

# KERNEL ANALYSIS FOR ESTIMATING THE CONNECTIVITY OF A NETWORK WITH EVENT SEQUENCES

Taro Tezuka<sup>1</sup>, Christophe Claramunt<sup>2</sup>

<sup>1</sup>*University of Tsukuba, Tsukuba, Japan,  
Email: tezuka@slis.tsukuba.ac.jp*

<sup>2</sup>*Naval Academy Research Institute, Brest, France,  
Email: claramunt@ecole-navale.fr*

## Abstract

Estimating the connectivity of a network from events observed at each node has many applications. One prominent example is found in neuroscience, where spike trains (sequences of action potentials) are observed at each neuron, but the way in which these neurons are connected is unknown. This paper introduces a novel method for estimating connections between nodes using a similarity measure between sequences of event times. Specifically, a normalized positive definite kernel defined on spike trains was used. The proposed method was evaluated using synthetic and real data, by comparing with methods using transfer entropy and the Victor-Purpura distance. Synthetic data was generated using CERM (Coupled Escape-Rate Model), a model that generates various spike trains. Real data recorded from the visual cortex of an anaesthetized cat was analyzed as well. The results showed that the proposed method provides an effective way of estimating the connectivity of a network when the time sequences of events are the only available information.

**Keywords:** connectivity estimation, neural network, kernel methods, spike train

## 1 Introduction

A network where events occur at each node is one of the most common models used for representing spatio-temporal dynamics. Many real world phenomena can be modeled using such an abstract framework. Examples include a social network, a commercial network, a metabolic system, and a biological neural network. They can all be modeled as sequences of events occurring on a certain graph (i.e. network), if simplified enough. We consider the case where the semantic content of each event is unimportant and only the time of its occurrence matters. Such a model is a special case of marked point processes where stochastic dependency be-

tween different nodes is regulated by time varying weights assigned to the edges of the graph.

In many real applications, it often occurs that events at the nodes are observable, but connections between the nodes are not. Estimating how the nodes are connected based on events observed at the nodes is an important research task having many applications. This paper proposes a new way to estimate such connections using kernel methods, which have been widely used elsewhere in machine learning [36]. The effectiveness of kernel methods relies heavily on defining an appropriate positive definite kernel that represents a similarity measure suited for the target data structure. In this paper, we pro-

pose to use a kernel defined on event sequences. The novelty of our proposal lies in applying a positive definite kernel for estimating the connections of a network.

There are already various proposals for estimating the connections of a network, including ones that use Granger causality [13], transfer entropy [35, 43], a directed transfer function [17, 18], directed information [30], dynamic causal modeling [10], a likelihood function [24], and joint entropy [11]. This paper proposes a new approach that uses a positive definite kernel on event sequences. Some of the characteristics of this approach are that it is based on a natural and intuitive framework of Gaussian processes, and that it is extensible to various multivariate analysis techniques.

The rest of the paper is organized as follows. Section 2 provides preliminaries for this work. Section 3 discusses related work while Section 4 introduces our modeling approach. Section 5 shows the experimental results, while Section 6 concludes the papers and draws some conclusion.

## 2 Preliminaries

This Section briefly introduces some of the main concepts.

### 2.1 Connections and connectivity

In terms of graph theory, a connection corresponds to an edge that connects two nodes of a graph. There are two types of connections, namely directed and undirected connections. A directed connection from node  $a$  to node  $b$ , indicated by  $\varepsilon_{a,b}$  does not imply  $\varepsilon_{b,a}$ , but in the case of an undirected connection,  $\varepsilon_{a,b}$  implies  $\varepsilon_{b,a}$ .

The *strength of a connection* is generally associated to the weight assigned to a given connection. Connectivity denotes the property of a network that represents the strength of these connections. Connectivity can be evaluated locally with respect to a given node or overall for the whole network. Locally, it is a mapping from a pair of nodes to a real value, that is,  $f_C : V \times V \rightarrow \mathbb{R}$ , where  $V$  is a set of nodes in a graph  $G(V, E)$ .

### 2.2 Event sequence

An event sequence is a sequence of event times. It is a sorted sequence of numbers belonging to the time range  $[t_0, t_1]$ . It can be considered as a sample of a point process. An event sequence can be of arbitrary length. For example, it could be that  $(0.25, 0.75, 2.00)$  and  $(0.5, 1.50, 3.25, 5.50)$  are both possible event sequences. A *spike train* is an example of event sequence that occurs on a biological neuron.

### 2.3 A graph with events

A graph where events occur stochastically at each node is a data structure that can be used to model various phenomena occurring in nature. An event sequence  $x^i$  is a sequence of time points that represent events that respectively occurred at node  $i$ . When two nodes are connected, the previous node influences the successor node directly. That is, there is direct stochastic dependence between two nodes that cannot be explained away using another node. For example, in the case of a biological neural network, when two nodes (neurons) are connected, an event (spike) at one node directly promote or suppress events (spikes) at the other node. Using the framework of probability theory, it can be expressed that if an event occurs at node  $i$ , events happen with higher probability at nodes connected to  $i$ . As a result, if two nodes are connected, their event sequences tend to be similar. This is why a similarity measure between two event sequences can be used to estimate connectivity between the nodes.

### 2.4 The task of connectivity estimation

The task of estimating connectivity from events observed at the nodes can be formalized as follows:

**Input:** Set of nodes  $V$ . Set of event sequences  $\{x^{(i)}\}_{i \in V}$ .

**Output:** Graph  $G(V, E)$  and weight function  $m : E \rightarrow \mathfrak{R}$  representing connectivity.

For graph  $G(V, E)$ ,  $V$  denotes the set of all nodes and  $E$  is the set of all edges. We use two terms, a node and a neuron, interchangeably, since the biological neural network is our primary example of a network with event sequences.

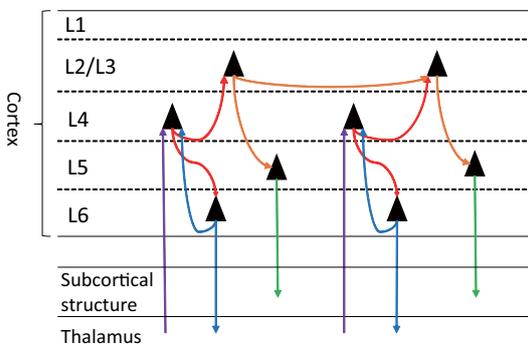
### 2.5 Positive definite kernel

For two geometrical vectors, the angle between the two is a commonly used measure of similarity. In information retrieval, cosine similarity which corresponds to the angle between a document vector and a search query is still widely used [34]. While the angle is defined only for geometrical vectors, it generalizes to the inner product for the general vector space. It can be further generalized to the *positive definite kernel* for data structures that are not a vector space. The positive definite kernel enables to extend various multivariate optimization methods to arbitrary data structures. This is the reason why positive definite kernels are intensively used in machine learning. These extensions are collectively called *kernel methods*. In this paper, a positive definite kernel is defined on event sequences, in order to estimate connectivity between nodes on which event sequences occur.

## 3 Related work

### 3.1 Connectivity estimation

Estimating connectivity of a network has been an active research area [13, 11]. Analyzing cortical neural networks has been one of the prominent examples of this task [10]. Figure 1 illustrates a schematic diagram of connectivity between pyramidal neurons in a sensory area. Although the basic framework is known, the real connectivity largely depends on the actual brain tissue. It is therefore necessary to estimate connectivity based on observed data.



