Analyses of time point sequences using kernel methods

Master's Thesis in Graduate School of Library,

Information and Media Studies

Analyses of time point sequences using kernel methods

March 2015

201321664

HU WEI

Analyses of time point sequences using kernel methods

HU WEI

Graduate School of Library, Information and Media Studies
University of Tsukuba

March 2015

CONTENTS

CHAPTER 1 – INTRODUCTION

1.1 **Background**

A time sequences is a sequence of data points, consisting of consecutive measurements taken at a time interval. The analysis of time sequences is a valuable way to extract meaningful information and characteristics of the data and predict future values in forecasting. Therefore, analysis of time sequences is very important as a fundamental problem, such as neural activity analysis.

One important example of time sequences that have become important in recent years is the spike train, which is generated by biological neurons. In our brain, there are over 100 billion neurons, and they communicate with each other using pulses by generating characteristic electrical signals called action potentials or spikes. A spike train is a sequence of spikes or action potentials fired by a neuron as time sequences. Although there are many research show that the information is coded in the spike sequences in the field of neural coding, it can't explain how spike works and the temporal structure of spike trains. We have grade interest in how these neurons transmit information rapidly and what relationships they have between each node (neural) in a cortical network. There is a big motivation to understand how to analyze and decode the information expressed by spike trains in the field of brain machine interfaces (BMI), which is to make it possible for the communication between the brain and machines.

In order to analyze spike trains, we propose the use of kernel methods, which can be for various tasks in machine learning, including regression, clustering and dimension reduction. Estimating the population activity patterns between two or more spike trains is a fundamental problem and the use of kernel methods provide an opportunity for building a framework deal with these problems by using spike train metric. In this paper, we use the memoryless cross intensity kernel (mCI kernel) to analyze spike trains, which introduced in chapter 4.

1.2 **General Motivation**

We not only want to know information contents of neural signal systems, but also want to find the structure of neural networks. It is therefore important to estimate the strength of connectivity between nodes in such a network. Such a method can also be used for other networks where events occur at each node, dependent on events occurring at other nodes. Some of the examples include events on a social network, spreading of an epidemic, and physiological pathways.

In order to approach this problem, this paper focuses on the spike timing information and then use the spike sequences alone to estimate the synaptic weights of the network. Specifically, we suggest kernel methods for multichannel spike trains that can provide an opportunity to measure spike trains. First we use the coupled escape rate model (CERM) to simulate multichannel spike trains data and propose a distance to prove the effectivity of the result. Then we extend the CERM to a network and predict the strength of connectivity of the network. Specifically, we apply the kernel ridge regression, where a positive definite kernel is calculated by convolving the spike trains and a smoothing function. We then discuss how the result is affected by the number of neurons and the connectivity of the network.

The proposed method can be used for analyzing networks other than biological neural networks. Possible applications include predicting purchasing behaviors of consumers connected by a social network or modeling the spread of an epidemic.

1.3 **Outline**

The remainder of this paper is organized as follows. In Chapter 2, we will describe related work about spike train kernels and the synaptic connectivity. In Chapter 3 and Chapter 4 we will introduce the simulation model for multichannel spike train, describe the memoryless cross intensity kernel for multichannel spike train model and how to use the kernel ridge regression to estimate the strength of connectivity. And then, various experiments were used to evaluate the result in Chapter 5. Finally, we will give the conclusion in Chapter 6.

CHAPTER 2 – RELATED WORKS

2.1 Connectivity estimation

Much work has been done in estimating the synaptic connectivity in cortical networks because of the lack of tools to measure the connectivity directly. Granger causality is one of the most popular used methods to estimate the connectivity in a network [1]. Quinn et al. proposed the directed information (an information and control theoretic concept) to deduce the connectivity between neurons through the simulated data and real data [2]. Okatan et al. used a likelihood method to infer the pair-wise interactions among neurons [3]. Knowlton et al. used standardized voltage and synaptic gating variable waveforms to estimate the functional architecture of a small neural network [4].

2.2 Similarity measure for spike trains

In order to estimate the connectivity of the network, we need to use a similarity measure for spike trains at each node. Defining a distance between spike trains in a spike metric space is a most commonly used for measuring the similarity between spike trains. Various distances have been proposed including the Victor-Purpura distance [5], van Rossum distance [6], ISI (inter spike interval) distance [7] and spike distance [8].

Although distances are more common in measuring similarity, another approach is to use a positive definite kernel with an inner product [9]. The advantage of using kernels is that it enables the use of various machine learning methods to spike trains [8], including support vector machine (SVM), kernel principal component analysis, and kernel ridge regression [10].

2.3 Spike trains kernels

In the previous section we discussed the advantage of the kernel methods. Here we introduce various spike train kernels have been proposed. Shpigelman et al. defined the Spikernel, which is tied to time bins and worked well in the field of brain machine interface (BMI) [11]. Schrauwen and Van Campenhout proposed using the Laplacian kernel and the Gaussian kernel to analysis the spike trains [12]. Park et al. introduced the strictly positive definite kernels on the space of spike trains through kernel principal component analysis and hypothesis testing [13]. Li et al. defined a tensor-product kernel framework to analysis the multiscale activities in neural decoding [14] and the kernel least mean square on the space of spike trains [15]. Paiva et al proposed the memoryless cross intensity kernel (mCI kernel) and the nonlinear cross intensity kernel (nCI kernel) to convolute the spike trains [9, 16, 17]. The aim of this paper is to extend the mCI kernel to multichannel spike trains.

2.4 Other spike train metrics

There are various other proposals to define metrics on spike trains. Johnson et al. defined an information-theoretic distance on spike trains to compute the responses in neural coding [18]. Kreuz et al. proposed the ISI (inter-spike interval) metric by evaluating the ratio of the instantaneous firing rates and the SPIKE metric that takes the fundamental advantages of the ISI-distance [5, 6, 19]. Rusu and Florian proposed the modulus metric and the max metric to measure distances from each time point to the spike trains [20]. Compared to these metrics, the advantage of our kernel based approach is that it enables to use various tools from machine learning by extending linear optimization techniques to non-linear relationships in a natural way.

