Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Clustering Neural Spike Trains with Transient Responses

John D. Hunter, Jianhong Wu and John G. Milton

Abstract— **The detection of transient responses, i.e. non–stationarities, that arise in a varying and small fraction of the total number of neural spike trains recorded from chronically implanted multielectrode grids becomes increasingly difficult as the number of electrodes grows. This paper presents a novel application of an unsupervised neural network for clustering neural spike trains with transient responses. This network is constructed by incorporating projective clustering into an adaptive resonance type neural network (ART) architecture resulting in a PART neural network. Since comparisons are made between inputs and learned patterns using only a subset of the total number of available dimensions, PART neural networks are ideally suited to the detection of transients. We show that PART neural networks are an effective tool for clustering neural spike trains that is easily implemented, computationally inexpensive, and well suited for detecting neural responses to dynamic environmental stimuli.**

I. INTRODUCTION

Developments in technology make it possible to simultaneously record electrical activity from hundreds of individual neurons using chronically implanted electrodes in awake and behaving animals, including humans, for extended periods [1]–[3]. By using grids of electrodes the workings of entire neural populations can be studied neuron–by–neuron [4]–[6]. This technology has translated into novel therapies which utilize neural stimulation [7] and, for example, the neural control of robotic limbs [8], [9]. Indeed the possibility of constructing direct neural–analog interfaces is becoming a real possibility [10], [11]. However, as the number of recording electrodes grows, the limitations of our current abilities to extract useful information from the resulting large data sets become increasingly apparent.

The temporal structure of the spike train often contains more information about the stimulus than do quantities derived over the entire train [4], [12], [13]. From a neurobiological point of view these localized transients often have significance, for example, the transient synchronizations associated in processing both sensory and cognitive information [5], [14], focal high frequency EEG oscillations observed at seizure onset [15], and the abrupt changes in neural spiking patterns observed both experimentally [16]–[18] and theoretically [18],[19] which have implications for neural encoding. Consequently features that characterize temporal structure, such as spike counts, interval statistics, and synchrony, must be computed locally in bins rather than over the entire response. The portion of the spike train that corresponds to a bin depends on the feature chosen for clustering, e.g. for a spike time feature the bin length corresponds to the width of a spike, whereas it will be longer if the feature is mean frequency. Consequently if we define a *dimension* as the value of a statistic in a given bin, the number of dimensions grows large as the length of the spike train grows relative to the bin size, and as the number of potentially relevant features grows [20], i.e. dimension = number of bins × number of features. Thus analysing populations of neural spike trains falls naturally into the domain of high dimensional clustering [21].

One well–studied approach for clustering data without supervision is to use a neural network with Adaptive Resonance Theory (ART) [22], [23] (Figure 1a). ART neural networks have been shown to be very effective in self–organized data clustering problems when data cluster on all of the input dimensions. However, as the number of dimensions becomes large it becomes increasingly unlikely that data is similar on all dimensions. Consequently ART neural network perfom poorly on realistic large data sets [24] – [26]. This problem cannot be solved by performing a dimension reduction before analysis using techniques such as Principal Components Analysis (PCA) [29].

This “dimensionality curse” can be overcome by incorporating projective clustering techniques into an ART neural network; the resultant neural network is designated PART [25]–[29] (Figure 1b). Since in a PART neural network similarities between clusters are based on comparisons involving only certain subsets of the dimensions, PART neural networks are particularly well suited to handle issues related to nonstationarities. This is because we can identify different time windows with different subsets of the cluster dimensions. Here we develop a PART neural network for the purpose of identifying transients clusters within multiple neural spikes trains on the basis of changes in neurobiologically relevant statistical features such as temporally modulated firing rate, coefficient of variation and spike patterns. The resulting PART neural network provides a powerful method for clustering of neural spike trains that is easily implemented, computationally inexpensive, and well suited for identifying responses in neural spike trains related to transient environmental stimuli.

This work was supported by grants from the National Institutes of Mental Health (JM, MH-47542), the National Science Foundation (JM, NSF-0617072), the National Science and Engineering Research Council of Canada (JW), and from MITACS (JM, JW). JM acknowledges support from the William R. Kenan, Jr Foundation.