**Figure 1.** Schematic connectivity of neurons in a sensory area of the brain. A triangle represents a neuron. Different colors represent from which node the connection came out of.

Granger causality is one of the most commonly used measures of connectivity estimation in a network [13]. It is defined using a stochastic test on prediction capability using linear regression. Vicente et al. suggested a measure of transfer entropy to evaluate the strength of connection in a network [43]. Since transfer entropy is based on information theory, it has a merit of not depending upon a specific stochastic model. It does not assume linear relationships either. Kaminski et al. proposed a directed transfer function that extends Granger causality for two channels to a large number of channels [17, 18]. Quinn et al. applied directed information for connectivity estimation [30]. This is similar to mutual information but incorporates causal relationships between stochastic processes at each time point. Friston et al. developed dynamic causal modeling and applied it for estimating effective connectivity in a cortical network [10]. Okatan et al. defined a likelihood function for connectivity and proposed a method of inference [24]. Garofalo et al. compared various methods of connectivity estimation and reported that transfer entropy and joint entropy give the best results [11].

Many of these methods are based either on information theory or a general generative model of a stochastic process. These can be applied to the estimation of connectivity in any stochastic phenomena occurring on a network. However, it is often the case that incorporating the characteristics of the given data into the estimation method makes it more fit. Moreover, it would make the result more interpretable, since most of the estimated parameters have specific meanings in the given model. This is the basic motivation for proposing a new method for connectivity estimation based on an event sequence similarity measure.

### 3.2 Transfer entropy

Transfer entropy proposed by Schreiber expresses how much information is transferred from one time series to another [35]. It is widely used to estimate the connectivity of a network, especially in computational neuroscience [43]. In this paper, our proposed method is compared against transfer entropy.

Transfer entropy is defined on two countable sequences of stochastic variables. Let them be  $I = (i_1, i_2, \dots, i_N)$  and  $J = (j_1, j_2, \dots, j_M)$ , where  $i_n$

and  $j_n$  for any time point  $n$  are stochastic variables. Transfer entropy assumes that the given sequences are stationary Markov processes, that is, joint distribution  $p(i_{n+1}, i_n^{(k)}, j_n^{(h)})$  does not change with respect to  $n$ .  $i_n^{(k)}$  is a sequence of  $k$  variables, tracing back in time, that is,  $i_n, i_{n-1}, \dots, i_{n-k+1}$ . Also,  $j_n^{(h)}$  is a sequence of  $h$  variables tracing back in time, that is,  $j_n, j_{n-1}, \dots, j_{n-h+1}$ . Then transfer entropy  $T_{J \rightarrow I}$  from sequence  $J$  to  $I$  is defined as

$$T_{J \rightarrow I} = \sum p(i_{n+1}, i_n^{(k)}, j_n^{(h)}) \log \frac{p(i_{n+1} | i_n^{(k)}, j_n^{(h)})}{p(i_{n+1} | i_n^{(k)})}. \quad (1)$$

For a computational reason,  $h = 1$  is often used, since large  $h$  requires a large computation time.  $k$  is a parameter that must be set. Note that because of the stationarity assumption, a single sample from  $(I, J)$  can be used to estimate the distribution  $p(i_{n+1}, i_n^{(k)}, j_n^{(h)})$ . A procedure for computing transfer entropy for event sequences will be discussed later in Section 5.

### 3.3 Similarity measures between spike trains

A spike train is a sequence of time points when a neuron fired, that is, generated an action potential [32]. Our work introduces a similarity measure on pairs of spike trains in order to study similarities between different sequences and estimate the connectivity of the network. The objective is to evaluate the transmission of influences from one node to another through the edges and the network. We assume that spike trains are similar when the nodes are connected. Until now, similarity measures between spike trains have been mainly developed in the field of computational neuroscience [14].

Two of the most common approaches for defining spike train distances are edit-distance and embedding [15]. The Victor-Purpura distance (VP distance) [45, 44, 46] is a typical example of the former, whereas the van Rossum distance [42] is a typical example of the latter. The VP distance is defined as the minimum cost of transforming one spike train to another using a sequence of elementary editing processes, consisting of adding, removing and translating a spike. The van Rossum distance is defined by embedding spike trains to

a function space by convolving with a smoothing function.

There are various extensions regarding the VP distance and the van Rossum distance. Since they both have parameters that must be optimized using training data, Kreuz et al. proposed the ISI (inter-spike interval) metric and the SPIKE metric, which extends the van Rossum distance and the Victor-Purpura distance, respectively, and make them adaptive, that is, without requiring to optimize parameters explicitly [20]. Dubbs et al. pointed out that the Victor-Purpura distance has a property of the  $L_1$  norm that causes it to degenerate. As an alternative, they introduced  $L_p$  distances for spike trains [7]. Wu and Srivastava discussed spike train distances as metrics in a Riemannian space [47]. Chi et al. modeled spiking activity as a sequence of commonly appearing templates and intervals between them, where spike patterns within each template are invariant, while the interval length between successive templates varies [4]. This is another example of edit-distance, but in a larger time scale than the VP distance. Rusu and Florian proposed the modulus metric, which is a parameterless similarity measure defined by integrating the absolute difference of the infimum-based distances from each time point to the spike trains [33].

Stochastic modeling has been applied for defining similarity measures as well. Dauwels et al. introduced a stochastic event synchrony (SES), which evaluates the similarity between generative models of spike trains [5]. Naud et al. explored the statistical properties of similarity measures for small numbers of spike trains [23]. Johnson et al. proposed an information-theoretic distance between spike trains using the Kullback-Leibler distance between estimated rate functions [16].

Although these various methods model important aspects of similarity patterns among spike trains, the method proposed in this paper develops a flexible kernel-based approach that can be applied, thanks to its versatility, to various machine learning applications.

### 3.4 Kernel-based similarity measure

Although distances are more commonly used when measuring similarity in a network, another approach is to define a positive definite kernel, which

is similar to an evaluation of the similarity using an angular value [25, 26, 29]. Our approach belongs to this group. Shpigelman et al. introduced the Spikernel, based on binning spike trains and aligning them using a temporal warping function [37, 38]. Eichhorn et al. developed the alignment kernel based on an edit-distance [8]. In order to apply the method to multineuron spike trains, they used an empirical kernel map to create a feature vector consisting of the kernels values obtained from different neurons. Shpigelman et al. also proposed the kernel-ARMA, which transforms multivariate spike trains to finite dimensional vectors based on binning [39]. Fisher and Banerjee introduced the  $K_{REEF}$  kernel [9], which models neurons using the Spike Response Model. Paiva et al. proposed a general framework that uses a reproducing kernel Hilbert space for analyzing spike trains [25, 26]. Park et al. discussed the importance of using strictly positive definite kernels in order to make the feature map injective [28]. Li et al. extended spike train kernels to the multivariate case using the sum kernel [21]. In a previous work, a general way to define multivariate spike train kernels based on linear combinations has been introduced [40, 41].

## 4 Modeling approach

A principle behind our model is to use a spike train kernel to estimate connectivity between nodes. More specifically, the linear functional kernel (LF kernel) is applied, which is also called a memoryless cross intensity kernel (mCI kernel) [26, 29].

The LF kernel  $\kappa_{LF}$  is defined by convolving the spike train with a smoothing function and then taking an inner product in a function space. It is similar to the van Rossum distance in that it uses convolution with a smoothing function and maps the spike train to a function space.

