and a learning process that looks at the time span that maximizes it (over all those considered).

Keywords: Synchronous spiking, parallel spike trains, parallel point processes, multiple electrode recordings, synchrony in spike-train databases, automatic learning

1. Introduction

The principles of neural coding and information processing in biological neural networks are still not well understood and are the topic of ongoing research. Several coding schemes have been proposed in the neuroscience literature (i.e., models about how information is represented in biological neural networks). One of the most prominent coding hypotheses is the so-called *temporal coordination scheme* (see, e.g., [1]), first advocated by D.O. Hebb [8] and driven by more recent physiological and anatomical evidence, according to which the coordinated emission of *spikes* (i.e., electrical impulses or action potentials), in particular *synchronous* spiking (see, e.g., [2, 6, 9]) by groups of neurons plays a major role in neuronal information processing. Such a view is in keeping with the so-called *synfire chain* (or synchronous firing chain) model: a feed-forward neural network in which neuronal impulses propagate back and forth from different synchronous groups of neurons (which we call *neuronal assemblies* and characterize as groups of neurons that fire synchronously *significantly* more often than it would be expected by chance under the assumption of independence in their respective neuronal impulses).

In order to understand the principles of coordinated neuronal activity and neural coding it is necessary to observe the activity of multiple neurons simultaneously. This is nowadays possible due to improvements in multiple-electrode recording and spike sorting (see, e.g., [4, 12]), which allows to monitor the activity of hundreds of neurons. Such recordings are typically represented by sequences of points in time (i.e., point processes) and referred to as *parallel spike trains*.

Determining what constitutes synchronous spiking in parallel spike trains is not a trivial issue, since *exact* spike-time coincidences cannot be expected (due to the underlying neural processes themselves or to limitations in the recording technology). Spike synchrony is typically characterized in the field in relation to a certain time span, say ω: two or more spikes are considered to be synchronous if they all lie within a distance at most ω from each other. The distance ω is normally determined from the outset by the modeler, i.e., he/she chooses ω according to what he/she considers the maximal time span for a set of synchronous spikes. However, since the mechanisms of neural processing are still under intense discussion and thus there is no obvious choice as to what value ω should be given, a method that *learns* ω automatically from data is clearly desirable. In this paper we offer methodology to do so.

Learning the *right* time span ω to assess spyke synchrony is certainly important, particularly when trying to identify assembly activity (i.e., significant spike synchrony). A time span ω shorter than that necessary to identify (all or almost all) synchronous spiking of a neuronal assembly may not be sufficient to identify the assembly itself (i.e., to identify such group of neurons as *significantly* synchronous). On the other hand, a time span ω too large (i.e., *much* larger than that necessary to identify all synchronous spiking of the assembly) may *dilute* the assembly and make it look like any other group of uncorrelated neurons.

In order to learn ω we employ measures of *(relative) synchrony* for sets of parallel spike trains—relative to the amount of synchrony observed and/or expected in sets of uncorrelated spike trains. Such measures we conceive as being computed from

the amounts of synchrony of individual neuronal patterns (i.e., strictly speaking, of the spike trains corresponding to individual patterns). Synchrony of neuronal patterns is assessed by means of our continuous model of synchrony that is based on the above characterization of spike synchrony. This model, which we briefly review in this paper, was first introduced in [13].

Once we have a measure of synchrony, we learn ω for a particular set of spike trains by measuring its synchrony with respect to distinct candidate values for ω and choose that value which satisfies certain criteria (e.g., as we will see later, the one that maximizes synchrony with respect to any of the measures that we define).

2. Notation and preliminary definitions

Let N be our set of neurons. We will be working with parallel spike trains, one for each neuron in N, formalized as spike-time sequences (or point processes) of the form $\{t_1^a, ..., t_{k_a}^a\} \subset (0, T]$, for $a \in N$ and $T \in \mathbb{R}$ (the recording time in seconds), where k_a is the number of times neuron a fires in the interval $(0, T]$. We denote the set of all these sequences by $\mathcal{S}$. Sets of sequences like $\mathcal{S}$ constitute our raw data.

