Spectral Clustering of Synchronous Spike Trains

António R. C. Paiva, Sudhir Rao, Il Park and José C. Príncipe

***Abstract*— In this paper a clustering algorithm that learns the groups of synchronized spike trains directly from data is proposed. Clustering of spike trains based on the presence of synchronous neural activity is of high relevance in neurophysiological studies. In this context such activity is thought to be associated with functional structures in the brain. In addition, clustering has the potential to analyze large volumes of data. The algorithm couples a distance between two spike trains recently proposed in the literature with spectral clustering. Finally, the algorithm is illustrated in sets of computer generated spike trains and analyzed for the dependence on its parameters and accuracy with respect to features of interest.**

I. Introduction

Current neuroscience research aims at understand how groups of neurons collectively represent information, in opposition to the single neuron studies done is the past [1]. Recent advances in recording devices now allow for multiple-electrode recordings that simultaneous collect the activity of many neurons (more that 20) [2]. In fact, current paradigms may require the simultaneous recording of more than 100 neurons [2]. Analysis of neural activity is not only fundamental for neurophysiological studies, but also for applications of this knowledge, such as Brain Machine Interfaces (BMIs) which directly try to map neural activity into behavior [3]–[5].

Yet, the enormous volumes of data collected in neural recordings make the analysis of this data a daunting task. In addition, because the spike train is defined only it terms of the event times (the firing or spike times) it is in fact a point process. Thus, standard statistical signal and data analysis tools, such as clustering, are of limited use [1]. This is because these methods were designed to operate on continuous random processes.

Clustering is an unsupervised learning method developed in machine learning and pattern recognition that learns groups of related data points directly from data [6], [7]. In the analysis of spike trains, clustering is of particular interest since the clusters represent groups of neurons yielding "similar" spike trains. Therefore, neurons responsible for spike trains within a cluster are likely to be connected, directly through synaptic connections or indirectly through other neurons. (In neuroscience these functional groups of neurons are called *neural assemblies* and are thought to be fundamental cognitive blocks in the brain [8].) In addition, compared to other statistical tools for the analysis of neural activity [1], clustering is a tool designed for the analysis of large volumes of data and accounting for information from all input spike trains. Note that in establishing these connections similarity is defined directly in terms of synchrony in the spike times [8], [9].

To the best of our knowledge, clustering of spike trains until now has relied mainly on binning[1] the neural activity [10], [11]. Although this approach is highly attractive since the stochasticity in time of the point process is transformed to randomness in the amplitude of a continuous random process and, therefore, conventional clustering methods can be employed. However, this approach quantizes time in a very coarse manner which largely disregards any interactions between single spikes. In fact, this approach is only valid if the similarity between spike trains is defined in terms of their firing rate patterns. If instead we define similarity in terms of spike synchrony then the approach of Victor and Purpura [12] can be used. In fact, their method is applicable with any measure of synchrony, but the caveat in this approach is that it assumes a priori knowledge of some reference spike trains which function as cluster prototype data points. Naturally, this is an unrealistic requirement is the general case.

In this paper we propose a clustering method that, in true spirit of clustering, learns the groups of spike trains entirely from the data and in an unsupervised manner. We will start by defining a distance between two spike trains as proposed by van Rossum [13]. Using this distance the Gaussian kernel maps this distance into the entries of an affinity matrix containing the distance between all pair combinations of spike trains. Then, spectral clustering as proposed by Ng et al. [14] is applied to the affinity matrix to derive the clusters. One of the main contributions of this work is the novel coupling of the spike train distance with spectral clustering in the way the affinity matrix is defined to achieve truly unsupervised clustering.

The remainder of the paper is organized as follows. In section II the distance between two spike trains required for clustering is defined. Then, in section III, the procedure for the construction of the affinity matrix and subsequent spectral clustering algorithm are presented. Section IV shows the application of the proposed algorithm for clustering of sets of simulated spike trains under various conditions. We conclude with a summary of the main accomplishments in section V.