Let  $i$  and  $j$  be nodes, and  $x^{(i)}$  and  $x^{(j)}$  be spike trains occurring at these nodes, respectively. These sequences are convolved with a smoothing function, thus mapping them to a function space. Gaussian or exponential functions are commonly used as smoothing functions. Let  $\eta_s$  be a smoothing function with parameter  $s$ . The smoothed function  $\zeta_{x^{(i)}}$  is defined as

$$\zeta_{x^{(i)}}(t) = \sum_{t_k^{(i)} \in x^{(i)}} \eta_s(t - t_k^{(i)}), \quad (2)$$

where  $t_k^{(i)}$  indicates the time of the  $k$ th spike in  $x^{(i)}$ .

Let  $T$  be the duration of observing spikes. The LF kernel uses the inner product of the smoothed functions, namely

$$\kappa_{LF}(x^{(i)}, x^{(j)}) = \int_0^T \zeta_{x^{(i)}}(t) \zeta_{x^{(j)}}(t) dt. \quad (3)$$

This is a positive definite kernel, since all inner products are positive definite. If  $\eta_s$  is symmetric with respect to  $t = 0$ , the resulting LF kernel symmetric.

In the following experiments, we used a Gaussian function for the smoothing function  $\eta_s$ . It has a time scale parameter  $s$ , which corresponds to the standard deviation parameter for a Gaussian function. It must be set based on physiological consideration, for example how much temporal fluctuation of a spike can be tolerated by postsynaptic neurons.

$$\eta_s(t) = \frac{1}{\sqrt{2\pi}s} \exp\left(-\frac{t^2}{2s^2}\right). \quad (4)$$

For computing the LF kernel, we used a Matlab library `spiketrainlib` implemented by Il Memming Park<sup>1</sup>. In order to reduce computation time, a windowed version of the LF kernel was used. This is denoted as “`wmci`” in `spiketrainlib`. The quadratic window was used.

Because the LF kernel is an inner product, it is positive definite, which means it fulfills the following condition of positive definiteness [22],

$$\forall D \in \mathbb{N}. \forall c \in \mathbb{C}^D. x \in \mathcal{X}^D. \sum_{i=1}^D \sum_{j=1}^D c_i \bar{c}_j \kappa(x^{(i)}, x^{(j)}) \geq 0, \quad (5)$$

where  $\mathbb{N}$  is natural numbers,  $\mathbb{C}$  is complex numbers, and  $\mathcal{X}$  is an arbitrary set. When we are considering a spike train kernel,  $\mathcal{X}$  corresponds to the set of all spike trains.

From a positive definite kernel  $\kappa$ , a normalized kernel  $\tilde{\kappa}$  is derived by

<sup>1</sup><https://code.google.com/p/spiketrainlib/>

$$\tilde{\kappa}(x, y) = \kappa(x, y) / \sqrt{\kappa(x, x)\kappa(y, y)} \quad (6)$$

The LF kernel is an inner product by definition, but for spike train kernels in general, one can derive that a positive definite kernel is an inner product in a reproducing kernel Hilbert space (RKHS) due to the Moore-Aronszajn theorem, and also of a space of Gaussian processes due to the Loeve theorem [1].

In other words, we have an induced reproducing kernel Hilbert space  $\mathbb{H}_\kappa$  such that  $\kappa(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathbb{H}_\kappa}$  where  $\phi : \mathcal{X} \rightarrow \mathbb{H}_\kappa$  is called a feature map. Spike trains  $x$  and  $y$  are mapped to feature vectors  $\phi(x)$  and  $\phi(y)$  in  $\mathbb{H}_\kappa$ . From the Cauchy-Schwarz inequality, a normalized kernel fulfills  $-1 \leq \tilde{\kappa} \leq 1$ , which makes a comparison among different data sets easier. Also, the value of a normalized kernel is easily interpretable. When  $\tilde{\kappa}(x, y) = 1$ , two nodes  $x$  and  $y$  are perfectly correlated, with respect to the similarity measure defined by kernel  $\kappa$ . When  $\tilde{\kappa}(x, y) = -1$ ,  $x$  and  $y$  are perfectly anti-correlated. When  $\tilde{\kappa}(x, y) = 0$ ,  $x$  and  $y$  are not correlated.

In fact, a normalized kernel acts like the cosine of an angle. This can be seen from the following equation in a Euclidean space, where  $\theta$  is an angle between two vectors  $a$  and  $b$ .

$$\cos \theta = \langle a, b \rangle / \|a\| \|b\| = \langle a, b \rangle / \sqrt{\langle a, a \rangle \langle b, b \rangle}. \quad (7)$$

The following equation indicates that  $\tilde{\kappa}(x, y)$  takes the same form as the right hand side of the equation above.

$$\tilde{\kappa}(x, y) = \langle \phi(x), \phi(y) \rangle_{\mathbb{H}_\kappa} / \sqrt{\langle \phi(x), \phi(x) \rangle_{\mathbb{H}_\kappa} \langle \phi(y), \phi(y) \rangle_{\mathbb{H}_\kappa}}. \quad (8)$$

This shows that  $\tilde{\kappa}(x, y)$  is an abstraction of “the cosine of an angle” between two vectors  $\phi(x)$  and  $\phi(y)$  in  $\mathbb{H}_\kappa$ . Therefore, a normalized kernel generalizes the cosine similarity measure commonly used in the vector space model for information retrieval [34]. Geometrically,  $\tilde{\kappa}(x, y) = 1$  corresponds to  $\phi(x)$  and  $\phi(y)$  pointing to the same direction.  $\tilde{\kappa}(x, y) = -1$  corresponds to  $\phi(x)$  and  $\phi(y)$  pointing to the opposite directions.  $\tilde{\kappa}(x, y) = 0$  corresponds to  $\phi(x)$  and  $\phi(y)$  being orthogonal. Note that this characterization is possible because of the positive definiteness of the original kernel  $\kappa$ .

We apply normalized kernels on spike trains in order to evaluate nodes similarity, and then to estimate connectivity between them. For each pair of nodes, the value of the normalized kernel is calculated using spike trains. Then the pairs with high values of the normalized kernel are considered as connected pairs, since the value of the kernel is assumed to represent the strength of connection. Here after, the value of the normalized kernel between a pair of nodes will be called the “score” of the pair. The top  $\lceil rN(N-1) \rceil$  pairs with the highest scores are estimated to be connected. Note that  $\lceil \alpha \rceil$  indicates the smallest integer above real number  $\alpha$ . This procedure requires the ratio of connectivity  $r$  to be known beforehand. In practice,  $r$  is estimated based on the observation of a small part of the target network. An alternative approach is to use classification method such as Fisher’s linear discriminant analysis or logistic regression to find the best threshold.

## 5 Evaluation

Let  $N$  be the number of nodes. There are  $N(N-1)$  possible directed connections. Such a network structure can be represented by a  $N \times N$  matrix  $J$ , where  $J_{ij}$  is the strength of connection for an edge from  $i$  to  $j$ . The goal of our method is to estimate the value of  $J_{ij}$ , especially whether it is zero or not, based solely on the observation of spike trains at nodes. We call  $J_{ij}$  the strength of connection from  $i$  to  $j$ .

### 5.1 Existing methods

We compare the kernel-based connectivity estimation method with transfer entropy based estimation, and also with a method using the Victor-Purpura distance (VP distance).