When formalizing our continuous model of synchrony we will be using *multisets* (i.e., collections of elements that can contain multiple copies of the same element, see e.g. [15]), which are formally defined as pairs $\langle X, h\rangle$, where X is a set and $h : X \to \mathbb{N} \cup \{0\}$. Intuitively, the function h *counts* the occurrences of the elements of X in $\langle X, h\rangle$. Throughout this paper, whenever convenient, we use set-like notation for multisets: for example, $\{x_1, x_1, x_2, x_2, x_3\}$ instead of $\langle X, h\rangle$, with $h(x_1) = 2$, $h(x_2) = 2$, $h(x_3) = 1$ and $h(x) = 0$ for any $x \in X \setminus \{x_1, x_2, x_3\}$.

3. Assessment of spike and spike-train synchrony

We assume the characterization of spike synchrony outlined in Section 1: two or more spikes are considered to be synchronous if they all lie within a certain distance, say ω, from each other.

Such characterization of spike synchrony is the one intended (yet not achieved) by the bin-based model, the almost exclusively applied model of synchrony in the field. The bin-based model relies on time discretization to assess spike synchrony: the recording time is partitioned into time intervals (bins) of equal length (e.g., ω). In this model, two or more spikes are considered to be synchronous if they lie in the same time bin. Note however that, in such a characterization, any two spikes that lie in distinct time bins will not be regarded as synchronous, even if they are separated by a time distance way smaller than ω (this we call the *boundary problem*).

We briefly review here a new synchrony model (first introduced in [13]), which we call *continuous model*, that builds on the notion of spike synchrony that we assume in this paper and that overcomes the problems derived from time discretization in the bin-based model (i.e., the boundary problem).

We formally characterize spike synchrony in our continuous model by means of the operator *Sync* which, for G a multiset over $(0, T]$ and $\omega \in \mathbb{R}^+$ the time span for spike synchrony, we define as follows:

$$Sync(G) = \begin{cases} 1 & \text{if } \max\{|t_i - t_j| \,|\, t_i, t_j \in G\} \leq \omega \\ 0 & \text{otherwise.} \end{cases}$$

We denote by E^A the collection of those multisets G in $\mathcal{S}$ that contain exactly one spike for each neuron in $A \subseteq N$ such that $Sync(G) = 1$. These multisets we call synchronous events of A in $\mathcal{S}$ or, in short, A-events.

We move now to the definition of the operator *Supp*, aimed at assessing spike-train synchrony (which we also call *support*) in our continuous model. Our characterization of synchrony among spike trains corresponding to a subset of neurons $A \subseteq N$ (in short, support of A) basically takes into account the amount of A-events in E^A, with the further restriction that at most one A-event per spike is considered in the summation. Notice though that this further restriction in itself is not sufficient to define *Supp* since we could possibly have several subsets of A-events in E^A that satisfy it. In order to discriminate among all these possible subsets of A-events we opt for a maximization criterion: we consider that (or those) which maximize the amount of synchrony.

Formally, let $A \subseteq N$. We define $\mathcal{H}$ as the collection of all subsets $H \subseteq E^A$ with at most one A-event per spike. We define the support operator *Supp* as follows:

$$Supp(A) = \max_{H \in \mathcal{H}} \{|H|\} \tag{1}$$

As is explained in detail in [13], the assessment of $Supp(A)$ as defined by Equation (1) can be seen as an instance of the *maximum independent subset* problem, which consists of finding an independent subset of maximal cardinality (i.e., that there is no other independent subset of higher cardinality): in our context, a maximal subset $H \subseteq E^A$ of independent A-events, where by independent A-events we mean A-events that do not share spikes. This problem is, for the general case, NP-complete. However, the resulting constraints of our particular problem instance allow an efficient solution by applying a greedy algorithm that linearly traverses the data and always selects the next synchronous group of spikes that does not share spikes with a previously selected group.