II. Distance between two spike trains

For clustering of data in any format the first requirement is the definition of distance that evaluates how "close" the data points are in some space. In machine learning and pattern recognition, clustering is typically applied to data

The authors are with the Computational Neuroengineering Laboratory (CNEL), Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611, USA (phone: 352-392-2682; fax: 352-392-0044; email: {arpaiva,sudhir,memming,principe}@cnel.ufl.edu)

[1] Counting the number of spikes in a sliding window of time.

points in $\mathbb{R}^n$, where n is the dimensionality of the data. Although many distances can be defined, a natural choice is the Euclidean distance. Spike trains however are point processes which are completely specified as a set of spike times and, for this reason, cannot be clustered as commonly devised.

As briefly reviewed in the introduction, if clustering is to be done in terms of firing rate patterns then binning can be used to transform the spike trains into n dimensional vectors, with n the number of bins. On the other hand, if the goal is to cluster spike trains based on synchronous neural activity, as intended here, then a distance directly dependent on the spikes times must be used. The reader may think that binning with small bin sizes ($1 \sim 5$ms) could be utilized but we must remark that in this situation the Euclidean distance is not sensitive to spikes in neighboring bins which may introduce artifacts in the distance measure [15].

In the literature two definitions of spike train distances have gained notorious attention: Victor and Purpura's non-Euclidean metric [12], [16] and van Rossum's [13] distance. Both of these distances utilize the full resolution of the spike times, but the latter distance is conceptually and computationally simpler. Anyway, except for slight adjustments due to differences in the dynamic range, the algorithm presented should work with either of these distance measures.

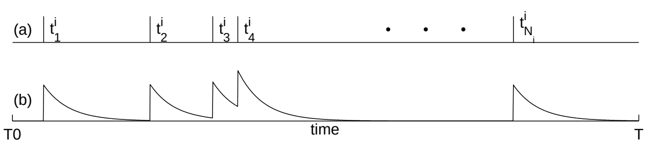

Fig. 1. Filtered spike train using the causal decaying exponential function.

Simply put, van Rossum's distance [13] is an euclidean distance on the continuous functions obtained as filtered spike trains. Given a spike train $s_i \in S$ with spike times $\{t_m^i : m = 1, \ldots, N_i\}$ the filtered spike train can be written as

$$f_i(t) = \sum_{m=1}^{N_i} h(t - t_m^i), \quad (1)$$

where N_i is the number of spike in the recording interval and $h(t)$ is the smoothing filter impulse response. van Rossum proposed to use a causal decaying exponential function, $h(t) = \exp(-t/\tau)u(t)$, with $u(t)$ the Heaviside step function, for simplicity and because it is biologically plausible. Figure 1 exemplifies this concept. Then, the distance between the ith and jth spike trains is defined as

$$d_{ij} = \frac{1}{\tau} \int_0^\infty [f_i(t) - f_j(t)]^2 \, dt. \quad (2)$$

However, evaluation of an integral is computational demanding and troublesome. Luckily, for the proposed smoothing filter the expression can be evaluated directly on the spike times. Substituting (1) with the exponential filter in (2) and evaluating the integral yields,

$$d_{ij} = \frac{1}{2} \left[\sum_{m=1}^{N_i} \sum_{n=1}^{N_i} L_\tau(t_m^i - t_n^i) + \sum_{m=1}^{N_j} \sum_{n=1}^{N_j} L_\tau(t_m^j - t_n^j) \right] + \sum_{m=1}^{N_i} \sum_{n=1}^{N_j} L_\tau(t_m^i - t_n^j), \quad (3)$$

where $L_\tau(\cdot) = \exp(-|\cdot|/\tau)$ is the Laplacian function.

Notice that this distance has one parameter, τ. This parameter controls the effect of near perfectly synchronized spikes. Because perfect synchrony is virtually impossible (zero probability in true continuous time) to be found in real data, the Laplacian function allows for some noise in the spike times, and τ controls the width of the time interval.