Transfer entropy was described earlier in the related work Section. It is defined on a pair of countable sequences of stochastic variables. Since a pair of event sequences is a sample taken from a marked point process, it should first be mapped to a sample generated from two countable sequences of stochastic variables. This is done by binning [6]. In other words, a set of windows covering the whole time range of the event sequences are prepared, and the number of events within each window is counted.

The result gives a pair of sequences consisting of the number of events in each bin. When describing parameter settings, the bin size is represented by  $e$ , the number of levels for grouping event numbers in a bin is represented by  $\ell$ , and the Markov chain length for sequence  $I$  is represented by  $k$ .

Another problem that needs to be solved is that when the maximum number of events in one bin is large, it becomes difficult to estimate the joint distribution of the Markov chain accurately. This is due to the sampling bias problem [27]. This can be summarized as follows. For any time point  $n$ , let variables  $i_n$  and  $j_n$  can take  $a$  and  $b$  types of values, respectively. Then samples generated from the joint distribution  $p(i_{n+1}, i_n^{(k)}, j_n^{(h)})$  can take  $a^{k+1}b^{h+1}$  different values. When  $a$  and  $b$  are large, the number of possible values for the sample would be huge and there will not be enough realizations for all of them in the observed sequences, thus making it extremely difficult to estimate the distribution correctly. To avoid this problem, we limit the number of values that  $i_n$  and  $j_n$  can take by grouping the number of events into a few number of levels. For example, if the maximum number of events in a bin is 7 and the number of levels is set to 4, then the groups will be  $\{0, 1\}$ ,  $\{2, 3\}$ ,  $\{4, 5\}$  and  $\{6, 7\}$ . These groups are indexed by 1, 2, 3, and 4, respectively. These levels (groups) are used as values that  $i_n$  and  $j_n$  can take. The sequences of the levels are then considered as a sample taken from  $(I, J)$ , and the joint distribution  $p(i_{n+1}, i_n^{(k)}, j_n^{(h)})$  is estimated using these sequences.

The VP distance is defined as the minimum total cost for transforming a spike train  $x^{(i)}$  to another spike train  $x^{(j)}$  using a sequence of elementary operations [45]. The elementary operations are [1] remove a spike, [2] add a spike, and [3] move a spike along the time axis. The costs of removing or adding a spike are both 1, whereas moving a spike  $\Delta t$  in time costs  $q\Delta t$ . The VP distance therefore has one parameter  $q$ , which has the dimension of inverse-time. The value of  $q$  must be optimized using observed data. To calculate the VP distance, we used an implementation by Thomas Kreuz<sup>2</sup>. Similar to the kernel-based methods, the  $\lceil rN(N-1) \rceil$  pairs of nodes having the lowest distances are estimated to be connected.

## 5.2 Synthetic data

Synthetic data used in this experiment are generated from the CERM (coupled escape-rate model) framework developed by Ryota Kobayashi and Katsunori Kitano [19]. CERM simulates a network with spike trains, where the occurrences of spikes at each node influence the occurrence probability at other nodes. The model has been shown to simulate the activity of a biological neural network using only a few parameters. This is an important assumption since one goal of our method is to estimate the connectivity of a real neural network based on observed spikes.

In CERM, spikes at each node are generated using the escape rate model, which is commonly used to produce spike trains [12]. At each node  $i$ , spikes occur stochastically following the inhomogeneous Poisson process with intensity  $\lambda^{(i)}$ , which is determined by past spike times and influences from other nodes, namely

$$\lambda^{(i)}(t) = \exp \left( u^{(i)}(t) + \alpha^{(i)}\xi^{(i)}(t) + \sum_j J_{ij}\zeta^{(i,j)}(t) \right). \quad (9)$$

$\lambda^{(i)}$  is also called the escape rate or hazard of a stochastic process [12].  $u^{(i)}$  is the time-varying external input to node  $i$ .  $\xi^{(i)}$  is the after-effect at node  $i$  after each spike, and  $\zeta^{(i,j)}$  is the strength of influence sent out from node  $i$  to node  $j$ .

Setting  $\alpha^{(i)} > 0$  makes another spike likely to occur just after one spike. This denotes an entraining effect. For  $\alpha^{(i)} < 0$ , once a spike occur, another spike is unlikely to occur for a certain amount of time. This is called a refractory period. The case where  $J_{ij} > 0$  is called an excitatory connection, and the case where  $J_{ij} < 0$  is called an inhibitory connection. The case where  $J_{ij} = 0$  corresponds to the case that there is no connection between nodes  $i$  and  $j$ . The values of  $J_{ij}$  are determined stochastically at the beginning of each simulation. This is done by the generation of a random graph. Among possible  $N(N-1)$  connections, ratio  $r$  are realized as edges.  $J_{ij}$  is sampled from a uniform distribution on  $[J_{\min}, J_{\max}]$ . Afterwards, all of the diagonal entries and  $1-r$  of the off-diagonal entries of  $J$  are set to zero.

<sup>2</sup><http://www-users.med.cornell.edu/>

[~jdvicto/spkd\\_qpara.html](http://jdvicto/spkd_qpara.html)

$\xi^{(i)}$  and  $\zeta^{(i,j)}$  evolve over time by the following differential equations,

$$\frac{d\xi^{(i)}}{dt} = -\frac{\xi^{(i)}}{\tau_\xi} + \sum_k \delta(t - t_k^{(i)}), \quad (10)$$

$$\frac{d\zeta^{(i,j)}}{dt} = -\frac{\zeta^{(i,j)}}{\tau_\zeta} + \sum_k \delta(t - t_k^{(j)}), \quad (11)$$

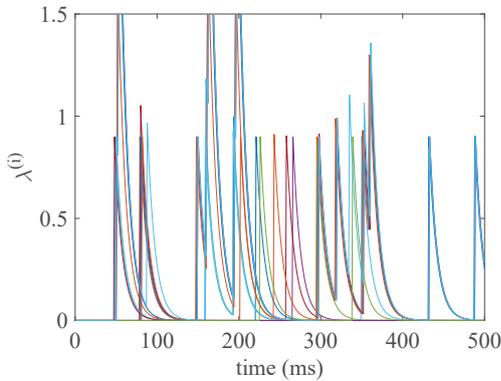
where  $\tau_\xi$  and  $\tau_\zeta$  are time scale parameters,  $\delta$  is the Dirac delta function, and  $t_k^{(i)}$  is the time of the  $k$ th spike at node  $i$ . Summation  $\sum_k$  is over all of spikes occurring at node  $i$  or  $j$ . Using the Laplace transform, we get the following solutions.

$$\xi^{(i)}(t) = \sum_k \exp\left(-\frac{t - t_k^{(i)}}{\tau_\xi}\right) \Theta(t - t_k^{(i)}), \quad (12)$$

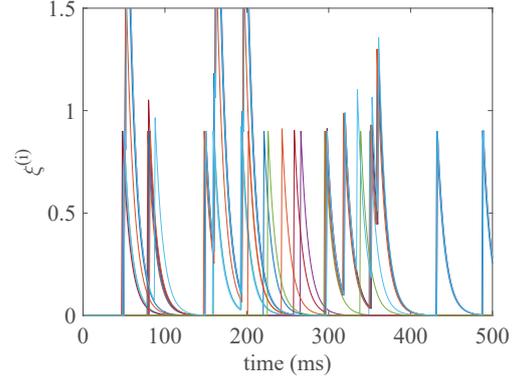
$$\zeta^{(i,j)}(t) = \sum_k \exp\left(-\frac{t - t_k^{(j)}}{\tau_\zeta}\right) \Theta(t - t_k^{(j)}), \quad (13)$$

where  $\Theta$  is the Heaviside step function.