CHAPTER 3 – SIMULATION MODEL FOR SPIKE TRAINS

3.1 Coupled escape rate model for two neurons

It is well known that the firing rate is changed over time in neural code experiments. We use inhomogeneous Poisson process to simulate spike trains. We also use the coupled escape rate model (CERM) [21] to model interactions among neurons. It takes many parameters from a realistic network model.

Here we considered a case in which there are two neurons a, b with synaptic coupling. The coupled escape rate model is defined as follows.

$$\lambda_a(t) = \exp\left[\, u_a + \alpha_a x_a(t) + J_{ab}(t) s_{ab}(t) \,\right] \tag{1}$$

$$\lambda_b(t) = \exp\left[\, u_b + \alpha_b x_b(t) + J_{ba}(t) s_{ba}(t) \,\right] \tag{2}$$

$$\frac{dx_a}{dt} = -\frac{x_a}{\tau_m} + \sum_k \delta\,(t - t_{a,k}) \tag{3}$$

$$\frac{ds_{ab}}{dt} = -\frac{s_{ab}}{\tau_s} + \sum_k \delta\,(t - t_{b,k}) \tag{4}$$

$\lambda_i(t)$ is the instantaneous firing rate of neuron i, most parameters are indicated in Figure 1. $t_{i,k}$ is the k-th spike time of neuron i, and $\delta(t)$ is the Dirac delta function. Both the time constants τ_m and τ_s were 10 ms. (3) and (4) were calculated with a time step of 1 ms.

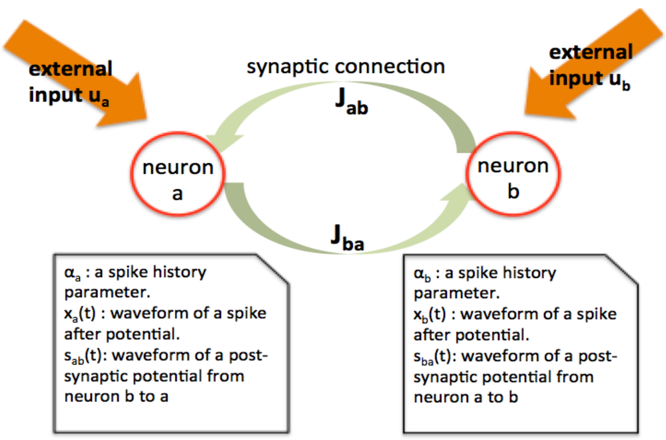

Figure 1 CERM model parameters: $\{u_{a,b}, a_{a,b}, J_{ab,ba}, x_{a,b}(t), s_{ab,ba}(t)\}$.

3.2 Network simulation

In this part, we extend the couple escape rate model to multi-neurons in a network because the CERM has a good property that it can imitate a wide variety of behaviors observed in real neural networks. The goal of our method is to estimate the strength of connectivity between neurons using spike trains only.

The coupled escape rate model is defined as follows.

$$\lambda_i(t) = \exp\left[\, u_i + \alpha_i x_i(t) + \sum_j J_{ij}(t) s_{ij}(t) \,\right] \tag{5}$$

$\lambda_i(t)$ is the instantaneous firing rate of neuron i, which is determined by its past event times and influences from other neurons. J_{ij} is the strength of the synaptic connection from neuron j to i. α_i represents a spike history function of neuron i, and $s_{ij}(t)$ is the effect from neuron j to i inducing or suppressing the forthcoming events. x_i and s_{ij} are determined by the following equations:

$$\frac{dx_i}{dt} = -\frac{x_i}{\tau_m} + \sum_k \delta\,(t - t_{i,k}) \tag{6}$$

$$\frac{ds_{ij}}{dt} = -\frac{s_{ij}}{\tau_s} + \sum_k \delta\left(t - t_{j,k}\right) \tag{7}$$

$t_{i,k}$ is the k-th spike time of neuron i, and δ is the Dirac delta function. Both the time constants τ_m and τ_s are set to 10 ms. (6) and (7) were calculated with a time step of 1 ms.

CHAPTER 4 – METHOD

4.1 Memoryless cross intensity kernel

There are many studies in the literature to use inner products to solve general machine learning problems. Here we use two spike train metrics to define inner products on functional representations of spike trains [9, 16, 17].

Consider two spike trains, x, y, $x=(x_1, x_2, \ldots, x_i)$, $y=(y_1, y_2, \ldots, y_j)$, with $i, j \in N$, and represent the spike train as a sum of Dirac delta functions $x(t) = \sum_i \delta(t - t_i)$, then define the intensity function by convolving the spike train and a smoothing function h:

$$\lambda(t) = \sum_i h(t - t_i) \qquad (8)$$

Here, t_i is the timing of the ith spike. The inner product can be defined as

$$k(x, y) = \int \lambda_x(t)\lambda_y(t)\,dt \qquad (9)$$

This is the memoryless cross intensity kernel (mCI kernel) proposed by Paiva, Park and Principe [16, 17]. Our aim in this paper is to extend the mCI kernel to multichannel spike trains. In our experiments, we used a Gaussian function for smoothing function h, which is a very common choice in the analysis of spike trains.

4.2 Multichannel spike train kernel

Our aim in this paper is to apply kernel ridge regression to multichannel spike trains, and evaluate it using simulated data.

Consider a case where there is a pair of multichannel spike train metrics $m^{(i)}$, $n^{(j)}$. Let $m^{(i)}=(m_1^{(i)}, m_2^{(i)}, \ldots m_d^{(i)})$ and $n^{(j)}=(n_1^{(j)}, n_2^{(j)}, \ldots n_d^{(j)})$ where each $m_d^{(i)}$ and $n_d^{(j)}$ indicates a spike train and $i, j \in N$. Here, $m^{(i)}$ will be called a component of m.

The most general way to define a kernel is to use a multichannel spike train metrics between $m^{(i)}$ and $n^{(j)}$ without imposing any structure [22, 23]. A kernel on multichannel spike trains can be expressed as k (m, n), here we can define the kernel on a pair of their components $m^{(i)}$ and $n^{(j)}$ as

$$k(m^{(i)}, n^{(j)}) = \sum_{i=1,j=1}^{d} P_{ij} k(m_i, n_j) \qquad (10)$$

where P is a coefficient matrix representing interchangeability or collaborative coding among spike trains [14]. When no prior information regarding the connectivity is provided, matrix P is set to P_{ii} =p and P_{ij} =q for $i \neq j$.