4. Synchrony in spike-train databases

We are interested in measures of synchrony of $\mathcal{S}$ that are sensitive to the time span ω for spike synchrony (i.e., measures of synchrony relative to ω). In particular, we look for measures of synchrony that build on ω and that are thus assessed from the amounts of synchrony observed for individual sets of neurons.

We start by identifying all *frequent patterns* $A \subseteq N$ in $\mathcal{S}$, by which we mean sets of neurons of at least a certain size $z_{\min}$ whose support in $\mathcal{S}$ (assessed by means of our operator *Supp*) is at least a certain threshold $c_{\min}$. Typically we choose $z_{\min}$, $c_{\min}$ equal to 2, i.e., patterns with at least two neurons and two occurrences in the database.

We compute synchrony of $\mathcal{S}$ by looking at the frequent patterns found. For each pattern, we take into account its size (z) and support (c), i.e., its signature $\langle z, c\rangle$. We denote by $sig(\mathcal{S})$ the set of signatures of all frequent patterns found in $\mathcal{S}$.

4.1. Frequent item set mining (FIM)

In order to help making as efficient as possible the identification of all frequent neuronal patterns $A \subseteq N$ in $\mathcal{S}$ we can take advantage of frequent item set mining methodology and data structures (see, e.g., [7]). For our support operator *Supp* the *anti-monotonicity* property holds: for $A, B \subseteq N$ and $A \subset B$, the amount of synchrony (i.e., support) of A is greater than or equal to the amount of synchrony of B. The anti-monotonicity of our support operator implies the so-called *apriori* property exploited by FIM algorithms in order to prune the search for frequent patterns: that no superset of an infrequent set of neurons can be frequent, where frequency is determined with respect to a support threshold $\sigma_{\min}$ (below which a set is considered infrequent). This holds for our support operator and, more generally, in maximum independent subset (or, in a graph interpretation, node set) approaches, as shown in [5] or [16].

Full details on FIM implementation in our domain can be found in [3].

4.2. Data randomization

We look at measures of synchrony for $\mathcal{S}$-like databases that are *relative* to the independence assumption among spike trains in $\mathcal{S}$. Such measures we typically express as a ratio between the (absolute) amount of synchrony found in $\mathcal{S}$ and that expected in independent data. In order to estimate the amount of synchrony in independent data we employ surrogate generation techniques (i.e., data randomization techniques): modifications of the original data $\mathcal{S}$ that are intended to keep all its essential features except synchrony, the feature we are testing—see, e.g., [11].

Generally speaking, surrogate generation techniques are particularly suitable when the underlying probabilistic model of the data is not known and is difficult to assess. When testing for significance (e.g., when assessing whether the amount of synchrony of our spike-train data set is significant or, on the contrary, it is the amount one would expect to find if spike trains were independent), data randomization allows us to represent the *null hypothesis* (of independence) implicitly: we compare the results on the original data to those of sufficiently many surrogate data sets. More specifically, from all the results in the surrogate data sets we select, depending on the desired significance level, a critical value above which we declare the result significant (e.g., in 1000 surrogate data sets, at a significance level of 1%, we would pick the 11^{th} highest value and would consider the amount of excess synchrony in our original data set significant if it lies above such value).

In what follows we will not be that concerned about significance as we are about assessing the amount of synchrony expected in independent spike-train data sets. In order to assess such expected amount of synchrony we average results over sufficiently many surrogate data sets (in the examples shown in our preliminary evaluation of our learning methodology in this paper, over 100 surrogate data sets).

4.3. Measures of synchrony

As was mentioned above, we look for measures of synchrony that are relative to the expected amount of synchrony under the assumption of uncorrelated spike trains. We express such measures as the ratio between the (absolute) amount of synchrony in our spike-train data and that expected under the independence assumption.

The measures we look at are sensitive to the time span ω for spike synchrony that is chosen for our continuous model when looking for frequent patterns in collections of spike trains.