III. Spectral clustering algorithm

Let $S = \{s_1, s_2, \ldots, s_n\}$ be the set of n spike trains to be clustered into k clusters. We use the spectral clustering algorithm proposed by Ng et al. [14] for its simplicity and small number of parameters. See Weiss [17] for a review. Spectral clustering was also shown to have a close relation with information theoretic methods [18]. The main difference and the main contribution of this work is in the way the elements of the affinity matrix $A \in \mathbb{R}^{n \times n}$ are computed. Note that the ijth entry of the affinity matrix quantifies the *similarity* between the ith and jth spike trains. The distance defined earlier is an effective measure of spike trains *dissimilarity*. So, distance and similarity are inversely related quantities. To achieve the desired effect, we apply the Gaussian kernel which nonlinearly scales and weights the distance between spike trains. In this manner, similar spike trains (smaller distance) have higher affinity values, whereas uncorrelated spike trains (higher distance) have smaller values.

The final algorithm, presented step-by-step, goes is as follows.

1) Compute the affinity matrix $A \in \mathbb{R}^{n \times n}$ from the n spike trains. The ijth entry of the affinity matrix is given by,

$$a_{ij} = \begin{cases} \exp\left(-\dfrac{d_{ij}^2}{2\sigma^2}\right), & \text{if } i \neq j \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

where d_{ij} is the distance between the ith and jth spike trains as defined earlier.

2) Construct D as a diagonal matrix with the ith element of the main diagonal equal to the sum of all elements in the ith row of A (or column, since A is symmetric). That is,

$$d_i = \sum_{j=1}^{n} a_{ij}.$$

3) Evaluate the matrix

$$L = (D^{-\frac{1}{2}})A(D^{-\frac{1}{2}}).$$

4) Find $x_1, x_2, \ldots, x_k$, the k eigenvectors of L corresponding to the largest eigenvalues, and form the matrix $X = [x_1, x_2, \ldots, x_k] \in \mathbb{R}^{n \times k}$.
5) Define $Y \in \mathbb{R}^{n \times k}$ as the matrix obtained from X after normalizing each row to unit norm. Consequently,

$$y_{ij} = \frac{x_{ij}}{\sqrt{\sum_{j=1}^{n} x_{ij}^2}}$$

6) Interpreting Y as a set of n points in $\mathbb{R}^k$, cluster these points into k clusters with k-means or similar algorithm.
7) Assign to the ith spike train the same label of the ith point (row) of Y.

The clustering itself only depends on one parameter: the Gaussian kernel size. This parameter regulates the decrease in the affinity value with the distances between two spike trains. Anyway, our experimental results revealed a large insensitivity to the actual kernel size used.

IV. Results

In this section, sets of simulated spike trains are used to illustrate the application of the algorithm and to study the influence of the parameters τ and σ in the analysis. We start by performing clustering on datasets under the ideal situation that spike times are perfectly coincident. Then, the algorithm behavior is studied when jitter noise is present in the spike times. This aims to model a more realistic scenario and to show that the algorithm still performs as expected.

A. Noiseless case

In the ideal case of perfect synchrony of the spike times, τ can be made as close to zero as desired (in which case the Laplacian function converges towards an impulse). Nevertheless, we were interested in evaluating the algorithm's performance, measured as ratio of correctly clustered spike trains, if higher values of τ were used. In addition, the main element for the performance of the algorithm was the synchrony level among spike trains. In this context, synchrony level denotes the relative frequency of synchronous spikes in the data.