Figures 2 and 3 illustrate how  $\lambda^{(i)}$  and  $\xi^{(i)}$ , respectively, evolve over time. In the figures, each node is indicated by a different color. Time points where  $\lambda^{(i)}$  increases rapidly and crosses a set threshold are considered as spikes (events).



**Figure 2.** Time evolution of intensity  $\lambda^{(i)}$ . Different neurons are represented by different colors. The abscissa is time in milliseconds. The ordinate is the value of  $\lambda$  for each neuron  $i$ .



**Figure 3.** Time evolution of after-effect  $\xi^{(i)}$ . Different neurons are represented by different colors. The abscissa is time in milliseconds. The ordinate is the value of  $\xi$  for each neuron  $i$ .

Figure 4 illustrates an example of spike trains generated by CERM. The abscissa denotes the time while the ordinate shows different nodes. The parameters are set to  $N = 20$ ,  $r = 0.05$ ,  $T = 5$ ,  $\tau_\xi = 0.01$ ,  $\tau_\zeta = 0.01$ ,  $u^{(i)} = 1$ ,  $\alpha^{(i)} = -10$ ,  $J_{\min} = 10$ , and  $J_{\max} = 15$ . It appears that it is not straightforward to visually determine which nodes are connected just by considering these spike trains. The aim of our method is actually to estimate this based solely on the analysis of spike trains.

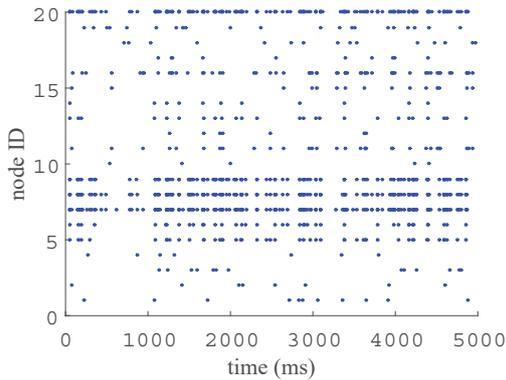
Figure 5 shows the result of calculating the score  $\tilde{\kappa}_{LF}$  for each pair of the nodes using the generated spike trains. The time scale parameter for the LF kernel was set  $s = 0.005$ . Each blue circle is a pair of nodes. The orange line is a threshold obtained by Fisher's linear discriminant analysis. For each pair  $(i, j)$  represented by a blue circle, the abscissa is the averaged connectivity  $(J_{ij} + J_{ji})/2$ , which represents the undirected strength of connection. Note that because of this averaging process, each blue circle represents two directed pairs  $(i, j)$  and  $(j, i)$ . The ordinate is the score  $\tilde{\kappa}_{LF}(x^{(i)}, x^{(j)})$ . The plot shows that when a pair of nodes is connected (i.e. the strength of connection is not zero), the score is generally larger than that of the non-connected pairs. Figure 6 shows the distribution of the scores for non-connected and connected node pairs. It shows that two classes (non-connected and connected) have two distinct distributions.

For one dimensional data, the threshold given by Fisher's linear discriminant analysis is the midpoint of the means of two classes. For this data, it was 0.3309. Table 1 shows the result of estimat-

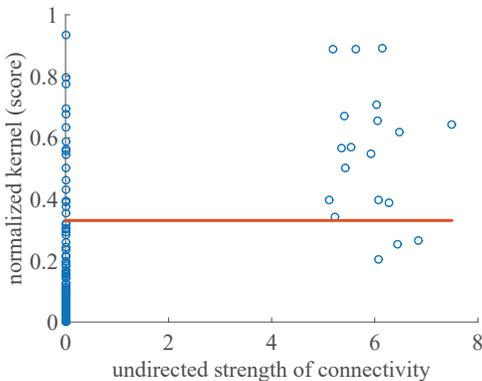
**Table 1.** Classification of non-connected and connected pairs of nodes.

	total	to non-connected	to connected	precision
non-connected	342	308	34	0.90
connected	38	6	32	0.89

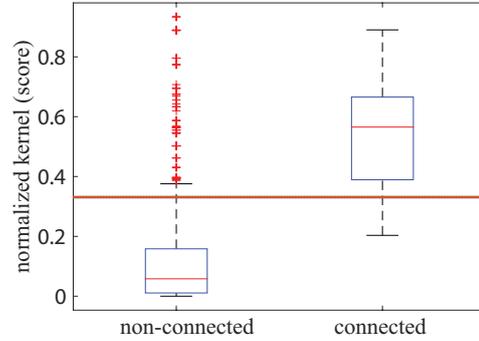
ing two classes using this threshold. Out of 342 non-connected pairs, 308 were correctly classified as non-connected. On the other hand, out of 38 connected pairs, 32 were correctly classified. This makes around 90 percent classification accuracy.



**Figure 4.** Spike trains generated by CERM. Each blue dot represents a spike. The abscissa is time in milliseconds. The ordinate is node ID.



**Figure 5.** The undirected strength of connection and scores by normalized kernel. Each marker (circle) represents a pair of neurons. The abscissa is the undirected strength of connection. The ordinate is the value of normalized kernel (score).



**Figure 6.** Distributions of scores for non-connected and connected node pairs. The horizontal line at the center of each box shows the mean of the estimations for samples corresponding to each direction. The top and bottom edges of a box represent the 25th and 75th percentiles, and the whiskers extend to the furthest data points not considered outliers.

In order to quantitatively evaluate the proposed method, we conducted 4 types of experiments. Common parameters are  $T = 5$ ,  $\tau_\xi = 0.01$ ,  $\tau_\zeta = 0.01$ ,  $u^{(i)} = 0.5$ ,  $\alpha^{(i)} = -10$ ,  $\ell = 4$ , and  $k = 2$ . The evaluation is conducted by a measure of precision, i.e., the number of pairs that are actually connected divided by the number of pairs that the method estimated to be connected.

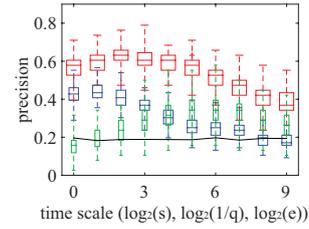
We only show precision and not recall, since in this experiment they are equal. Let  $C$  be the number of correctly retrieved connections,  $E$  be the number of estimated connections, and  $A$  be the number of actual connections. Then recall is  $C/E$  and precision is  $C/A$ . In this experiment,  $E = A$  since it is assumed that the ratio of connection is known. Therefore the recall is equal to the precision.

In all of the box plot figures, the large red boxes indicate the precision of the proposed kernel-based method, the middle-sized blue boxes indicate that of the transfer entropy based method, and the small green boxes indicate that of the VP distance. The black line indicates the average of precision obtained by random selection of pairs, which ran-

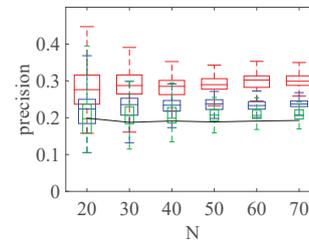
domly selects  $rN(N-1)/2$  edges from all of the  $N(N-1)/2$  possible undirected edges. In other words, the black line indicates the chance level precision. For each type of experiment, 100 trials were carried out. Each trial is a 5-second long spike train. In box plots, the horizontal line at the center of each box shows the mean of the estimations for samples corresponding to each direction. The top and bottom edges of a box represent the 25th and 75th percentiles, and the whiskers extend to the furthest data points not considered outliers.