The kernel for multichannel spike trains can be written as

$$k(\mathrm{m},\mathrm{n}) = \begin{bmatrix} k\left(m^{(1)},n^{(1)}\right) & k\left(m^{(1)},n^{(2)}\right) & \cdots & k\left(m^{(1)},n^{(N)}\right) \\ k\left(m^{(2)},n^{(1)}\right) & k\left(m^{(2)},n^{(2)}\right) & \cdots & k\left(m^{(2)},n^{(N)}\right) \\ \vdots & & \ddots & \vdots \\ k\left(m^{(N)},n^{(1)}\right) & k\left(m^{(N)},n^{(1)}\right) & \cdots & k\left(m^{(N)},n^{(N)}\right) \end{bmatrix} \tag{11}$$

4.3 Kernel ridge regression

Kernel ridge regression is a non-linear extension of linear regression. One important benefit of kernel ridge regression is that it can be used for non-vector data. Once a similarity measure between data elements are defined in terms of a positive definite kernel, regression can be conducted in an implicit inner product space. This space is called a reproducing kernel Hilbert space.

Let $X=[x^{(1)},\ldots x^{(n)}]^T$ be independent variables and $y=[y^{(1)},\ldots y^{(n)}]^T$ be independent variables, where n is the number of samples. Linear regression can be expressed as follows,

$$\mathrm{y} = \mathrm{X}\mathrm{w}^{\mathrm{T}} \tag{12}$$

where w is the weight vector and $x^{(i)}=(x_1^{(i)}, x_2^{(i)},\ldots x_d^{(i)})$.

The solution for w is obtained by the normal equation, namely

$$\mathrm{w} = (\mathrm{X}^{\mathrm{T}}\mathrm{X})^{-1}\mathrm{X}^{\mathrm{T}}\mathrm{y} \tag{13}$$

Alternatively, we consider the case where parameter w is expressed as $\mathrm{w} = \mathrm{a}^{\mathrm{T}}\mathrm{X}$, so (12) can be rewritten as $\mathrm{y} = \mathrm{X}\mathrm{X}^{\mathrm{T}}\mathrm{a}$. Such representation is possible in the case of linear regression, due to the representer theorem [24]. Therefore, a can be computed as

$$\mathrm{a} = (\mathrm{X}\mathrm{X}^{\mathrm{T}})^{-1}\mathrm{y} \tag{14}$$

This solution is viable only if the inverse of X^TX exists. This is not always true. To overcome this problem, a regularization parameter ρ is introduced [25],

$$a = (X^T X + \rho\mathrm{I})^{-1} y \qquad (15)$$
$$= (K + \rho\mathrm{I})^{-1} y$$

where ρ>0 is a scalar and I is the D dimensional identity matrix. $K = XX^T$ is called the kernel matrix. All of the entries of K are inner products of the independent variables. This formulation indicates that we just need to know the inner products between data, which makes the computation much easier. In addition, it indicates that regression can be conducted on non-vector data such as multichannel spike trains.

To predict a new synaptic connection between neurons, the similarity measures between multichannel spike trains are defined. The prediction of the dependent variable for test data (multichannel spike train) x' can be obtained by

$$\mathrm{y}' = \mathrm{w}^{\mathrm{T}}\mathrm{x}' = \sum_{\mathrm{i}}^{\mathrm{N}} \mathrm{a_i} \mathrm{k}(\mathrm{x_i}, \mathrm{x}') \qquad (16)$$

CHAPTER 5 – EVALUATION

In this section, we used the couple rate model for two neurons and for a network to simulate the spike trains and estimated by a distance and kernel methods.

5.1 Evaluation for a distance

5.1.1 Simulation data

We used the coupled escape rate model defined in the previous section. There are two neurons used in the simulation that produce Poisson spike trains with time-varying firing rates.

Most of the parameters were summarized in Table 1. The data had been classified into ten conditions by changing u_1=[1.4: 0.2: 3.2]. For each condition, 20 trials were carried out. The time step is set to 1 ms and each trial lasted for 500ms. Our simulation model was run using MATLAB.

Table 1 Summary of parameters used in our simulation model.

a_1, a_2	-0.6, -0.9
J_{12}, J_{21}	-0.5, -0.4
u_1	[1.4 : 0.2 : 3.2]
u_2	1.7
$x_1(1)$, $x_2(1)$	0, 0
$s_{12}(1)$, $s_{21}(1)$	0, 0

With the same stimuli, the neuron will get different respond which one maybe has subtle differences. During each condition, we just change the external input to neuron 1, and get 200 times spike trains data in total. So when u_1= 1.4, we can get 20 times spike trains as t=(t^1, t^2, t^3, ... , t^{20}), the same to when u_1= 1.6, another 20 times spike trains as s=(s^1, s^2, s^3, ... , s^{20}) and so on.

As the result of the simulation model, in Figure 1 we compute the average firing rate of multichannel spike trains.

Since we get multichannel spike trains t and s using the coupled escape rate model, we should evaluate them by changing it to a distance. This is because a distance is usually used as a basis for classification, regression and other means of analyzing data.

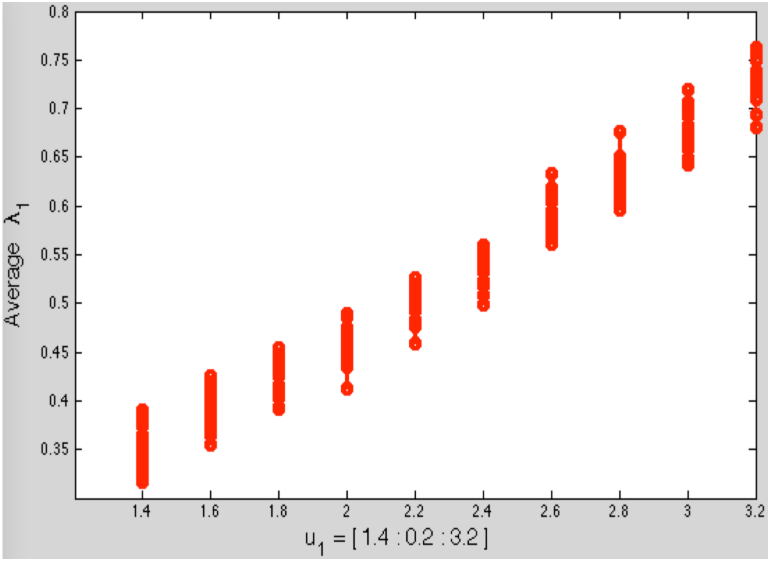

Figure 1. When changing u_1=[1.4: 0.2: 3.2], we will get ten conditions with each condition carried out for 20 times. We can see a subtle difference for each condition by using the average firing rate of multichannel spike trains.