In the analysis, 10 sets of spike trains were generated for each synchrony level and τ value. Each set comprised 100 spike trains modeled as homogeneous Poisson point processes and were 2 seconds long with average firing rate of 20 spikes/s. The spike trains were grouped in three clusters. Initially, each spike trains in this set, modeled as homogeneous Poisson point processes, was independently generated with average firing rate $(1 - \varepsilon)\lambda$, where ε is the synchrony level and λ the intended average firing rate (20 spikes/s in this case). Then, three reference spikes trains (corresponding to each cluster), also homogeneous and Poisson distributed, were generated with firing rate $\varepsilon\lambda$. The spikes in these spike trains define ensemble wide synchronous activity. With this purpose, the spike times from one of the latter spike trains randomly selected was copied to each of the initial 100 spike trains. An absolute refractory period of 3ms was enforced by removing the nearest spike to a synchronous spike if necessary.

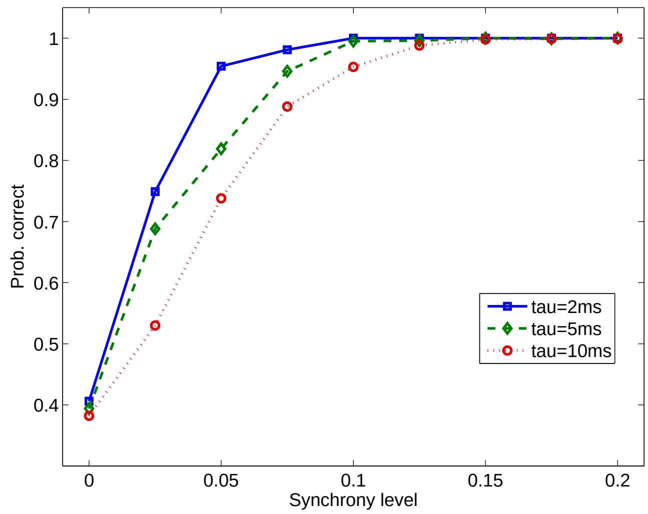

Fig. 2. Probability of correctly clustered spike trains for the noiseless case as a function of the synchrony level for three values of τ. The Gaussian kernel size (Eq. 4) was 10.

The relative frequency of correctly clustered spike trains averaged over 10 realizations (sets of spike trains) is shown in Fig. 2. As expected, the best results are achieved for the smallest τ. As discussed before, in this particular example τ could be chosen as close to zero as desired. In this context, intuitively we may think that increasing τ increases the number of spikes considered by the distance measure which are only due to chance and, therefore, only contribute "noise" to the distance estimate. Yet, as stated earlier, the synchrony level is the quantity that effectively controls how accurate the clustering is. Notice that for the values of τ chosen the distance measure employed is most sensitive to synchrony in the spike trains. For this reason, the synchrony level determines the separability of the clusters. Furthermore, for larger synchrony levels the clustering solution becomes insensitive to τ which is appealing in practice. One might think that in the independent case the spike trains are distributed across the space and progressively agglomerate around some point as the synchrony level increases. This perspective is depicted in Fig. 3 through the points of the Y matrix used in the clustering with k-means (step 6 of the algorithm). Finally, it is worth mentioning that although the results shown are for $\sigma = 10$ we have experimented with values in the interval $5 \sim 20$ without significant differences.

B. With jittered spike times

The scenario depicted in the previous example is idealistic. In real sets of spike trains perfect synchrony is extremely unlikely. Consequently, the spike train distance measure must be capable of accounting for spike times occurring close in time which may be considered synchronous. Of course, in doing so the distance also allows for spikes occurring "close" in time exclusively due to chance to be considered as