J. Hunter is with Tradelink, 200 West Jackson Blvd., Chicago, IL 60606, USA; jdh2358@gmail.com

J. Wu is with the Laboratory for Industrial and Applied Mathematics, York University, Toronto, ON M3P 1P3, Canada; wujh@mathstat.yorku.ca

J. Milton is with the Joint Science Department, The Claremont Colleges, 925 N. Mills Ave., Claremont, CA 91711, USA; jmilton@jsd.claremont.edu

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

II. ART NEURAL NETWORKS

The dynamics of the unsupervised ART (Fig. 1a) and PART (Fig. 1b) networks are governed by the equations that describe the short term (STM) and long term (LTM) memory traces together with a similarity condition [25], [26]. STM corresponds to the type of memory that can be readily reset without leaving an enduring trace. LTM represents the type of memory usually associated with learning. It is stored in the adaptive weights of both the bottom–up and top–down paths (labelled z in (Fig. 1b)). Learning takes the form of changes in these synaptic weights.

A. Short–term memory (STM) equations

Following [25], [26], the STM equations for layers F_1 and F_2 in a PART neural network (Fig. 1b) are

$$\epsilon\frac{dx_i}{dt} = -x_i + I_i \quad (1)$$

$$\epsilon\frac{dx_j}{dt} = -x_j + (1 - Ax_j)J_j^+ - (B - Cx_j)J_j^- \quad (2)$$

The F_2 equations derive from a Hodgkin-Huxley formulation with center-on, surround-off, connectivity [22]. Following the convention of [22], we use the index i to refer to the F_1 nodes and j to refer to the F_2 nodes. We take $A = 1$, $B = 0$ and $C > 0$ [25], [26]. The terms J^+ and J^- give the F_2 layer self–excitation and self–inhibition, and have their typical form

$$J_j^+ = g(x_j) + T_j \quad (3)$$

$$J_j^- = \sum_{k \neq j, k \subset F_2} g(x_k) \quad (4)$$

where g is a signal function, the notation $k \subset F_2$ means the indices of all the elements in the F_2 node, and the total input T_j to the j-th node of F_2 is given by

$$T_j = \sum_{i \subset F_1} z_{ij} h_{ij} \quad (5)$$

where z_{ij} and z_{ji} are the bottom–up and top–down synaptic weights, and h_{ij} is the selective output (see below). Specific forms of the signal function g are not important; however, the simulations in this paper are based on the usual binary function.

B. Long-term memory (LTM) equations

A node in F_2 layer is called *committed* if it has learned some input patterns in previous learning traces and *non–committed* is it has not yet learned an input pattern. The evolution of the bottom-up weights, z_{ij} is given by

$$\delta\frac{dz_{ij}}{dt} = f_2(x_j)\left[(1 - z_{ij})h_{ij}L - z_{ij}\sum_{k \neq i, k \subset F_1} h_{kj}\right] \quad (6)$$

for the committed neurons and

$$\delta\frac{dz_{ij}}{dt} = (1 - z_{ij})L - z_{ij}(m - 1) \quad (7)$$

for the uncommitted neurons where L is a constant parameter set to 2 in this paper. The top-down weights for the uncommitted neurons evolve as

$$\delta\frac{dz_{ji}}{dt} = f_2(x_j)\left[-z_{ji} + f_1(x_i)\right] \quad (8)$$

and the committed neurons as

$$\gamma\frac{dz_{ji}}{dt} = f_2(x_j)\left[-z_{ji} + f_1(x_i)\right] \quad (9)$$

Since the effect of $f_1(x)$ on the F_2 layer is mediated through the comparison layer (Fig. 1) we incorporate it into the definition of h_{ij} (see below). Under the steady state assumptions discussed in Section IV, $f_2(x)$ equals 1 if j is the winning node and is 0 otherwise. It should be noted that z_{ij}, z_{ji}, h_{ij}, and x are all time dependent variables; however, we have omitted the (t) argument in order to simplify the notation.