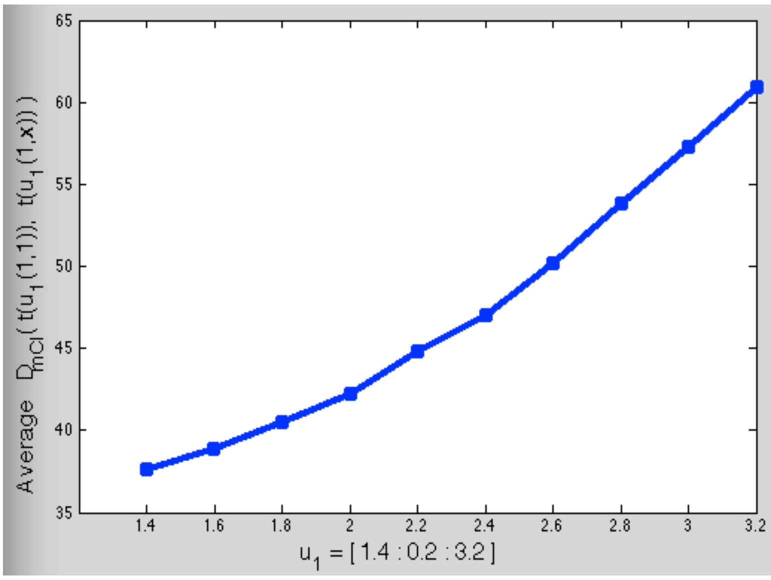

Figure 2. We calculate the norm distance for ten times, one is for the base condition $D_{mCI}(t_{u1},t_{u1})$, and others are between the base condition and each other condition $D_{mCI}(t_{u1},t_{ux})$, x = 2,3, ... ,10.

The norm distance is a commonly used distance obtained from a kernel [5]. The norm distance between two spike trains is defined as follows.

$$d(t,s) = \sqrt{k(t,t) - 2k(t,s) + k(s,s)} \qquad (17)$$

As this experiment has proposed for ten conditions. In order to see the obvious difference, we set $u_1 = 1.4$, $t=(t^1, t^2, t^3, \dots, t^{20})$, as the base condition to compare to other conditions, $s=(s^1, s^2, s^3, \dots, s^{20})$. From Figure 2, we can see a result that the distance of $D_{t,s}$ 9 times is larger than the distance of $D_{t,t}$. This means that with the same u_1, the distance between spike trains (t^i, t^j) is small and with the different u_1, the distance between spike trains (t^i, t^j) is much smaller than the distance between spike trains (t^i, s^j).

5.1.2 Comparison with other parameters

Since we get the result by changing the parameter of u_1, it's necessary to prove this conclusion in the other way. So we do the same experiment with changing the parameter of J_{12} and a_1.

In the evaluation with a_1 =[-0.6: 0.05: -0.15], we used their original parameter settings which obtained u_1=1.4. The result is showed in Figure 3 and Figure 4. AS there is a big change with the average firing rate, the difference of the norm distance is large.

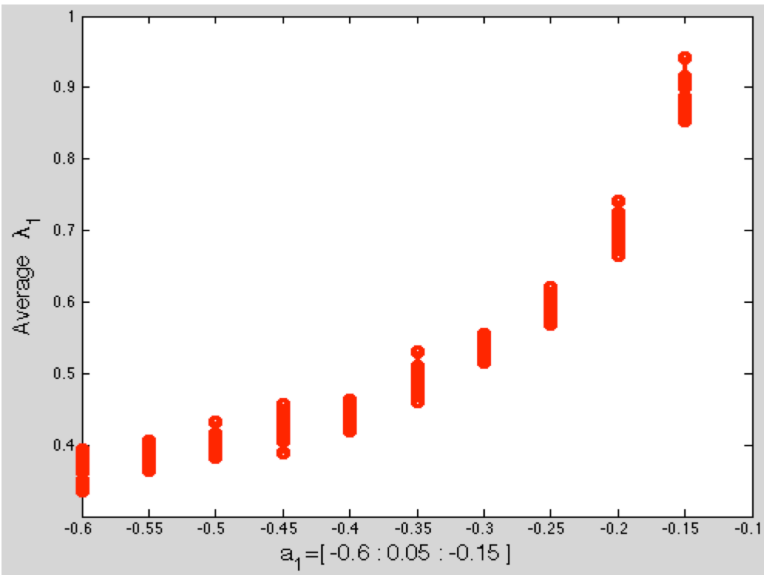

Figure 3. When changing a_1=[-0.6: 0.05: -0.15], the average firing rate of multichannel spike trains.

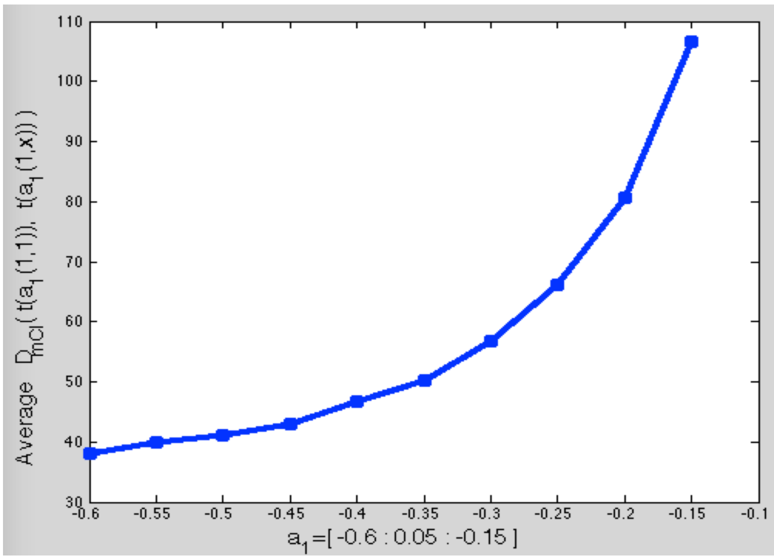

Figure 4. We consider a_1 = -0.6 as the base condition and calculate the norm distance for ten times.