Let m_ω be a measure of synchrony, defined with respect to the time span ω for synchronous spiking, and let $\mathcal{S}$ be a collection of spike trains. We will see values greater than 1 for $m_\omega(\mathcal{S})$ as an indication of excess synchrony—compared to independent spiking—and values smaller than 1 as an indication of a lack of synchrony (we are not particularly concerned in this paper about the significance of such excess or lack of synchrony).

We assign weights (i.e., magnitudes) to signatures that will be later employed in the definition of our synchrony measures. For $\langle z, c\rangle$ a signature we denote by $\rho(\langle z, c\rangle)$ its weight, with ρ the weight operator. The characterization $\rho(\langle z, c\rangle) = z * c$ is probably the most intuitive one when thinking about the magnitude of a signature (note that such a magnitude corresponds to the amount of spikes that are involved in a pattern that has this signature). How-

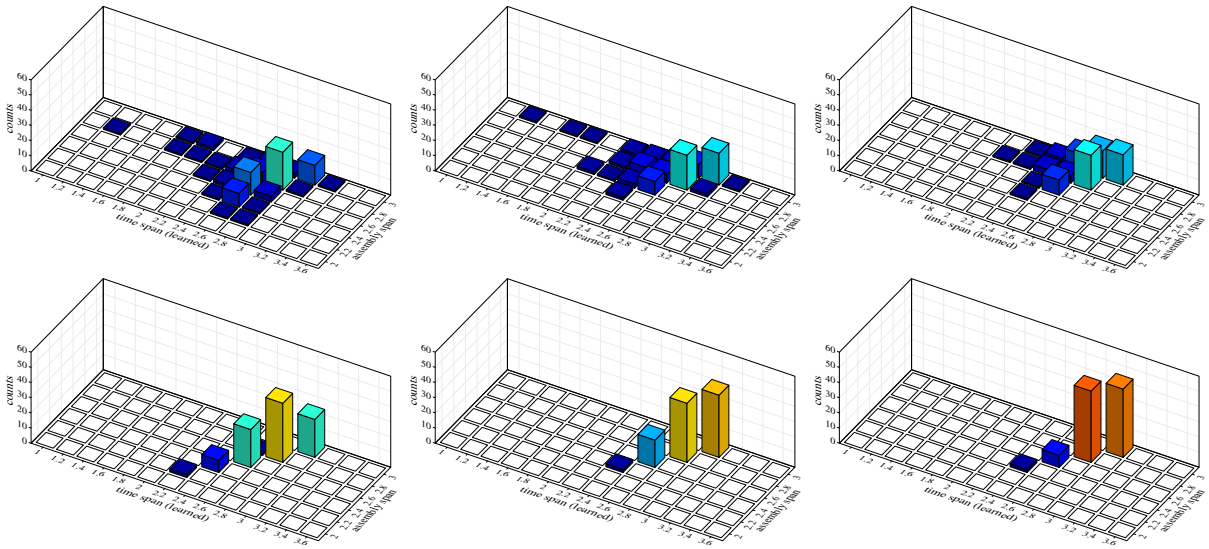

Figure 1: Counts over 100 trials. **First row**: measure m^1_ω. Assemblies consisting of a pattern of signature $\langle 6,6\rangle$ (first diagram), $\langle 7,7\rangle$ (second diagram) and $\langle 8,8\rangle$ (last diagram) injected. **Second row**: measure m^2_ω. Assemblies consisting of a pattern of signature $\langle 6,6\rangle$ (first diagram), $\langle 7,7\rangle$ (second diagram) and $\langle 8,8\rangle$ (last diagram) injected.

ever, since our focus is on coincident spikes and patterns of size $z = 1$ show no coincident spikes with other neurons, we normally consider the alternatives $\rho(\langle z,c\rangle) = (z-1)\cdot(c)$ and $\rho(\langle z,c\rangle) = (z-1)\cdot(c-1)$ (the latter is justified by the fact that we do not count as frequent patterns those with a single coincidence).