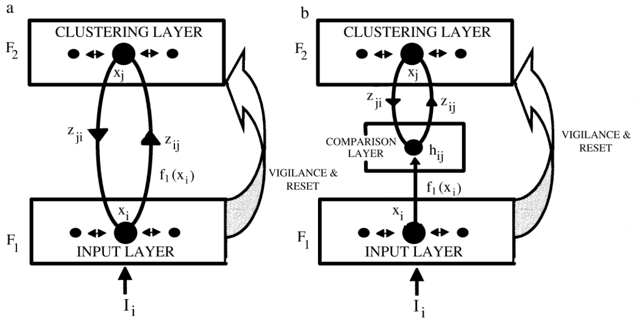

Fig. 1. Schematic of ART and PART networks. a) ART network: The total signal received by an F_2 neuron is the weighted (by bottom-up weights z_{ij}) sum of the outputs of the F_1 neurons, each of which encodes a dimension of the input space. The neuron in F_2 that is activated by a given input is chosen by a "winner–take–all" mechanism. Inputs are clustered by identifying the input with the winning F_2 node. b) The key element for the dimensionality curse proposed by [25] was the addition of a hidden layer between the F_1 and F_2 layers. This hidden layer calculates the similarity between the output of a F_1 neuron and the top-down weight z_{ji}; a signal is propagated from F_1 to F_2 only if the similarity measure is sufficiently high. Thus a given F_2 clustering node receives input only from a subset of the F_1 neurons; this subset determines the dimensions of the projective cluster.

III. PART NEURAL NETWORKS

The essential differences between a PART and an ART neural network reside in the definition of the selective output mechanism, h_{ij}, that enables outputs to be forwarded from the F_1 to the F_2 node. In a PART network, this selection process takes place in a hidden layer between the F_1 and F_2 layers [25], [26]. The hidden layer calculates the similarity between the output of a F_1 neuron and the corresponding feature of the cluster represented by a neuron in the F_2–layer: a signal is transmitted without significant loss from F_1 to F_2 only if this measure of similarity is sufficiently high. In particular we have

$$h_{ij} = h_\sigma(f_1(x_i), z_{ji})\ell_\theta(z_{ij}) \quad (10)$$

where h_σ compares the bottom–up signal $f_1(x_i)$ and the top-down template z_{ji}, and ℓ_θ is the function that insures the bottom-up weight z_{ij} is sufficiently large. The neuron in

F_2 that learns the input is chosen through a "winner–take–all" mechanism, i.e. the neuron in F_2 with the highest net input wins. This selection mechanism is coupled with a reset mechanism: a winning F_2 node will be reset so that it will always be inactive during the remainder of the current trial if it does not satisfy some vigilance conditions. In particular, a winning (active) F_2 node will be reset if at any time the degree of match, r_j,

$$r_j = \sum_i h_{ij}$$

is less than a perscribed vigilance

$$r_j < \rho \tag{11}$$

where ρ is the vigilance parameter. The vigilance condition ensures that clusters are formed in a subspace of significant dimensions and thus avoids the selection of clusters based on trivial unimportant featres. Note that this does not require clusters to be formed in the full space.

In contrast, for an ART neural network there is no vigilance layer (Fig. 1a) and the focus is to find clusters with respect to all variables of the input vector. In ART the total signal received by a F_2–neuron is the weighted sum of the outputs of the F_1–neurons. Numerical simulations demonstrate that PART algorithms significantly out-perform ART networks for clustering moderate and high dimensional data sets [25], [26].

We found that for spike time clustering it was sufficient to use the following, particularly simple choices of h_σ and ℓ_θ

$$h_\sigma = \begin{cases} 1 & \text{if } c(a,b) < \sigma \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

and

$$\ell_\theta = \begin{cases} 1 & \text{if } z_{ij} > \theta \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

where $c(a,b)$ is a measure of the closeness of $f_1(x_i)$ and z_{ji} [25], [26]. For spike train clustering we choose $c(a,b) = |a-b|$; however, any meaningful function that measures closeness can be used. Thus the hidden layer transmits a signal from an F_1 node to an F_2 node only if the top–down weight and bottom–up activity are sufficiently close, and the bottom–up weight is sufficiently large.

IV. SPIKE TRAIN CLUSTERING USING PART

In a PART architecture there are three different time scales: 1) the rate of the STM traces in layers F_1 and F_2 governed by ϵ; 2) the rate of the LTM traces for the bottom-up and uncommitted top-down weights governed by δ; and 3) the LTM trace for the top-down weights of the committed F_2 neurons governed by γ. Since STM is a form of dynamic memory, it evolves rapidly compared to the LTM. In addition we assume that the committed nodes in F_2 evolve much more slowly than the uncommitted nodes. These conditions imply $0 < \delta, \epsilon \ll \gamma$; thus on the time scale of the LTM equations for the committed F_2-nodes, the STM equations and other synaptic weight equations are approximating steady state.