For the condition of J_{12} =[-0.5: 0.05: -0.05], we also can the same result from Figure 5 and Figure 6.

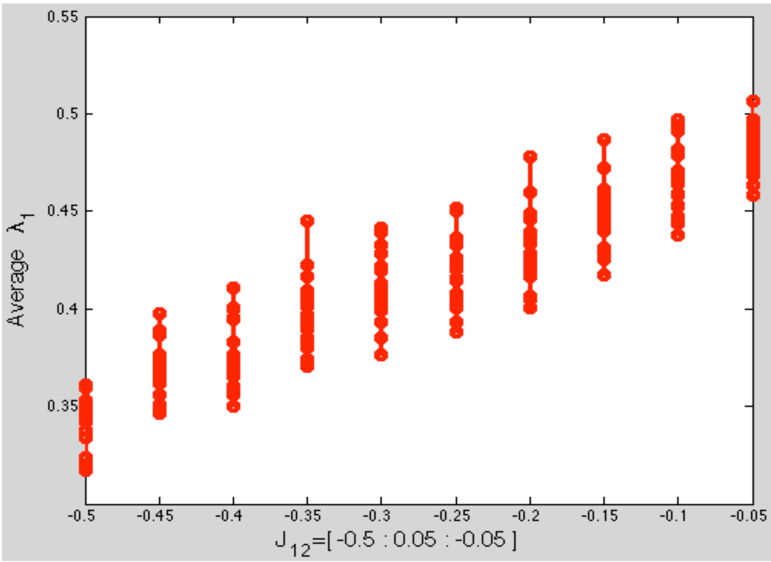

Figure 5. When changing J_{12}=[-0.5: 0.05: -0.05], the average firing rate of multichannel spike trains.

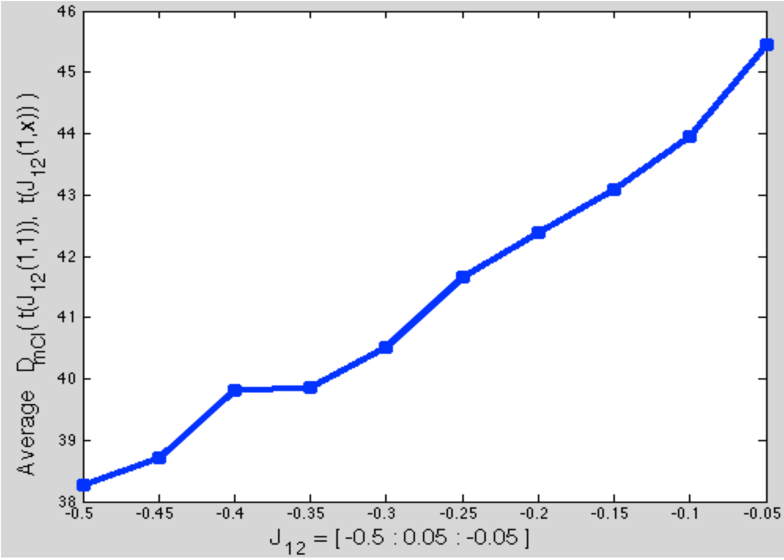

Figure 6. We consider J_{12} = -0.5 as the base condition and calculate the norm distance for ten times.

5.2 Evaluation for network

As the coupled escape rate model worked well in the previous section, we extend the CERM to a network and predict the strength of connectivity of the network. We assume that the parameters of the network simulated by CERM are known. So we can focus on the spike timing information alone to estimate the synaptic weights of the network. The number of neurons in the network was set to 10. The strength of the synaptic connection J can be both positive and negative, representing excitatory and inhibitory connections, respectively. In this paper, we aimed at estimating the strength of connection for a single pair of neurons, represented by J_{12}. Other synaptic connections J_{ij} are sampled randomly from the uniform distribution on [0, 1]. The training data is generated for 10 conditions by setting the value of J_{12} in [-5: 1.5: 8.5]. For each condition, 32 trials were carried out, which means the size of the kernel matrix for training data was 320 by 320. The test data of J_{12} also had been classified into 10 conditions by changing the value of J_{12} in [-2: 1: 7]. For each condition, 100 trials were carried out.

We conducted 3 types of experiments to evaluate the proposed method. Specifically, we compared the correct strength of connectivity J_{12} with the estimated strength of connectivity. In all of the box plot figures, the blue box is the estimated strength of connectivity, and the block circle is the correct strength of connectivity. The common parameters are set toα_i = 0.1, u_i = -9, $x_i(1)$ = 0, $s_{ij}(1)$ =0, the regularization parameter ρ = 1, and the Gaussian parameter σ= 5. The time step is set to 1 ms and each trial lasted for 500 ms. All the computation was run using MATLAB.

Figure 7 is a raster plot indicating an example of multichannel spike trains generated by CERM. Each dot represents an occurrence of a spike. One can see it is a very demanding task to estimate how the neurons are connected to each other just by looking at these spike trains. What we want to propose in this paper is a method of uncovering the relationships among neurons based solely on such multichannel spike train data.