**Experiment 1:** Time scale parameter  $s$  of the LF kernel was changed. The parameters were set as  $N = 20$ ,  $r = 0.1$ ,  $J_{\min} = 3$ ,  $J_{\max} = 5$ . The size of the bin  $e$  for computing transfer entropy was changed from 1 ms to 512 ms by scale of 2. The inverse time constant  $q$  for the VP distance is changed from  $2^{-9}$  to 1 by scale of 2. Figure 7 illustrates the results. The unit for  $s$  is in milliseconds. The precision for the LF kernel is maximized when the time scale parameter is set to around 4 milliseconds, that is, when  $\log_2(s) = 2$ . This indicates the importance of setting of the time scale parameter (which can be considered as a temporal constant) in order to achieve high precision. The result for transfer entropy (blue) is maximized at around 2 ms. The result for the VP distance (green) is aligned that the inverse time scale parameter  $q$  is mapped to the time scale parameter by  $s = 1/q$ . The optimal inverse-time constant for the VP distance was  $q = 0.0156$ , that is, when  $\log_2(1/q) = 6$ . The difference between the optimal time constants for the kernel method and the VP distance comes from different effects that time constants give to the objective functions.

**Experiment 2:** The number of nodes in the network was changed. The parameters were set as  $r = 0.1$ ,  $s = 5$ ,  $q = 0.0156$ ,  $J_{\min} = 1$ ,  $J_{\max} = 2$ , and  $e = 10$  ms. The values of  $J_{\min} = 1$  and  $J_{\max} = 2$  are set differently from the previous experiment, in order to avoid excessive firing in all neurons. Otherwise, estimating connectivity becomes impossible. Figure 8 illustrates the scalability of our method when increasing the number of nodes within this range. One can remark that as the number of nodes increases, so does the number of possible connections, therefore precision for the random selection algorithm decreases.



**Figure 7.** Precisions as time scale parameter  $s$  is changed. The large red boxes indicate precision of the proposed kernel-based method. The medium-sized blue boxes indicate precision for transfer entropy. The small-sized green boxes indicate precision for the VP distance. The black line indicates the averaged precision of the random selection algorithm.

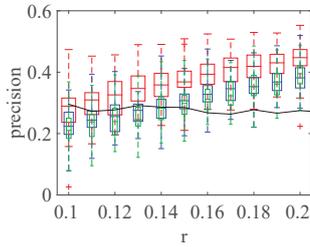


**Figure 8.** Precisions as the number of nodes  $N$  is changed. The large red boxes indicate precision of the proposed kernel-based method. The medium-sized blue boxes indicate precision for transfer entropy. The small-sized green boxes indicate precision for the VP distance. The black line indicates the averaged precision of the random selection algorithm.

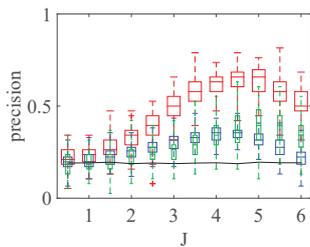
**Experiment 3:** The ratio of the connectivity of the network was changed. The parameters were set as  $N = 20$ ,  $s = 5$ ,  $q = 0.0156$ ,  $J_{\min} = 1$ ,  $J_{\max} = 2$ , and  $e = 10$  ms. Figure 9 indicates the result.

**Experiment 4:** The strength of connection  $J_{ij}$  between connected pairs were changed from 1 to 6. The parameters were set as  $N = 20$ ,  $r = 0.1$ ,  $s = 5$ ,  $q = 0.0156$ , and  $e = 10$  ms. Figure 10 shows that our method performs better than the VP distance and the random selection algorithm. It also shows that precision is sensitive to the value of  $J_{ij}$ . For low  $J_{ij}$ , connectivity is too weak that the influence of a spike does not transfer from one node to another. On the other hand, for high  $J_{ij}$ , where the connectivity is strong, one spike often results in a global occurrence of spikes throughout the network, there-

fore making it more difficult to judge whether two nodes are connected or not based solely on the similarity of spikes occurrence.



**Figure 9.** Precisions as the ratio of connectivity  $r$  is changed. The large red boxes indicate precision of the proposed kernel-based method. The medium-sized blue boxes indicate precision for transfer entropy. The small-sized green boxes indicate precision for the VP distance. The black line indicates the averaged precision of the random selection algorithm.



**Figure 10.** Precisions as the strength of connection  $J_{ij}$  is changed. The large red boxes indicate precision of the proposed kernel-based method. The medium-sized blue boxes indicate precision for transfer entropy. The small-sized green boxes indicate precision for the VP distance. The black line indicates the averaged precision of the random selection algorithm.

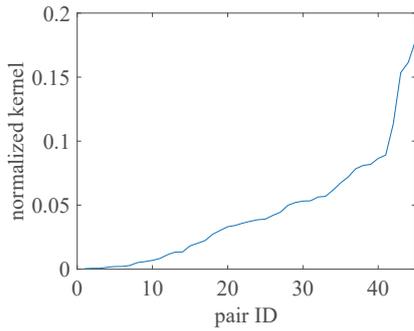
Overall the results show that our kernel-based method (red) performs better than those based on transfer entropy (blue) and the VP distance (green), and also much better than the chance level (black). Although the absolute level of precision is still not high, the method can be used to extract possible connection candidates. One can also combine our method with other connectivity estimation methods in order to find more reliable results.

### 5.3 Real data

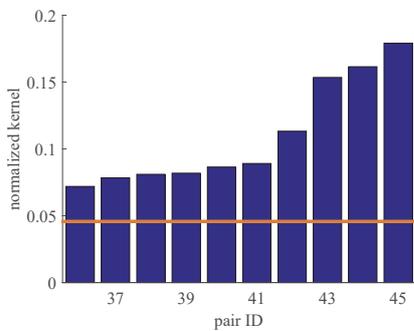
The method has been applied to an estimation of the functional connectivity among neurons in a cortical circuit. Such a task is important when analyzing the mechanism of cortical information processing and also developing brain machine interfaces (BMI). The experiment has been applied to real data available at the CRCNS (Collaborative Research in Computational Neuroscience)<sup>3</sup>. Specifically, we used PVC-3 data, which is a data set recorded from area 17 (visual cortex) of an anaesthetized cat, in response to presenting oriented drifting bars [3]. The data is recorded using polytrodes [2]. By applying spike sorting to raw data, spike trains (sequences of times that neurons produced action potentials) from 10 units were identified. Each unit is assumed to correspond to a neuron. As a part of a network, each unit is a node.

For these 10 nodes, there are 45 possible undirected connections. We estimated the strength of each of these possible connections (node pairs) using the proposed method. Time scale parameter was set to 3 milliseconds, based on a physiological consideration. The total recording time is 722 seconds, but out of it, 20 seconds were used for the analysis. Figure 11 illustrates the score  $\tilde{\kappa}$  sorted in increasing order. Figure 12 shows the scores for the top 10 pairs using the blue bar graph. It also shows the average score for all of the pairs by the orange line. The graphs show that the scores vary significantly, indicating that some pairs are connected while others are not actually connected. In order to determine the threshold for judging whether a pair is connected or not, one must conduct a biological experiment and obtain the threshold for the score. One possible way to do this is to use a model such as CERM using observed parameters, and determine the threshold by comparing the real data with the synthetic data. Also, if there is a biological neural network where the connectivity is known, the threshold can be optimized in a supervised learning manner.