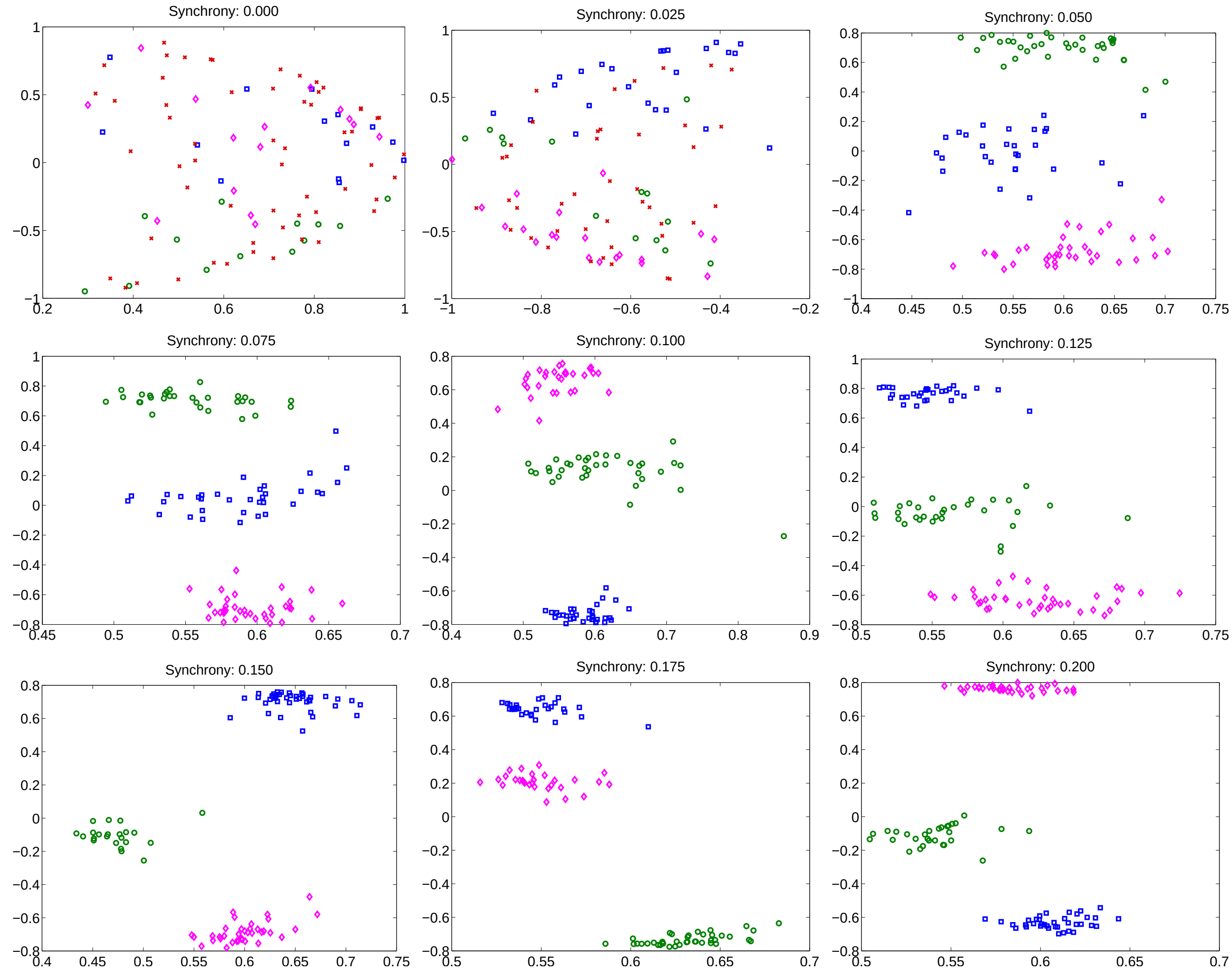

Fig. 3. Evolution of the clusters for the noiseless case as obtained from the Y matrix (step 6 of the algorithm) with the increase in synchrony level. In the plots, red crosses denote points corresponding to incorrectly clustered spike trains. The Gaussian kernel size and τ were set to 10 and 2ms, respectively.

synchronous. The distance proposed by van Rossum's [13], and which we use, deals elegantly with this situation by nonlinearly weighting the contribution of each spike with the Laplacian function. The τ parameter controls the effect of near perfectly synchronized spikes.