We consider here some families of measures, for distinct values $\omega \in \mathbb{R}^+$ and for $Surr$ the set of surrogate data sets obtained from original data:

- Summation of weights of all signatures found in original data divided by the expected summation of weights under the assumption of independence (which we denote by Exp and assess from $Surr$):

$$m^1_\omega(\mathcal{S}) = \frac{\sum_{\langle z,c\rangle \in sig(\mathcal{S})} \rho(\langle z,c\rangle)}{Exp}$$

  where Exp is estimated from $Surr$ as follows:

$$Exp = \sum_{\substack{\langle z,c\rangle \in sig(\mathcal{S}')\\ \mathcal{S}' \in Surr}} \frac{\rho(\langle z,c\rangle)}{|Surr|}$$

- Maximum signature weight in $\mathcal{S}$ divided by the expected maximum signature weight under the assumption of independence (denoted by Exp and assessed from the set $Surr$):

$$m^2_\omega(\mathcal{S}) = \frac{\max_{\langle z,c\rangle \in sig(\mathcal{S})} \rho(\langle z,c\rangle)}{Exp}$$

  where Exp is estimated by averaging over $Surr$ as follows:

$$Exp = \sum_{\mathcal{S}' \in Surr} \frac{\max_{\langle z,c\rangle \in sig(\mathcal{S}')} \rho(\langle z,c\rangle)}{|Surr|}$$

5. Learning spike synchrony

For $W \subseteq \mathbb{R}^+$ a set of candidate values for the time span ω that defines spike synchrony and m^i_ω the selected measure of synchrony, for $i \in \{1,2\}$, we choose $\omega \in W$ (i.e., the time span to define spike synchrony learned from data, which we denote by ω_L) such that, for all $\omega \in W$, the following holds, for $\mathcal{S}$ a collection of spike trains:

$$m_{\omega_L}(\mathcal{S}) \geq m_\omega(\mathcal{S}).$$

In words, we choose the time span ω that maximizes $m^i_\omega(\mathcal{S})$.

Note that the (absolute) amount of synchrony of any spike-train data set, whether measured as the summation over the weights of all signatures found in the data set (as in m^1_ω) or simply as the maximum signature weight (as in m^2_ω), should increase for increasing values of ω (since a larger ω implies that sets of spikes that are not considered synchronous with a smaller ω become synchronous, increasing the support and possibly also the size of the pattern). This, as we just mentioned, holds for independent data and also for data with assembly activity (i.e., with significant synchronous spiking). However, for data with assembly activity a steeper increase is expected as long as ω is below the actual time span of the synchronous spiking events that constitute the assembly activity. Such an increase becomes less prominent once ω goes above the actual span of the synchronous spiking events. As a consequence, a maximum value for our relative measures of synchrony over W should be attained for ω equal or very close to the actual span of the assembly activity.