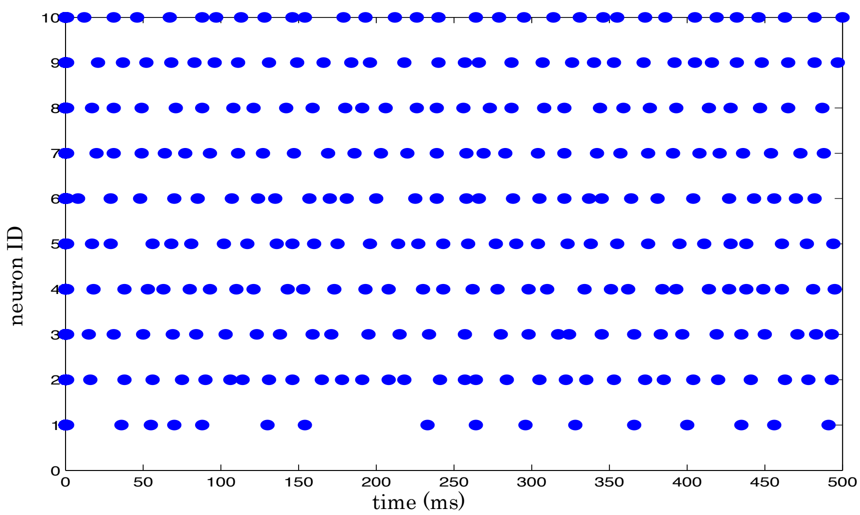

Figure 7. Example multichannel spike train generated by CERM neuron

Experiment 1: The coefficient matrix P for training data was changed. We set the number of neurons N = 10, p = 1, q taken from [-1, -0.5, 0, 0.5, 1], and for each condition and trail, the other synaptic connection J was unchanged. From Figure 8, we can see a result that the curve fits the best when q = 1. This indicates that there is a collaborative coding among neurons.

In Figure 9, we compare the results when the strength of connectivity for other pairs of neurons was changed. The result is similar to that of Figure 2.

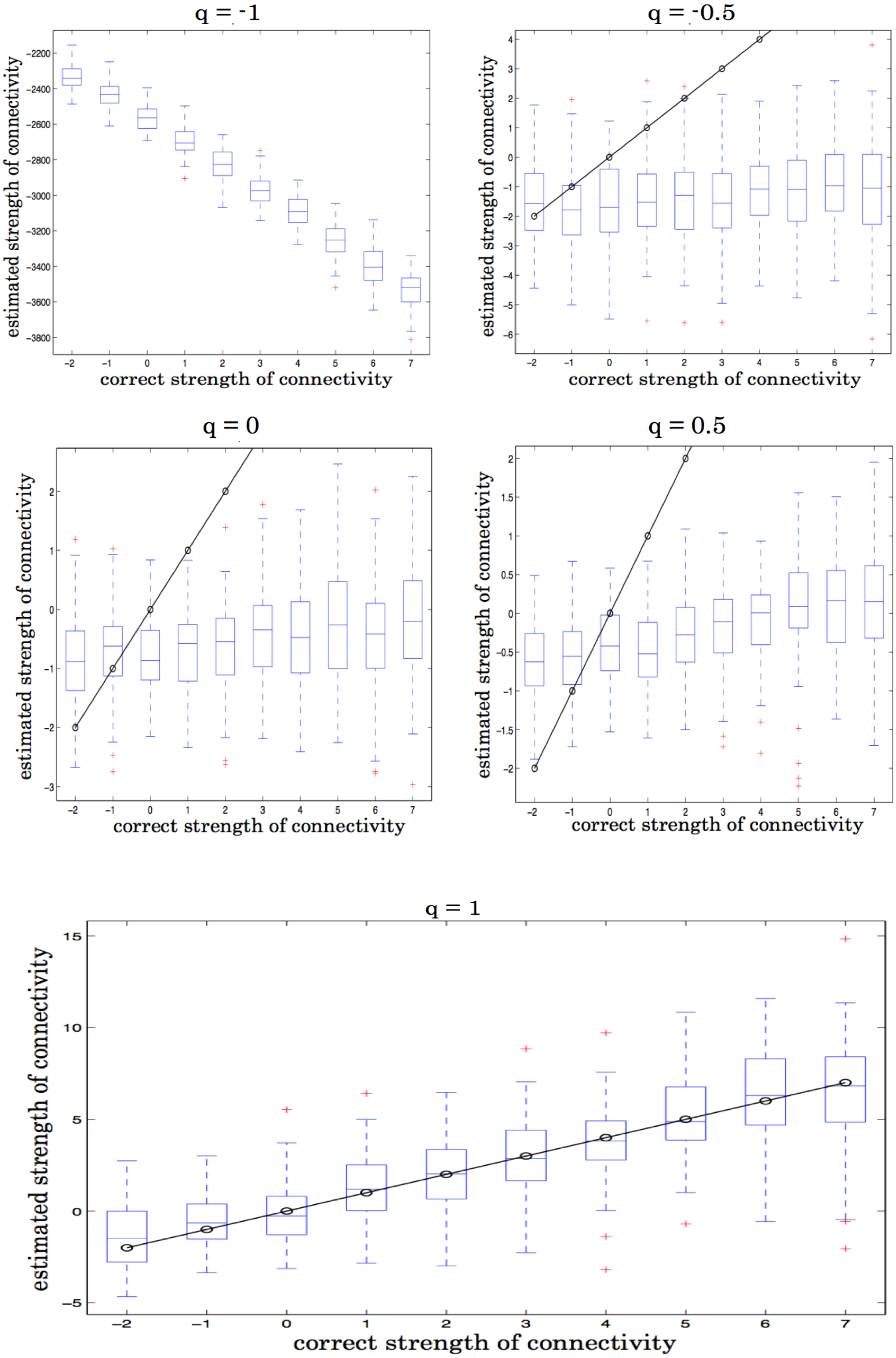

Figure 8. The correct strength of connectivity and the estimated strength of connectivity when the strength of connectivity for other pairs unchanged