PART clustering algorithm

```
1. Step 0: Initialization
    • Set the number, $m$, of nodes in $F_1$
    • Set the number of nodes, $n$ in $F_2$ to be much larger than the
      expected number of clusters.
    • Choose $\rho$ and $\sigma$.
2. Step 1: Set all $F_2$ nodes to be noncomitted
3. Step 2: For each spike train do Steps (a)-(f)
    (a) Compute $h_{ij}$ using (12) and (13) FOR all $F_1$ nodes $x_i$ and all
        committed $F_2$ nodes $x_j$. If all $F_2$ nodes are noncommitted then
        go to Step c.
    (b) Use (5) to compute $T_{ij}$ for all committed nodes in $F_2$.
    (c) Select the winning $F_2$ node $x_J$. If no $F_2$ can be selected, put the
        spike train into outlier and go to Step (f).
    (d) If the winner is a committed node, computer $r_J$, otherwise go to
        Step (f).
    (e) If $r_J \geq \rho$, go to Step (f), otherwise reset the winner $x_J$ and go
        back to Step (c).
    (f) Set the winner $x_J$ as committed, and update the bottom–up and
        top–down weights for the winning node using (14) and (15).
4. Step 3: Repeat Step 2 for each spike train.
```

Fig. 2. Algorithm for PART spike train clustering. A detailed numerical example illutrating this alogorithm in given in [26].

Taking the steady state approximation of Eq 1 and the signal function f_1 to be identity, we have $x_i = I_i$. As mentioned previously, $f_2(x_j)$ equals 1 if j is the winning node and 0 otherwise. Our choices of ℓ_θ and h_σ require that h_{ij} is 1 if F_1-node i is active to the F_2-node and is 0 otherwise. The steady state approximations for the weight updates for an input F_1-node i and winning F_2-node j are thus

$$z^1_{ji} \quad = \quad \begin{cases} (1-\alpha) z^0_{ji} + \alpha I_i & \text{committed} \\ I_i & \text{noncommitted} \end{cases} \tag{14}$$

and

$$z^1_{ij} \quad = \quad \begin{cases} L/(L + |X| - 1) & \text{committed} \\ 0 & \text{noncommitted} \end{cases} \tag{15}$$

where the superscripts indicate the values at time steps 1 and 0 and $|X|$ is the number of nodes i active to j, as in [25], [26]. The expression for updating the z_{ji} of committed nodes in Eq. 14 is obtained by directly integrating Eq. 9 over the unit interval assuming constant I_i and h_{ij} over that interval, and setting the learning rate $\alpha = 1 - e^{-1/\gamma}$. The other expressions in Eqs 14 and 15 are steady state approximations of their respective differential equations, since the bottom-up weights and top-down weights of noncommitted nodes all evolve rapidly compared to the synaptic weights of the committed nodes. Unless otherwise noted, $\alpha = 0.1$, $\theta = 0$.

The algorithm for clustering spike trains is shown in Fig. 2. A cluster corresponds to those spike trains that led to the same winning node in F_2. Using these steady state approximations it can be shown analytically that the PART neural network defined by (11)–(15) exhibits a regular computational performance characterized by [26]

- Selected output clusters remain constant,
- Winner–take–all paradigm: the F_2 node with the largest bottom–up filter input becomes the winner and only this node is activated after some finite time.
- The set of dimensions of a specific projected cluster is non–increasing in time.

V. APPLICATION TO SPIKE TRAINS

Equations (11)–(15) define an iterative procedure for clustering neural networks using a PART neural network that doesn't require explicit integration of the differential equations. Under the steady state approximation there are only two free parameters to be specified in order to implement a PART neural network, namely, ρ and σ. Spike train inputs for the PART neural network have the general form

$$\left(\overbrace{s_{11}, s_{12}, \cdots, s_{1p}}^{bin_1}, \overbrace{s_{21}, s_{22}, \cdots, s_{2p}}^{bin_2}, \cdots, \overbrace{s_{k1}, s_{k2}, \cdots, s_{kp}}^{bin_k}\right)$$