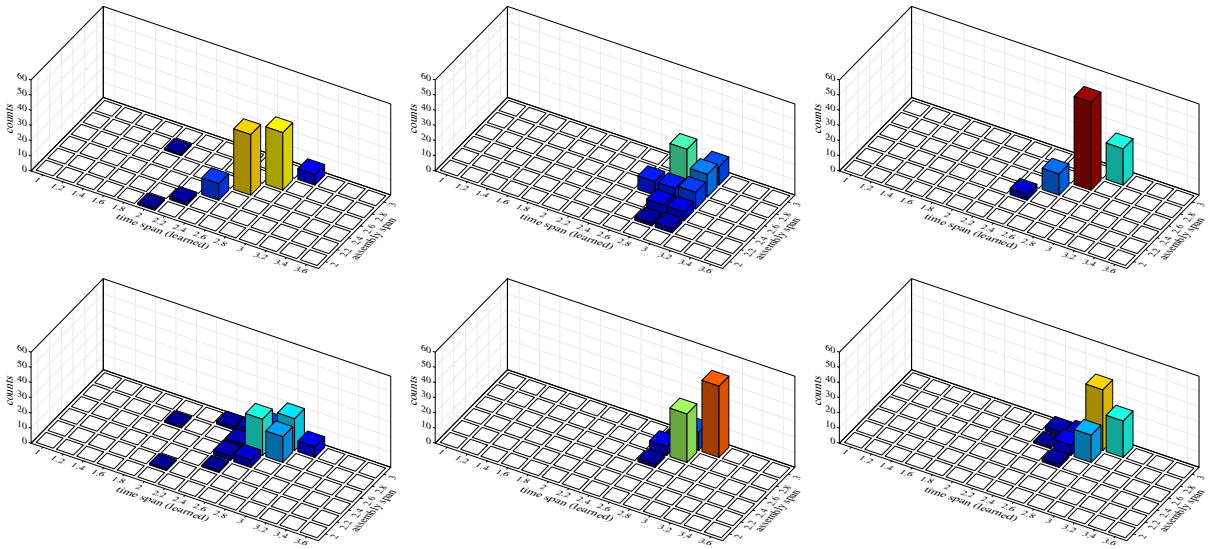

Figure 2: Counts over 100 trials. Two neuronal assemblies. **First row**: ω_L learned with respect to the time span of the first identified assembly (i.e., all its spike coincidences). **Second row:** ω_L learned with respect to the time span of the second identified assembly. **First column:** two injected patterns of signature $\langle 6, 6\rangle$. **Second column:** two injected patterns of signatures $\langle 6, 6\rangle$ and $\langle 8, 8\rangle$. **Third column:** two injected patterns of signature $\langle 8, 8\rangle$.

5.1. Evaluation

In this section we present results of an initial evaluation of the learning process of ω_L on different collections of artificially generated spike-train data. The data sets generated for our evaluation consist of 100 trials with injected assembly activity (100 spike trains in each trial—i.e., $|N| = 100$). In order to generate correlation among spike trains we essentially adopted the basic features of the SIP (Single Interaction Process) model—described in [10]—along with some modifications aimed at generating non-exact coincidences among spikes: for each data set we generated trials of independent spike trains (modeled as Poisson processes with constant rate 20Hz) in order to represent the background activity of neurons in N, each with a time duration of 3 seconds (i.e., $T = 3$). In order to add assembly activity we considered, for each data set, a particular choice of signatures of the form $\langle z, c\rangle$—with z, c in the range $\{6, 7, 8\}$ (i.e., z, c sufficiently small to better assess the effectiveness of our learning process). For given $\langle z, c\rangle$, a random choice of c points in the interval $(0, T]$ was considered for each trial. Each spike generated this way was added to the background spiking activity by adding some random time deviation in order to produce *non-exact* spike coincidences, modeled by means of a uniform random variable on the interval $[-0.0015, 0.0015]$ (i.e., +/- 1.5 millisecond shift, for $\omega = 3$ milliseconds).

Our collections of surrogate data sets (*Surr*) are obtained from our generated spike-train data sets by means of *spike randomization*: from each spike train in a trial we generate a new randomized spike train by uniformly distributing its amount of spikes over the time span $(0, T]$.

Figures 1 and 2 show some results for one and two non-overlapping injected synchronous patterns respectively. All these results are based on the assessment of synchrony (either by m^1_ω or by m^2_ω) in terms of our weight operator $\rho(\langle z, c\rangle) = (z-1)*(c-1)$, for $\langle z, c\rangle$ a signature (results in terms of other weight operators—defined previously in this paper—do not differ substantially from the reported results).

In all diagrams shown in figures 1 and 2, the y-axis features the time span of an injected assembly (i.e., the minimum time span ω—among those tested, with a 0.2-millisecond resolution—that allows us to identify all spike coincidences of the assembly; we call it *assembly* or *coincidence* span). We stress here that *non-exact* spike coincidences are modeled by shifting spikes uniformly over a three-millisecond time span. Thus, the maximal distance between any two spikes of an injected coincidence can well be below 3 milliseconds (the smaller z—i.e., the number of spikes in the coincidence—the more likely it is so, as can be expected and clearly seen from the diagrams). The x-axis corresponds to the time span ω_L (in milliseconds) learned by our process and the z-axis (the height) to the amount of trials counted with respect to the assembly span and the learned ω_L.