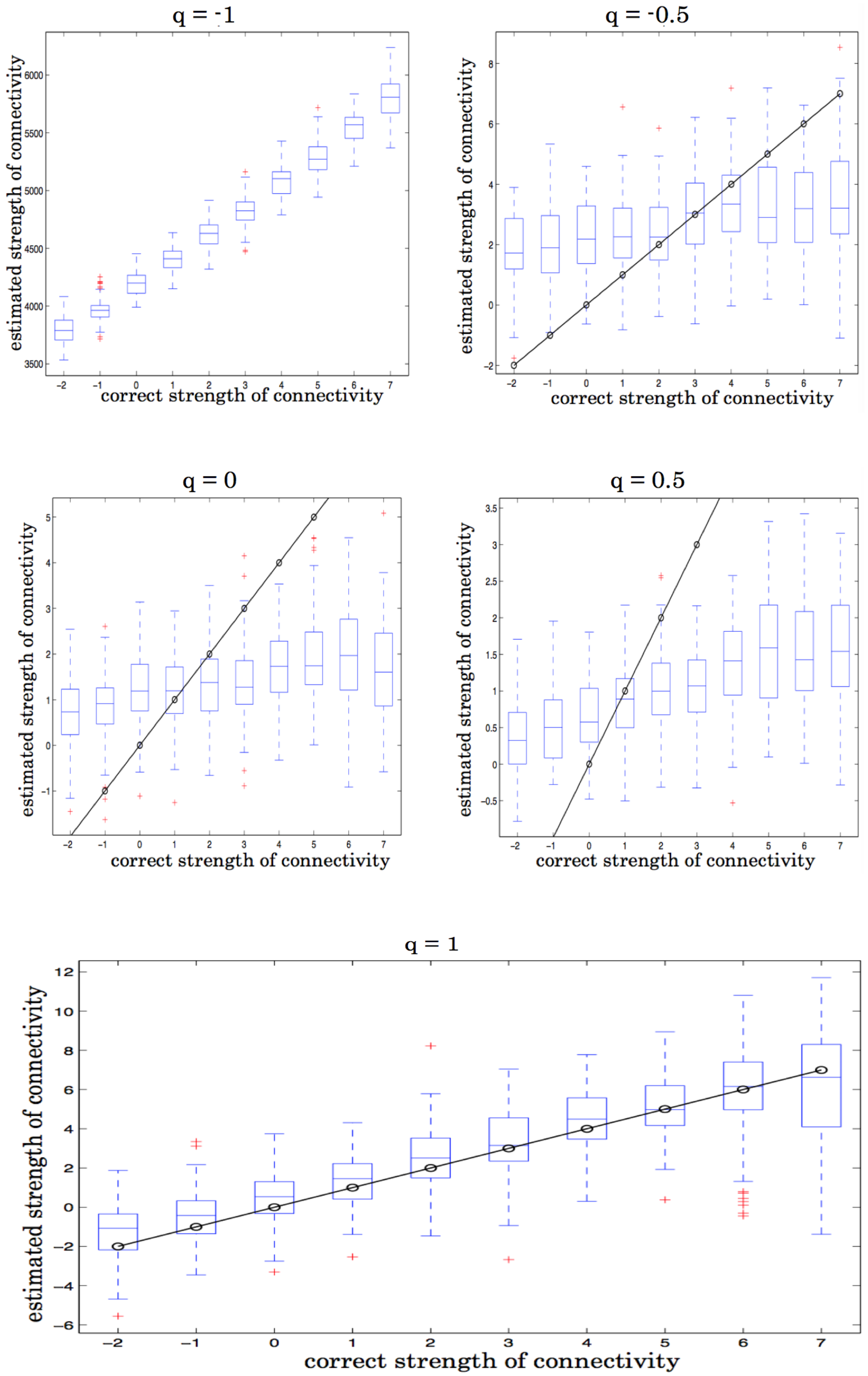

Figure 9. The correct strength of connectivity and estimated strength of connectivity when the strength of connectivity for other pairs was changed.

Experiment 2: The number of neurons N in the network was changed. As indicated in Figure 8 and Figure 9, the best result was obtained when q = 1. We therefore set p = 1, q = 1, J_{12} = 3, and tested for N taken from [2, 4, 10, 15, 20, 25, 30]. Figure 10 indicates the result. When the number of neurons increases, the behavior of the network becomes more complicated, and it becomes more difficult to estimate the strength of connectivity using a pair of spike trains alone. In such a case, we are considering to use spike trains observed at other neurons as well.

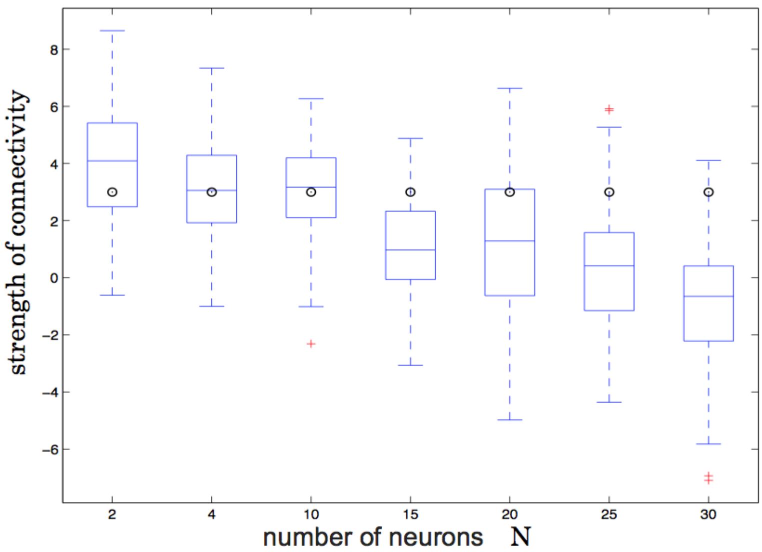

Figure 10. When the number of neurons N was changed, the result for the correct strength of connectivity and estimated strength of connectivity

Experiment 3: The error of estimation was compared as the total number of training samples M was changed. We set p = 1, q = 1, the number of neurons N to 10, and M taken from [40, 80, 160, 320, 640]. We evaluated the error using the root mean squared error (RMSE). Figure 11 indicates that the error will becomes smaller as the sample size increases.