where the dimension, m, is equal to the number of bins, bin_k, of size Δt times the number of statistical features of interest, and the notation s_{kp} denotes the p–th statistical feature evaluated for the k–th bin. The number of nodes in F_1 is m and the number of nodes, n, in F_2 is much greater than the expected number of clusters. At onset all of the F_2 nodes are non–committed. The single fixed point of this iterative procedure corresponds to a committed node in F_2 which, in turn, corresponds to a cluster. A detailed numerical example of this clustering process is given in [26]. Briefly, h_{ij} is computed using (12)–(13) and the winning F_2 is selected. If this node passes the vigilance criterion, (11), then the top–down and bottom–up weights, respectively, z_{ji}, z_{ij}, are updated accorrding to (14)–(15), h_{ij} re–computed, and so on. Typically the single committed node in F_2 is determined within just a few iterations of this procedure. Once the committed F_2 node has been determined, the next spike train is presented. All spike trains that belong to the same committed F_2 node belong to the same cluster.

The number of input patterns that can be learned by a PART neural network is limited only by the finiteness of the number and length of spike trains that can be presented to it. There are a number of consequences for the practical application of PART neural networks: 1) it is better to cluster data sets with respect to a few, e.g. one, statistical features at a time; 2) the order of presentation of spike trains may have an influence of the clustering results; 3) the number of nodes in F_2 must be larger than the number of suspected clusters, i.e. $n \gg m$; and 4) there will always be a small number of spike trains which do not cluster well: following [26] we placed all such data into an *outlier node*.

The PART clustering algorithm described by (11)–(15) was validated on populations of neural spike trains constructed using two types of model neurons: 1) the leaky integrate and fire (LIF) model [30]–[32], and 2) a reduced Hodgkin–Huxley model neuron [37], [38]. The goal in constructing these data sets was to pose a difficult clustering problem consistent with the known physiological responses of neurons. Validation using this procedure is facilitated by the fact that the natures and numbers of the true clusters are known. It was observed that in all cases this PART neural network correctly classified the clusters when $sigma$ was between 0.1 and 0.2 (data not shown). This was true whether the spike train dimensions were chosen to be related to instantaneous firing rate clusters or to synchronization.

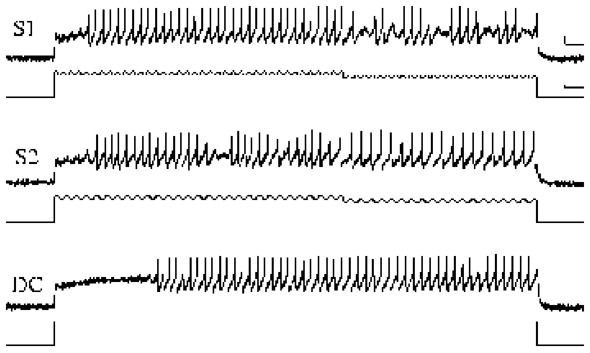

Fig. 3. Stimuli presented to *Aplysia* motoneuron: S1) A sinusoidal input at 12.5 Hz plus a DC current at 13 nA plus a 1 nA hyperpolarizing current with onset at 3.5 s; S2) As in S1 except the sinusoidal frequency was 9.5 Hz; DC) The DC current only. The current inputs are shown below each of the three voltage responses. The scale bars in the upper right are 0.5 s by 10 mV and 0.5 s by 2 nA and apply to all three conditions.

VI. APPLICATION TO INVERTEBRATE NEURONS

Aplysia motoneurons exhibit dynamic changes in synchronization to periodic or aperiodic inputs in response to small changes in their firing rate [33], [34]. We used this as a model preparation to test the ability of the PART algorithm to cluster these transient behaviors in living neurons.

Aplysia care and dissection were performed as described in [39]. Recordings from identified neurons in the buccal motor ganglion were made in two electrode current clamp mode using an AxoClamp 2B amplifier (Axon Instruments, Foster City, CA) with electrode resistances $\approx 4\,\text{M}\Omega$. Data were low pass filtered at 1 kHz with a Frequency Devices 902 low pass filter (Frequency Devices, Haverhill, MA) and digitally sampled at 2 kHz [33]– [35]. Sampling at twice the corner frequency rather than twice the stop frequency of the filter does allow some frequency aliasing to occur, the amouint will be quite small since the corner attenuation is 3 dB. These filtering choices might be a problem if the object is analayze submembarne fluctuations; however, it is neglible when the object is spikle detection. All recording and stimulation protocols were automated using an AD2210 A/D board (Real Time Devices, State College, PA) interfaced with a personal computer.