Diagrams in Figure 1 show the performance of our learning process in data trials with a single synchronous pattern injected (with signatures $\langle 6, 6\rangle$, $\langle 7, 7\rangle$ and $\langle 8, 8\rangle$). Diagrams in the first row show the assessment of synchrony with the measure m^1_ω and those in the second row show the assessment with m^2_ω. Although in these trials the learning pro-

cess based on m^1_ω works reasonably well (note that in most cases the time span ω_L learned by the process matches the assembly span) it is clearly outperformed by that based on m^2_ω, for which the learned time span ω_L coincides in nearly all trials with the assembly span (to be more precise, in all trials except for a single one with an injected synchronous pattern of signature $\langle 6, 6\rangle$).

In Figure 2, diagrams in the first row feature the time span ω_L learned with respect to the time span of the injected assembly that is first identified by our model (i.e., all its spike coincidences) while those in the second row show the time span ω_L learned by our process in relation to the time span of the assembly whose spike coincidences (i.e., all its spike coincidences) are identified the last. Learning of ω_L in these experiments is based on the measure of synchrony m^2_ω. As is possibly clear, the time spans of the two injected assemblies often differ: as can be seen in the diagrams of the first and third columns (i.e., those corresponding to injected assemblies of the same signature) the time span learned generally coincides with the time span of the first assembly with all its spike coincidences identified, as is seen in the diagrams of the first row. However, as can be seen in the diagrams of the second row, the span ω_L learned may not be enough to identify all spike coincidences of the other assembly. This is not so in trials with two injected assemblies of signatures $\langle 6, 6\rangle$ and $\langle 8, 8\rangle$ (second column). Here the learning process is *dominated* by the *second* assembly—i.e., the last one among the two to have all its spike coincidences identified (very likely that with the signature of largest magnitude—$\langle 8, 8\rangle$). In these trials the learned time span ω_L tends to coincide with the time span of the assembly with (very likely) largest magnitude, which is sufficient to also identify all spike coincidences of the other assembly.

6. Conclusion and future work

In this paper we have presented methodology to automatically learn the time span in order to define spike synchrony—i.e., that distance within which two or more spikes will be regarded as synchronous (since, as was mentioned, exact spike-time coincidences cannot be expected)—from parallel spike trains. Such methodology involves the assessment of the amount of synchrony in spike-train databases, which we approached in this paper by means of the synchrony operators m^1_ω and m^2_ω. Such operators were defined as *relative* measures of synchrony: they express ratios between the (absolute) amount of synchrony of the target or original database and the amount of synchrony that would be expected under the assumption of uncorrelated spike trains (which is assessed by means of surrogate generation methods from the original data). The learning process for the time span to characterize synchrony (which we called ω_L) picks, over a certain set, the span that maximizes synchrony in the original database.

Our methodology was tested on artificially generated spike trains and proved to work very well (particularly with respect to the measure m^2_ω) on the data sets with injected spike coincidences for a single neuronal assembly: the learned time span ω_L from these data sets and the assembly span almost always coincide. Learning ω_L from data sets with two non-overlapping neuronal assemblies worked reasonably well but presented some problems: in some cases the time span ω_L learned was not enough to identify all spike coincidences of one of the assemblies (which in most cases should not constitute a major problem when looking to identify significant spike synchrony, as long as enough coincidences are identified to allow the spiking activity of the assembly to be regarded as significant). In other cases the span ω_L learned was longer than necessary in order to identify all coincidences (which, again, should not be a major problem as long as the learned span is not *much* longer than necessary, which could dilute the significance of the synchronous spiking of the assembly). This said, an *incremental* approach which first learns ω_L for the identification of one of the assemblies and then a new ω_L to detect the other should work fine as long as the assemblies do not overlap (we are currently evaluating this possibility).