In this example, the performance of the clustering algorithm is studied in the presence of jitter noise in the firing times of the synchronous spikes. The generation of the testing sets of spike trains follow roughly the same steps as the previous example, except that when copying the ensemble synchronous spike times for each spike train, each spike time is disturbed with zero-mean Gaussian distributed noise.

Figure 4 summarizes the results for two levels of synchrony. With small synchrony level (i.e., smaller number of actually synchronized spikes) the clusters are not so well defined and are hard to discriminate. Consequently, it is not surprising the higher sensitivity to jitter. Conversely, for greater synchrony levels the good discrimination among clusters allows for some noise in the distance estimation due to jitter without sacrificing accuracy. Also noticeable in the figure is the faster decrease in performance for a synchrony level of 0.2 when $\tau = 2$ms. This is completely natural as a more stringent definition of synchrony is more likely to disregard synchronous spikes due to the jitter. As for the previous example, the Gaussian kernel size was not critical and values in $5 \sim 20$ yielded extremely similar results.

V. Conclusions

This paper proposes a clustering algorithm that finds groups of spike trains with synchronous activity. The proposed algorithm avoids the difficulty in computing template spike trains (i.e., cluster "centers") to which other spike trains can be matched. This is accomplished through the construction of an affinity matrix which quantifies the similarity between spike trains and to which spectral clustering can be applied. The results obtained suggest τ must be carefully selected according to the feature of interest and knowledge of the data, whereas Gaussian kernel size is not a crucial

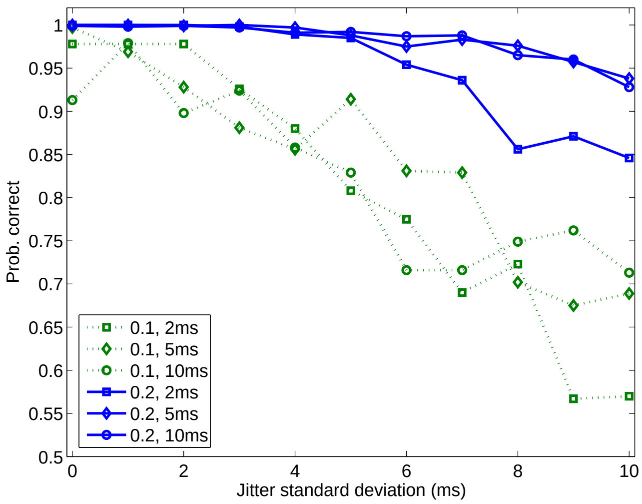

Fig. 4. Probability of correctly clustered spike trains as a function of the spike times jitter standard deviation for two synchrony levels (0.1 and 0.2) and three values of τ. The Gaussian kernel size was set to 10.

parameter and a wide range of values can be used.

The methodology employed here exemplifies the use of spectral clustering as a simple and effective perspective which eases in defining clustering techniques regardless of the nature of the data. All that is needed is a similarity measure that connects the domain of the data to the domain of the feature evaluated by the similarity measure. Indeed, as our results show, the clustering obtained in this way largely depend on which particular feature of the data points the similarity measure is sensitive. In the particular case considered here, the similarity was defined through a nonlinear mapping (Eq. 4) of van Rossum's distance [13] which for the range of time constants employed is most sensitive to the firing synchrony. Yet, if τ is increased considerably ($50 \sim 250$ms) then the focus of the measure may be directed towards similarity in the firing rate patterns. That is to say that in this method the distance measure time constant τ smears the distinction in analysis between firing synchrony and firing rates. It must be remarked however that such property is not exclusive of van Rossum's distance. For example, the parameter q in Victor and Purpura's [16] distance $D^{\text{spike}}[q]$ plays a similar role.

For future work it is worthwhile further understanding how the Gaussian kernel and the distance measure affect the clustering performance. This might suggest better mappings and/or distance measures. Moreover, a situation not analyzed in this work was the effect of the number of clusters. In the results we assumed the number of clusters to be known. However, in the general case the actual number of cluster is unknown and must be estimated.