We used spike trains recorded from *Aplysia* motoneurons to test the performance of the PART algorithm for identifying transient responses in living neurons in response to periodic inputs [33], [34]. Three experimental conditions were chosen (Fig. 3). In the first two conditions, $S1$ and $S2$, the motoneuron was presented with sinusoidal and direct currents in order to induce phase-locking [34] in only one part of the response - first half ($S1$) or second half ($S2$). The third condition (DC) was designed to create an outlier set. This was accomplished by injecting a direct current which is not expected to generate phase locking and thus not form a synchrony cluster [33], [34], [40].

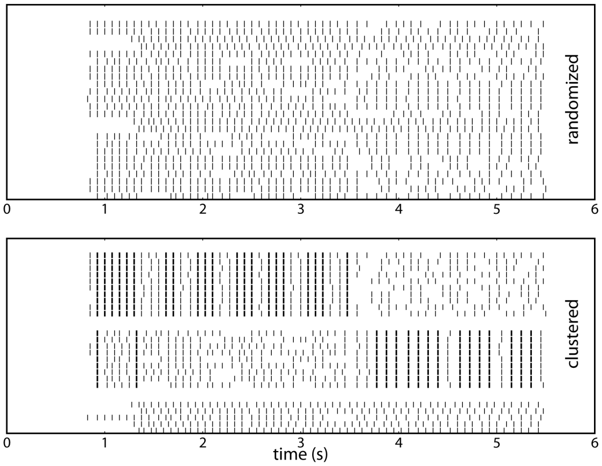

Fig. 4. *Clustering spike trains from Aplysia buccal ganglion.* A) Raster response of repeated presentations of the three stimuli from Figure 3 presented in randomized order. B) The trains from A) were presented to the PART clustering algorithm with $\rho = 15$ and $\sigma = 0$ (synchrony detection) which sorted the trains two clusters and one outlier cluster; bold spikes indicate the dimensions of the projected subspace. These three clusters correspond closely to the three stimulus conditions (see Table I).

The stimuli were repeatedly presented to the neuron in random order and the spike train responses are shown in the upper raster plot of Figure 4. The spike trains were presented to the PART algorithm tuned to detect synchronous spiking, as in the model data above. The algorithm identified 2 clusters and one outlier cluster, which are shown in the lower rasters of Figure 4. The algorithm correctly identifies the different input conditions as clusters, as well as the sub regions of the spike trains where synchronous spiking occurs for groups S1 and S2; the detected synchronous spikes are colored blue. The contingency table shown in Table I shows that 23 of 24 spike of the spike trains were correctly classified on the basis of their inputs.

TABLE I

CONTINGENCY TABLE FOR THE SYNCHRONY CLUSTERS IN *Aplysia* DATA SHOWN IN FIG. 4.

	S_1	S_2	DC	Total
S_1	10	0	0	10
S_2	0	9	0	9
OUT	0	1	4	5
Total	10	10	4	24

VII. DISCUSSION

Neural responses are often transient. Thus it is likely that spike trains cluster on only a subset of the time bins (dimensions); for example, if a subpopulation responds to only part of a stimulus, it will cluster only over the time bins in which it responds. The strength of the PART neural network algorithm for clustering spike trains is that it elegantly handles nonstationarities of this type. If spike trains are similar only over a subset of the time bins, PART will identify both the clusters and the time bins over which the trains are similar. A different cluster can be similar over a different subset of the bins. Thus the algorithm is readily capable of identifying one cluster that has similar rate modulation over the first half of a stimulus, and another cluster that exhibits a transient synchronization over the second half.

We have shown that a PART network is a powerful tool to sort large numbers of neural spike trains that share common features. This point is illustrated in Figure 5a, in which two distinct clusters of data points are scatter plotted as gray or black over three arbitrary dimensions. On the other hand, ART networks identify three clusters in the full space (Figure 5b). The false third cluster (light gray) arises because points on it are close to one another in the full dimensional spaces. It should be emphasized that this problem cannot be solved by doing a dimension reduction before analysis, using techniques such as Principal Components Analysis (PCA), because all of the dimensions are required to represent these two clusters and PCA generates poor results in data that have multiple clusters [42].