Future work

In relation to the contect of this paper, a more thorough evaluation and refinement of our methodology is necessary when dealing with spike-train datasets with multiple assembly activity. We are also working on methodology to assess significance of the amount of synchrony (e.g., of our measures m^1_ω and m^2_ω, among others) in spike-train databases, which would also be used in our learning methodology.

Acknowledgments

The work presented in this paper was partially supported by the Spanish Ministry for Economy and Competitiveness (MINECO Grant TIN2012-31372) and by the Goverment of the Principality of Asturias, (Programa Asturias Grant CT14-05-2-06).

References

[1] Abeles, M. *Local Cortical Circuits: An Electrophysiological Study.* Springer, Berlin, Germany 1982.

[2] Abeles, M. *Role of the Cortical Neuron: Integrator or Coincidence Detector?* Israel Journal of Medical Science, 18 (1982), 83–92.

[3] Borgelt, C. and Picado Muiño, D. *Finding Frequent Patterns in Parallel Point Processes.*

12th Int. Symp. on Intelligent Data Analysis. Springer, LNCS 8207 (2013), 116–126.

[4] Buzśaki, G. *Large-Scale Recording of Neuronal Ensembles.* Nature Neuroscience, 7 (2004), 446–451.

[5] Fiedler, M. and Borgelt, C. *Subgraph Support in a Single Graph.* Proc. IEEE Int. Workshop on Mining Graphs and Complex Data, 399–404. IEEE Press, Piscataway, New Jersey, USA 2007.

[6] Goedeke, S. and Diesmann, M. *The Mechanism of Synchronization in Feed-Forward Neuronal Networks.* New Journal of Physics, 10 (2008).

[7] Goethals, B. *Frequent Set Mining.* Data Mining and Knowledge Discovery Handbook (2nd ed.), 321–338. Springer, Berlin, Germany 2010.

[8] Hebb, D.O. *The Organization of Behavior.* J. Wiley & Sons, New York, USA, 1949.

[9] König, P., Engel, A.K. and Singer, W. *Integrator or Coincidence Detector? The Role of the Cortical Neuron Revisited.* Trends Neuroscience, 19 (1996), 130–137.

[10] Kuhn, A., Aertsen, A. and Rotter, S. *Higher-order Statistics of Input Ensembles and the Response of Simple Model Neurons.* Neural Computation, 15 (2003), 67–101.

[11] Louis, S., Borgelt, C. and Grün, S. *Generation and Selection of Surrogate Methods for Correlation Analysis.* In: Grün, S. and Rotter, S. (Eds.) *Analysis of Parallel Spike Trains.* Springer Series in Computational Neuroscience (2010), 359–382.

[12] Marre, O., Amodei, D., Deshmukh, N., Sadeghi, K., Soo, F., Holy, T.E. and Berry II, M.J. *Mapping a Complete Neural Population in the Retina.* The Journal of Neuroscience, 32/43 (2012), 14859–14873.

[13] Picado Muiño, D. and Borgelt, C. *Frequent Item Set Mining for Sequential Data: Synchrony in Neuronal Spike Trains.* Intelligent Data Analysis, 18/6 (2014), 997–1012.

[14] Picado-Muiño, D., Borgelt, C., Berger, D., Gerstein, G.L. and Grün, S. *Finding Neural Assemblies with Frequent Item Set Mining.* Frontiers in Neuroinformatics, 7/9 (2013).

[15] Syropoulos, A. *Mathematics of Multisets.* In: Calude, C.S., Paun, G., Rozenberg, G. and Salomaa, A. (Eds.). *Multiset Processing: Mathematical, Computer Science and Molecular Computing Points of View.* Springer, LNCS 2235 (2001), 347–358.

[16] Vanetik, N., Gudes, E. and Shimony, S.E. *Computing Frequent Graph Patterns from Semistructured Data.* Proc. IEEE Int. Conf. on Data Mining, 458–465. IEEE Press, Piscataway, New Jersey, USA 2002.