<sup>3</sup><https://crcns.org/>

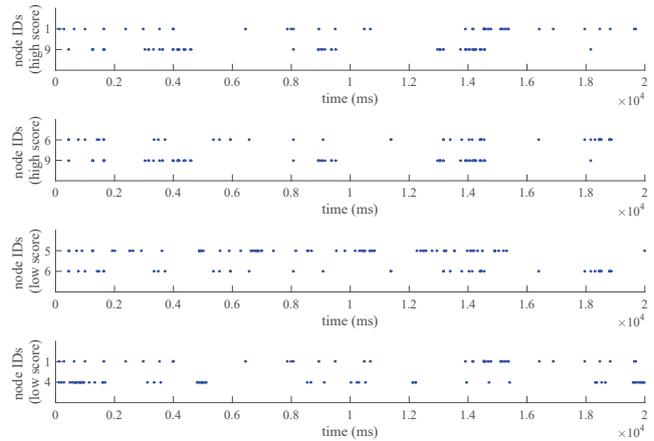


**Figure 11.** Normalized kernel values for all pair sorted in the increasing order. The abscissa is pair ID ordered by increasing value of normalized kernel. The ordinate is the value of normalized kernel.



**Figure 12.** Top 10 pairs with high normalized kernel values. The abscissa is pair ID from 36 to 45. The blue bars represent the values of normalized kernel. The orange line represents the average score for all of the pairs.

Figure 13 illustrates spike trains for pairs of nodes that had high scores and low scores. The top two diagrams show the pairs with high scores, while the two bottom diagrams show the pairs with low scores. It indicates that when the score is high, spikes are synchronized more often. This can be explained by a model where when two neurons are connected, they tend to fire in a sequence, since firing of a presynaptic neuron induces firing of a post-synaptic neuron.



**Figure 13.** Spike trains from connected and disconnected pairs of nodes according to the proposed estimation method. Panels represent different selections of pairs. Upper two panels are for pairs with high scores, and lower two panels are pairs with low scores. Each blue dot represents a spike. The abscissa is time in milliseconds. The ordinate is node ID.

## 6 Conclusion

The method developed in this paper showed that a normalized positive definite kernel on event sequences (spike trains) provides a novel way to estimate the connectivity of a network. The proposed method relies only on the times of events occurring at each node. The results of the experiments showed that the kernel-based connectivity estimation method is more effective than methods based on transfer entropy and the Victor-Purpura distance (VP distance). For the wide range of parameter values, our method was shown to be effective. The cases where the method did not perform very well is due to too many or too few events occurring at each node, such that all event sequences turned out to be similar, irrespective of connectivity among them.

In future work, we plan to extend the method such that the directions of the connections can be estimated as well. This is important for applying the method further to the analysis of biological neural networks, for example. We would also like to combine our method with other connectivity estimation methods to improve its precision. Moreover, the kernel used is mainly oriented towards the coin-

cidences of event times in a short time scale, determined by time scale parameter  $s$ . The method therefore does not capture the global structure of how events are temporally distributed at large scale. Using a spike train kernel that does not introduce smoothing, for example Spikernel [37, 38], is one way to introduce similarity measure based on global features. For example, kernel discriminant analysis or kernel logistic regression could be used for achieving more accurate classification between non-connected and connected pairs [31]. Finally, the current model only considers pair-wise connectivity. Introducing a multivariate spike train kernel [29, 21, 40] would extend the analysis to the case where spike trains are governed by population activity.

## Acknowledgment

This work was supported in part by JSPS KAKENHI Grant Numbers 21700121, 25280110, and 25540159.

## References

- [1] Alain Berlinet and Christine Thomas-Agnan, Reproducing Kernel Hilbert Spaces in Probability and Statistics, Kluwer Academic Publishers, 2001.
- [2] Timothy J. Blanche, Martin A. Spacek, Jamille F. Hetke, and Nicholas V. Swindale, Polytrodes: high density silicon electrode arrays for large scale multiunit recording, *Journal of Neurophysiology*, Vol.93, No.5, pp.2987-3000, 2005.
- [3] Tim Blanche, Multi-neuron recordings in primary visual cortex. CRCNS.org. 2009. <http://dx.doi.org/10.6080/KOMW2F2J>
- [4] Zhiyi Chi, Wei Wu, Zach Haga, Nicholas G. Hatsopoulos, and Daniel Margoliash, Template-based spike pattern identification with linear convolution and dynamic time warping, *Journal of Neurophysiology*, Vol.97, pp.1221-1235, 2007.
- [5] Justin Dauwels, François Vialatte, Theophane Weber, and Andrzej Cichocki, On similarity measures for spike trains, in *Proceeding of the 15th International Conference on Advances in Neuro-Information Processing*, pp.177-185, 2009.
- [6] Peter Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, MIT Press, 2001.
- [7] Alexander J. Dubbs, Brad A. Seiler, and Marcelo O. Magnasco, A fast  $L_p$  spike alignment metric, *Neural Computation*, Vol.22, pp.2785-2808, 2010.
- [8] Jan Eichhorn, Andreas Tolias, Alexander Zien, Malte Kuss, Carl Edward Rasmussen, Jason Weston, Nikos Logothetis, and Bernhard Schölkopf, Prediction on spike data using kernel algorithms, *Advances in Neural Information Processing Systems*, Vol.16, pp.1367-1374, 2004.
- [9] Nicholas Fisher and Arunava Banerjee, A novel kernel for learning a neuron model from spike train data, *Advances in Neural Information Processing Systems*, Vol.23, pp.595-603, 2010.
- [10] K.J. Friston, L. Harrison, and W. Penny, Dynamic causal modeling, *NeuroImage*, Vol.19, no. 4, pg.1273-1302, 2003.
- [11] Matteo Garofalo, Thierry Nieuws, Paolo Massobrio, and Sergio Martinoia, Evaluation of the performance of information theory-based methods and cross-correlation to estimate the functional connectivity in cortical networks, *PLoS One*, Vol.4, No.8, e6482, 2009.
- [12] Wulfram Gerstner, Werner M. Kistler, Richard Naud, and Liam Paninski, *Neuronal Dynamics*, Cambridge University Press, 2014.
- [13] C.W.J. Granger, Investigating causal relations by econometric models and cross-spectral methods, *Econometrica*, Vol.37, No.3, 424-438, 1969.
- [14] Conor Houghton and Thomas Kreuz, Measures of spike train synchrony: From single neurons to populations, in Misha Meyer Pesenson (Ed.), *Multiscale Analysis and Nonlinear Dynamics: From Genes to the Brain*, John Wiley & Sons, Inc., 2013.
- [15] Conor Houghton and Jonathan Victor, Measuring representational distances - the spike-train metrics approach, in Nikolaus Kriegeskorte and Gabriel Kreiman (Eds.), *Understanding Visual Population Codes: Towards a Common Multivariate Framework for Cell Recording and Functional Imaging*, MIT Press, 2011.
- [16] Don H. Johnson, Charlotte M. Gruner, Keith Baggerly, and Chandran Seshagiri, Information-theoretic analysis of neural coding, *Journal of Computational Neuroscience*, Vol.10, pp.47-69, 2001.
- [17] Maciej Kaminski and Katarzyna J. Blinowska, A new method of the description of the information flow in the brain structures, *Biological Cybernetics*, Vol. 65, No.3, pp.203-210, 1991.
- [18] Maciej Kaminski, Mingzhou Ding, Wilson A. Truccolo, and Steven L. Bressler, Evaluating causal relations in neural systems: Granger causality, directed transfer function and statistical assessment of