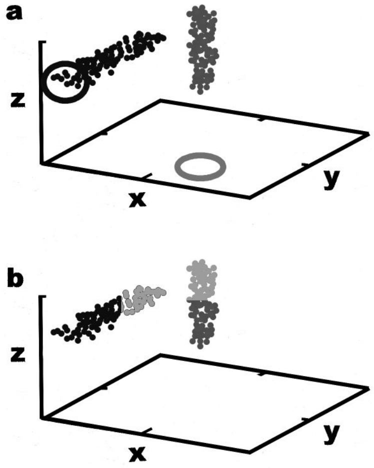

Fig. 5. a) Scatter plots of two clusters (gray dots and black dots) in three dimensional space with arbitrary dimensions x, y, and z. The gray dots cluster on dimensions x and y, and the black dots on x and z. The cluster outlines in the project subspace are drawn as gray and black ellipses on the respective planes. PART identifided 2 clusters and the dimensions of the subspaces. b) ART identifies clusters in the full space, and so finds a false third cluster (light gray) the encloses the dots from both the black and gray clusters that are close to one another in the full space.

The advantage of this unsupervised clustering method is that the features which delineate a group are discovered and learned by the network automatically. Even in its simple form used here, the PART network is highly effective for clustering neural spike trains. Since the update equations are reduced to an iterative map rather than numerically integrated, it is computationally feasible to rapidly cluster large numbers of spike trains over a large number of dimensions even with modest computer hardware. Inclusion of sigmoidal signal functions, rather than the identity and hard threshold functions used here, might further improve the performance of the network, as might numerically integrating the full system of differential equations. However, the network in the reduced state is quite powerful, and is amenable to analytic treatment [25], [26]. Other algorithms have been proposed for cluster analysis of neural spike trains [22], [23], including the recently emphasized methods for the identification of spiking patterns within neural spike trains based on the K–means algorithm [17], [18]. Since all these methods require that data cluster on all of the dimensions, they suffer the dimensionality curse as the dimensions become larger.

The use of our PART algorithm is not limited to clustering neural spike trains. In principle any signal can serve as input. Thus it may be possible to use PART networks for spike sorting or classification of EMG and EEG signals. Of course, in these applications appropriate features to encode the inputs need to be identified; for example, parameters related to the waveform for spike sorting applications, parameters related to frequency content for EMG and EEG, and so on. A major advantage of techniques based on PART networks is that they minimize the effects of the variability of human performance on the outcomes of the analysis.

REFERENCES

[1] J. L. Novak and B. C. Wheeler, *J. Neurosci. Meth.*, vol. 23, pp. 149–159, 1988.

[2] C. T. Nordhausen, P. J. Rousche, and R. A. Normann, *Brain Res.*, vol. 637, pp. 27–36, 1994.

[3] T. A. Fofonoff, S. M. Martel, N. G. Hatsopoulos, J. P. Donoghue, and I. W. Hunter, *I*EEE Trans. Biomed. Engng., vol. 51, pp. 890–895, 2004.

[4] B. J. Richmond and L. M. Optican, *J. Neurophysiol.*, vol. 64, pp. 370–380, 1990.

[5] K. D. Harris, J. Csicsvari, H. Hirase, G. Dragoi, and G. Buzsáki, *Nature (London)*, vol. 424, pp. 552–556, 2003.

[6] L. Paninski, S. Shoham, M. R. Fellows, N. G. Hatsopoulos, and J. R. Donoghue, *J. Neurosci.*, vol. 24, pp. 8551–8561, 2004.

[7] J. J. Pancrazio, D. Chen, S. J. Fertig, R. L. Miller, E. Oliver, G. C. Y. Peng, N. L. Shinowara, M. Weinrich, and N. Kleitman,*Neuromodulation*, vol. 9, pp. 1–7, 2006.

[8] J. K. Chapin, K. A. Moxon, R. S. Markowitz, and M. A. L. Nicolelis, *Nature Neurosci.*, vol. 2, pp. 664–670, 1999.

[9] S. Shoham, L. M. Paniski, M. R. Fellows, N. G. Hatsopoulos, J. P. Donoghue, and R. A. Normann, *IEEE Trans. Biomed. Engng.*, vol. 52, pp. 1312–1322, 2005.

[10] P. Fromherz and A. Stett, *Phys. Rev. Lett.*, vol. 75, pp. 1670–1673, 1995.

[11] D. W. Branch, B. C. Wheeler, G. J. Brewer, and D. E. Leckband, *IEEE Trans. Biomed. Engng.*, vol. 47, pp. 290–300, 2000.