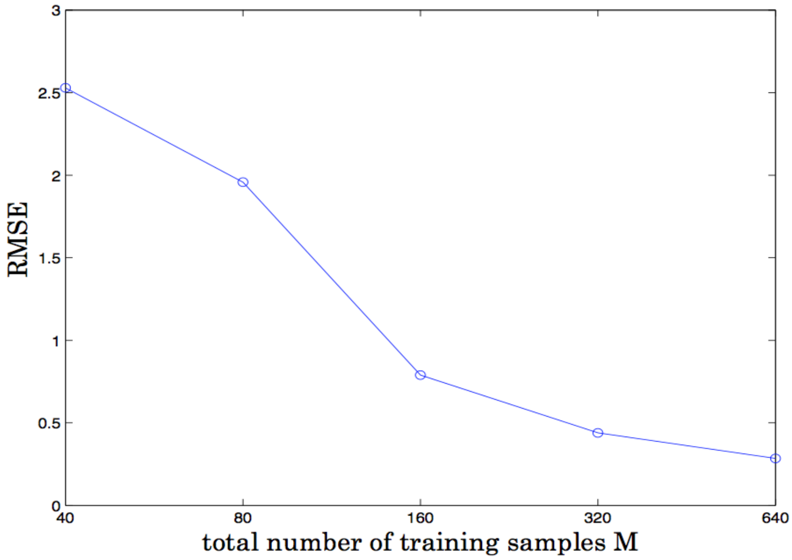

Figure 11. RMSE (root mean squared error) of estimating the strength of connectivity as the total number of training samples M was changed.

CHAPTER 6 – CONCLUSION

In this paper, we used a distance and kernel methods to analyze multichannel spike trains. We also proposed to use kernel ridge regression on multichannel spike trains as a way to estimate the strength of connectivity in a network when only the knowledge of the spike trains at each neuron is given. We tested our method using different parameters, and indicated that it can be effectively used for estimating the connectivity strength.

In future work, we plan to extend this method to estimate the directions of the connectivity. We also plan to use other simulation models as well, which will make simulated data more similar to that obtained from biological neural networks. Finally, we are planning to apply our method on real data as well, and evaluate how accurately it can estimate actual connections between neurons in a biological neural network.

ACKNOWLEDGEMENTS

This master thesis has been carried out at the Graduate School of Library, Information and Media Studies, University of Tsukuba in 2015. I would like to express my warmest gratitude to my supervisor, Dr. Taro Tezuka, for having accepted me as his student, and for his experienced guidance and advice. Furthermore I would like to thank Dr. Tetsuji Satoh, Dr. Kei Wakabayashi, Dr. Masahiko Mikawa, for serving as my committee member, and giving many valuable advice to my research. Also, I like to thank Yasumasa san, Nakaoka san, Furutani san, Zhong san, Atenna san, Arimura san, Sayama san, Yamakoshi san, Yamauchi san, Shibasaki san, Aida san, Ikeda san for the wonderful time in the Lab.

REFERENCES

[1] C.W.J. Granger: Investing causal relations by econometric models and cross-spectral methods, Econometrica, Vol. 37, No. 3, pp. 424-438 (1969).

[2] Christopher J. Quinn, Todd P. Coleman, Negar Kiyavash and Nicholas G. Hatsopoulos: Estimating the directed information to infer causal relationship in ensemble neural spike train

recordings, Journal of Computational Neuroscience (2010).

[3] M. Okatan, M.A. Wilson, and E.N. Brown: Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity, Neural Computation, Vol.17, pp.1927-1961 (2005).

[4] Chris Knowlton, C. Daninel Meliza, Daniel Margoliash, and Henry D. I. Abarbanel: Dynamical estimation of neuron and network properties III: network analysis using neuron spike times, Springer, Vol.108, Issue 3, pp.261-273 (2014).

[5] Conor Houghton and Jonathan Victor: Measuring representational distances – the spike train metrics approach, The MIT Press; 1 edition, pp. 213-245 (2011).

[6] Thomas Kreuz, Daniel Chicharro, Ralph G. Andrejak, Julie S. Haas and Henry D.I. Abarbanel: Measuring multiple spike train synchrony, Journal of Neuroscience Methods, Vol.183, No.2, pp. 287-29 (2009).

[7] Thomas Kreuz, Daniel Chicharro, Conor Houghton, Ralph G. Andrzejak and Florian Mormann: Monitoring spike train synchrony, The Journal Neurophysiology 109, pp. 1457-1472 (2013).

[8] 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).

[9] Antonio. R. C. Paiva: Reproducing kernel Hilbert space for point processes with applications to neural activity analysis, Ph.D. thesis, University of Florida (2008).

[10] Bernhard Scholkopf and Alexander J. Smola.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press (2001).

[11] L.Shpigelman, Y.Singer, R. Paz and E. Vaadia: Spikernels: Predicting arm movements by embedding population spike rate patterns in inner product spaces, Neural Computation, Vol. 17, No. 3, pp. 671–690 (2005).

[12] Schrauwen Benjamin and Van Campenhout Jan: Linking non-binned spike train kernels to several existing spike train metrics. Neurocomputing, Vol. 70, No. 7-9, pp. 1247–1253 (2007).

[13] 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).

[14] Li, L., Brockmeier, A.J., Choi, J.S., Francis, J.T., Sanchez, J.C., and Jose C. Principe: A tensor-product-kernel framework for multiscale neural activity decoding and control.

Computational Intelligence and Neuroscience (2014).

[15] Lin Li, Il Memming Park, Sohan Seth, John S. Choi Joseph T. Francis, Justin C. Sanchez, and Jose C. Principe: An adaptive decoder from spike trains to micro-stimulation using kernel least-mean-squares (KLMS), MLSP, pp. 1-6 (2011).

[16] 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).

[17] 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).

[18] 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).

[19] Conor Houghton and Thomas Kreuz: Measures of spike train synchrony: From single neurons to populations, In Multiscale analysis and nonlinear dynamics: from genes to the brain. Eds: Misha Meyer Pesenson and Heinz Georg Schuster, Wiley (2013).

[20] Catalin V. Rusu and Razvan V. Florian, R: A new class of metrics for spike trains. Neural Computation, Vol. 26, No. 2, pp. 306-348 (2014).

[21] 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).

[22] Taro Tezuka: Spike train kernels for multiple neuron recordings, Proceedings of the 39th International Conference on Acoustics, Speech and Signal Processing (ICASSP2014), Florence, Italy, pp.6035-6039 (2014).

[23] Taro Tezuka: Parallel spike trains analysis using positive definite kernels, Proceedings of the 23rd Annual Computational Neuroscience Meeting (CNS2014), Quebec City, Canada (2014).

[24] S.Y. Kung, Kernel Methods and Machine Learning, Cambridge University Press (2014).

[25] John Shawe-Taylor and Nello Cristianini.: Kernel methods for pattern analysis. Cambridge University Press, New York (2004).