- significance, *Biological Cybernetics*, Vol. 85, No.2, pp.145-157, 2001.
- [19] Ryota Kobayashi and Katsunori Kitano, Impact of network topology on inference of synaptic connectivity from multi-neuronal spike data simulated by a large-scale cortical network model, *Journal of Computational Neuroscience*, Vol.35, pp.109-124, 2013.
- [20] Thomas Kreuz, Daniel Chicharro, Conor Houghton, Ralph G. Andrzejak, and Florian Mormann, Monitoring spike train synchrony, *Journal of Neurophysiology*, Vol.109, pp.1457-1472, 2012.
- [21] Lin Li, Austin J. Brockmeier, John S. Choi, Joseph T. Francis, Justin C. Sanchez, and Jose C. Principe, A tensor-product-kernel framework for multiscale neural activity decoding and control, *Computational Intelligence and Neuroscience*, 2014.
- [22] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar, *Foundations of Machine Learning*, MIT Press, 2012.
- [23] Richard Naud, Felipe Gerhard, Skander Mensi, and Wulfram Gerstner, Improved similarity measures for small sets of spike trains, *Neural Computation*, Vol.23, pp.3016-3069, 2011.
- [24] Murat Okatan, Mathew A. Wilson, and Emery N. Brown, Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity, *Neural Computation*, Vol.17, pp.1927-1961, 2005.
- [25] Antonio R.C. Paiva, Il Park, and Jose C. Principe, A reproducing kernel Hilbert space framework for spike train signal processing, *Neural Computation*, Vol.21, No.2, pp.424-449, 2009.
- [26] Antonio R.C. Paiva, Il Park, and Jose C. Principe, Inner products for representation and learning in the spike train domain, in Karim G. Oweiss (Ed.), *Statistical Signal Processing for Neuroscience and Neurotechnology*, Academic Press, 2010.
- [27] Stefano Panzeri and Alessandro Treves, Analytical estimates of limited sampling in different information measures, *Network: Computation in Neural Systems*, 7, pp.87-107, 1996.
- [28] Il Memming Park, Sohan Seth, Murali Rao, and Jose C. Principe, Strictly positive definite spike train kernels for point process divergences, *Neural Computation*, Vol.24, pp.2223-2250, 2012.
- [29] Il Memming Park, Sohan Seth, Antonio R.C. Paiva, Lin Li, and Jose C. Principe, Kernel methods on spike train space for neuroscience: a tutorial, *Signal Processing Magazine*, Vol.30, No.4, pp.149-160, 2013.
- [30] Christopher J. Quinn, Todd P. Coleman, Negar Kiyavash, and Nicholas G. Hatsopoulos, Estimating the directed information to infer causal relationships in ensemble neural spike train recordings, *Journal of Computational Neuroscience*, 2010.
- [31] Carl Edward Rasmussen and Christopher K. I. Williams, *Gaussian Processes for Machine Learning*, the MIT Press, 2006.
- [32] Fred Rieke, David Warland, Rob de Ruyter van Steveninck, and William Bialek, *Spikes: Exploring the Neural Code*, MIT Press, 1997.
- [33] Catalin V. Rusu and Razvan V. Florian, A new class of metrics for spike trains, *Neural Computation*, Vol.26, No.2, pp.306-348, 2014.
- [34] Gerard M. Salton, Andrew Wong, and Chungshu Yang, A vector space model for automatic indexing, *Communications of the ACM*, Vol.18, No.11, pp.613-620, 1975.
- [35] Thomas Schreiber, Measuring information transfer, *Physical Review Letters*, Vol.85, No.2, 2000.
- [36] John Shawe-Taylor and Nello Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.
- [37] Lavi Shpigelman, Yoram Singer, Rony Paz, and Eilon Vaadia, Spikernels: embedding spiking neurons in inner product spaces, *Advances in Neural Information Processing Systems*, Vol.15, pp.125-132, 2003.
- [38] Lavi Shpigelman, Yoram Singer, Rony Paz, and Eilon Vaadia, Spikernels: predicting arm movements by embedding population spike rate patterns in inner-product spaces, *Neural Computation*, Vol.17, pp.671-690, 2005.
- [39] Lavi Shpigelman, Hagai Lalazar, and Eilon Vaadia, Kernel-ARMA for hand tracking and brain-machine interfacing during 3D motor control, *Advances in neural information processing systems*, Vol.21, 1489-1496, 2008.
- [40] Taro Tezuka, Spike train kernels for multiple neuron recordings, *Proceedings of the 39th International Conference on Acoustics, Speech and Signal Processing*, pp.6035-6039, 2014.
- [41] Taro Tezuka and Christophe Claramunt, Connectivity estimation of neural networks using a spike train kernel, *Proceedings of the 2015 International Joint Conference on Neural Networks*, pp.1-7, Killarney, Ireland, July 12-17, 2015.
- [42] M.C.W. van Rossum, A novel spike distance, *Neural Computation*, Vol.13, pp.751-763, 2001.

- [43] Raul Vicente, Michael Wibral, Michael Lindner, and Gordon Pipa, Transfer entropy - a model-free measure of effective connectivity for the neurosciences. *Journal of Computational Neuroscience*, Vol.30, No.1, pp.45-67, 2011.
- [44] Jonathan D. Victor, Spike train metrics, *Current Opinion in Neurobiology*, Vol.15, pp.585-592, 2005.
- [45] Jonathan D. Victor and Keith P. Purpura, Nature and precision of temporal coding in visual cortex: a metric-space analysis, *Journal of Neurophysiology*, Vol.76, pp.1310-1326, 1996.
- [46] Jonathan D. Victor and Keith P. Purpura, Spike metrics, in Nikolaus Kriegeskorte and Gabriel Kreiman (Eds.), *Understanding Visual Population Codes: Towards a Common Multivariate Framework for Cell Recording and Functional Imaging*, MIT Press, 2011.
- [47] Wei Wu and Anuj Srivastava, An information-geometric framework for statistical inferences in the neural spike train space, *Journal of Computational Neuroscience*, Vol.31, No.3, pp.725-48, 2011.



**Taro Tezuka** is an associate professor at the Faculty of Library, Information and Media Science in the University of Tsukuba. He got his PhD in informatics from Kyoto University in 2005. He worked in Ritsumeikan University prior to joining the University of Tsukuba in 2011. He was a visiting researcher at the Naval Academy Research Institute in France from 2014 to 2015. He is currently working on sparse coding and dictionary learning for image processing.



Dr **Christophe Claramunt** is a Professor in Computer Science and chair of the Naval Academy Research Institute in North West France. He received a PhD in computer science from the University of Burgundy and a “Habilitation a Diriger des Recherches” from the University of Rouen, in 1998 and 2001, respectively. Before joining the Naval Academy Research Institute, he has been a senior lecturer in computing at the Nottingham Trent University and a senior researcher at the Swiss Federal Institute of Technology in Lausanne. His research work concerns theoretical and pluridisciplinary aspects of geographical information science and their application to urban, natural and maritime environments. Over the past few years he has been involved in several EU and industrial-funded projects. He has widely published in the domain of GIS, currently serves in the editorial boards of several leading international GIS journals, and is regularly involved in the chairing and program committees of several inter.