[12] J. Victor and K. Purpura, *J. Neurophysiol.*, vol. 80, pp. 554–571, 1998.

[13] D. Reich, F. Mechler, and J. Victor, *J. Neurophysiol.*, vol. 85, pp. 1039–1050, 2001.

[14] W. Singer, *Ann. Rev. Physiol.*, vol. 18, pp. 349–374, 1993.

[15] D. Jirsch, E. Urrestarasu, P. Levan, A. Olivier, F. Dubeau, and J. Gotman, *Brain*, vol. 129, pp. 1593–1608, 2006.

[16] J. C. Middlebrooks, A. E. Clock, L. Xu, and D. M. Green, *Science*, vol. 264, pp. 842-844, 1994.

[17] J.M. Fellous, P. H. E. Tiesinga, P. J. Thomas, and T. J. Sejnowski, *J. Neurosci*, vol. 24, pp. 2989–3001, 2004.

[18] J. V. Toups and P. H. E. Tiesinga, *Neurocomputing*, vol. 69, pp. 1362–1365, 2006.

[19] J. Foss, F. Moss, and J. Milton, *Phys. Rev. E*, vol. 55, pp. 4536–4543, 1997.

[20] R. Segev, M. Benveniste, E. Hulata, A. Palevski, E. Kapon, Y Shapira and E. Ben–Jacob, *Phys. Rev. Lett.*, vol. 88, 118102, 2002.

[21] A. Jain, M. Murty and P. Flynn, *ACM Computing Reviews*, vol. 31, pp. 264–323, 1999.

[22] G. A. Carpenter and S. Grossberg, *Comp. Vis. Graphics Image Proc.*, vol. 37, pp. 54–115, 1987.

[23] G. A. Carpenter and S. Grossberg, *Neural Networks*, vol. 4, pp. 759–771, 1991.

[24] A. Hinneburg, C. C. Aggarwal, and D. A. Kein, *Proc. 26th Intl. Conf. Very Large Data Bases*, pp. 506–515, 2000.

[25] Y. Cao and J. Wu, *Neural Networks*, vol. 15, pp. 105–120, 2002.

[26] Y. Cao and J. Wu, *IEEE Trans. Neural Networks*, vol. 15, pp. 245–260, 2004.

[27] C. C. Aggarwal, C. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park, *SIGMOD '99*, pp. 61–72, 1999.

[28] C. C. Aggarwal and P. S. Yu, *SIGMOD '00*, pp. 70-81, 2000.

[29] C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. M. Murali, *SIGMOD '02*, pp. 418–427, 2002.

[30] A. Rescigno, R. B. Stein, R. L. Purple, and R. E. Poppele, *Bull. Math. Biophys.*, vol. 32, pp. 337–353, 1970.

[31] B. K. Knight, *J. Gen. Physiol.*, vol. 59, pp. 734–766, 1972.

[32] P. H. E. Tiesinga, *Phys. Rev. E*, vol. 65, 041913, 2002.

[33] J. D. Hunter, J. G. Milton, P. J. Thomas, and J. D. Cowan, *J. Neurophysiol.*, vol. 80, pp. 1427–1238, 1998.

[34] J. D. Hunter and J. G. Milton, *J. Neurophysiol.*, vol. 90, pp. 387–394, 2003.

[35] J. D. Hunter and J. G. Milton, *J. Neurosci.*, vol. 21, pp. 5781–5793, 2001.

[36] U. Beierholm, C. D. Nielsen, J. Ryge, P. Atsrm, and O. Kiehn, *J. Neurophysiol.*, vol. 86, pp. 1858–1868, 2001.

[37] E. M. Izhikevich, *IEEE Trans. Neural Networks*, vol. 14, pp. 1569–1572, 2003.

[38] E. M. Izhikevich, *IEEE Trans. Neural Networks*, vol. 15, pp. 1063–1070, 2004.

[39] P. J. Church and P. E. Lloyd, *J. Neurophyiol.*, vol. 77, pp. 1794–1809, 1994.

[40] Z. F. Mainen and T. J. Sejnowski, *Science*, vol. 268, pp. 1503–1506, 1995.

[41] M. Steriade, *Neuronal Substrates of Sleep and Epilepsy*. Cambridge: Cambridge University Press, 2003.

[42] C. M. Procopiuc, M. Jones, P. K. Agarwal and T. M. Murali, *SIGMOD '02*, pp. 418–427, 2002.