Acknowledgements

This work was partially supported by NSF grants ECS-0601271, ECS-0422718 and CISE-0541241. A. R. C. Paiva was supported by Fundação para a Ciência e a Tecnologia under grant SFRH/BD/18217/2004.

References

[1] E. N. Brown, R. E. Kass, and P. P. Mitra, "Multiple neural spike train data analysis: state-of-the-art and future challenges," *Nature Neuroscience*, vol. 7, pp. 456–461, 2004.

[2] M. A. L. Nicolelis, C. R. Stambaugh, A. Bristen, and M. Laubach, "Methods for simultaneous multisite neural ensemble recordings in behaving primates," in *Methods for Neural Ensemble Recordings*, M. A. L. Nicolelis, Ed. CRC Press, 1999, ch. 7, pp. 157–177.

[3] M. A. L. Nicolelis, "Brain-machine interfaces to restore motor function and probe neural circuits," *Nature Reviews Neuroscience*, vol. 4, pp. 417–422, 2003.

[4] J. Wessberg, C. R. Stambaugh, J. D. Kralik, P. D. Beck, M. Laubach, J. K. Chapin, J. Kim, S. J. Biggs, M. A. Srinivasan, and M. A. L. Nicolelis, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, no. 6810, pp. 361–365, Nov. 2000.

[5] J. M. Carmena, M. A. Lebedev, R. E. Crist, J. E. O'Doherty, D. M. Santucci, D. F. Dimitrov, P. G. Patil, C. S. Henriquez, and M. A. L. Nicolelis, "Learning to control a brain-machine interface for reaching and grasping by primates," *PLoS Biology*, vol. 1, no. 2, Nov. 2003.

[6] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*, 2nd ed. Wiley Interscience, 2000.

[7] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*. Prentice Hall, 1988.

[8] G. L. Gerstein and M. R. Turner, "Neural assemblies as building blocks of cortical computation," in *Computational Neuroscience*, E. L. Schwartz, Ed. MIT Press, 1990, pp. 179–191.

[9] W. A. Freiwald, A. K. Kreiter, and W. Singer, "Synchronization and assembly formation in the visual cortex," in *Progress in Brain Research*, M. A. L. Nicolelis, Ed. Elsevier Science, 2001, pp. 111–140.

[10] K. MacLeod, A. Bäcker, and G. Laurent, "Who reads temporal information contained across synchronized and oscillatory spike trains?" *Nature*, vol. 395, pp. 693–698, Oct. 1998.

[11] W. S. Geisler, D. G. Albrecht, R. J. Salvi, and S. S. Saunders, "Discrimination performance of single neurons: rate and temporal-pattern information," *Journal of Neurophysiology*, vol. 66, no. 1, pp. 334–362, Jul. 1991.

[12] J. D. Victor and K. P. Purpura, "Metric-space analysis of spike trains: theory, algorithms, and application," *Network*, vol. 8, pp. 127–164, Oct. 1998.

[13] M. C. W. van Rossum, "A novel spike distance," *Neural Computation*, vol. 13, no. 4, pp. 751–764, 2001.

[14] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 14, 2001.

[15] S. Gruñ, M. Diesmann, F. Grammont, A. Riehle, and A. Aertsen, "Detecting unitary events without discretization of time," *Journal of Neuroscience Methods*, vol. 93, no. 1, pp. 67–79, Dec. 1999.

[16] J. D. Victor, "Spike train metrics," *Current Opinion in Neurobiology*, vol. 15, no. 5, pp. 585–592, Sep. 2005.

[17] Y. Weiss, "Segmentation using eigenvectors: a unifying view," in *Proc. of Intl. Conf. on Computer Vision*, 1999.

[18] R. Jenssen, "An information theoretic approach to machine learning," Ph.D. dissertation, University of Tromsø, 2005.