Proceedings of the 26th Annual International Conference of the IEEE EMBS
San Francisco, CA, USA • September 1-5, 2004

Automatic Spike Sorting for Neural Decoding

F. Wood* M. Fellows† J. P. Donoghue† M. J. Black*
*Department of Computer Science, Brown University, Providence, RI, USA
†Department of Neuroscience, Brown University, Providence, RI, USA

***Abstract*— While various automated spike sorting techniques have been developed, their impact on neural decoding has not been investigated. In this paper we extend previous Gaussian mixture models and Expectation Maximization (EM) techniques for automatic spike sorting [1]. We suggest that good initialization of EM is critical and can be achieved via spectral clustering. To account for noise we extend the mixture model to include a uniform outlier process. Automatically determining the number of neurons recorded per electrode is a challenging problem which we solve using a greedy optimization algorithm that selects models with different numbers of neurons according to their decoding accuracy. We focus on data recorded from motor cortex and evaluate performance with respect to the decoding of hand kinematics from firing rates. We found that spike trains obtained by our automated technique result in more accurate neural decoding than those obtained by human experts.**

***Index Terms*— Spike sorting, spectral clustering, expectation maximization, mixture models, motor cortex, neural decoding, neural prosthesis.**

I. INTRODUCTION

Traditional neuroscience involves the study of small populations of cells where neural activity is recorded and manually processed for ‘off-line’ analysis. Neural prosthesis applications may involve simultaneously recording from hundreds of electrodes and decoding neural activity ‘on-line’ [2]–[5]. These constraints make hand sorting of neural data impractical and highlight the need for automated methods. Traditionally the goal of spike sorting is to identify the activity of individual cells for later analysis. For neural prostheses the goal is somewhat different and we seek an automated method that optimizes decoding accuracy for neural control tasks. We develop an automatic spike sorting algorithm that produces decoding results that are as good or better than those obtained by the best human spike sorter. To evaluate decoding accuracy we use a Kalman filter to infer hand motion from neural recordings made in the motor cortex of a monkey performing a cursor control task [3].

Decoding algorithms typically exploit some function relating the firing rate of a population of cells to kinematic data such as hand position. Firing rates can be determined by counting the number of spikes a neuron produces in some time window (or bin), but detecting and assigning spikes to neurons is difficult to do. This is especially true when recording with the chronically implantable, relatively low-impedance multi-electrode arrays [6] desirable for use in neuroprosthetic applications. Given detected waveforms from such an array the problem of spike sorting involves determining which waveforms were noise, how many neurons were recorded, and which neuron each non-noise waveform came from. It has been shown that knowledgeable domain experts often disagree about not only what is and is not a spike but also about the number of neurons present in recordings for a given electrode [1], [7]. In this study we found similar variability between experts. Note that here we do not address spike detection as the waveforms we sorted were captured after crossing experimentally determined thresholds.

A number of general spike sorting techniques have previously been investigated, see [1] for a review. These techniques include template matching and k-means or Gaussian mixture model clustering of waveform principle components. Neural networks [8] and t-distribution mixture model clustering [9] have also been developed. None of the methods above adequately addressed either of the two problems that drove our research, namely, automatically determining the number of units recorded and identifying noise.

We propose an automatic spike sorting algorithm that is an extension of the Gaussian mixture model Expectation Maximization (EM) technique [1]. Our new algorithm initializes the mixture model distribution parameters by clustering a subset of the data using spectral clustering [10]. Our algorithm performs model selection by greedily optimizing decoding error. For each channel of a recording, the algorithm decodes using multiple models of varying numbers of densities. The model that produces the best decoding results is kept. We introduce decoding accuracy as a new objective means of evaluating spike sorting performance, and with it demonstrate that our algorithm performs as well as or better than expert human sorters.

II. METHODS

We studied the effect of spike sorting on decoding accuracy in the following manner. A single dataset was recorded then sorted by 4 human volunteers, an automated algorithm of our design, and two ‘control’ algorithms, resulting in 7 different spike trains. We evaluated these spike trains by decoding using the Kalman filter method of [11]. The Kalman filter was trained on 3 minutes of rate and kinematic data binned in 70ms time bins, and average decoding results are given for 5 independent 1 minute data segments. The kinematics were encoded as a 6 attribute vector comprised of x, y hand position, velocity, and acceleration. Our implementation differed from [11] in

This work was supported by supported in part by NIH-NINDS #NS25074, NIH-NINDS #N01-NS-2-2345, NSF ITR #0113679

Correspondence F. Wood (fwood@cs.brown.edu) that we found that a time lag of 0ms instead of 140ms was optimal for our data.

A. Recording

In a single monkey, following task training, a Bionic Technologies LLC (BTL) 100-electrode silicon array [6] was implanted in the arm area of primary motor cortex (MI). The recording setup was similar to that used in [3] where an animal was trained to use a two-joint planar manipulandum to control the motion of a feedback cursor on a computer screen. The simultaneous recording of hand kinematics and neural activity allowed the study of motor cortical encoding of hand motion [12] and the training of decoding methods [3], [11], [13]. A recording from this animal performing a “pinball” tracking task [11] was used in this study.

The recording used for this study was a collection of 96 independent channels where the waveform capture threshold for each channel was empirically determined at the time of recording, and was set low enough to ensure that a majority of the spiking activity was captured. The array was placed in MI, but by virtue of array design and insertion technique the location of each electrode with respect to individual neurons was unknown. Correspondingly, every channel may have had waveforms from multiple neurons. Additionally, on each channel some number of non-spike, threshold-crossing waveforms were recorded. We call these non-spike waveforms “noise”. A single contiguous 600 second segment of data was extracted from a part of the recording featuring high arm movement and neural firing rates. Unlike other studies where decoding is done only on data recorded between start and stop cues, we extracted a single continuous segment where there were periods of no arm movement. Of this 600 second segment, different independent sub-segments were used for greedy model selection, to train the Kalman filter, and to test decoding accuracy.

B. Automatic Sorting

Without loss of generality, the exposition below describes sorting data on a single channel (electrode). The same process applies to all channels. To automatically sort each channel we first reduced the dimensionality of the waveforms via principle component analysis (PCA). We then used expectation maximization (EM) [15] to fit a mixture of Gaussians (mixture model) to the waveform PCA coefficients. The mixture model cluster means, covariances, and membership weights were initialized by applying a spectral clustering algorithm to a subset of the data. A uniform probability noise process was added to the EM framework and was used to identify and eliminate noise.

Let $C = [\vec{\omega}^1, \ldots, \vec{\omega}^N]$ represent the N waveforms on a single channel, where each waveform is a vector of n voltage samples $\vec{\omega}^i = [\omega_1^i, \ldots, \omega_n^i]^T \in \mathbb{R}^n$. In our experiments $n = 40$.

The $\vec{\omega}^i$ can be approximated by a linear combination of PCA bases as $\vec{\omega}^i \approx \sum_{d=1}^{D} c_d U_d$ where we take $D = 3$ bases corresponding to approximately 70% of the variance in the data, U_d is the d^{th} PCA basis vector, and c_d are the linear coefficients. Let $\vec{c}^i = [c_1^i \ldots c_D^i]$, then the probability $P(\vec{\omega}^i|j)$ can be approximated by $P(\vec{c}^i|j)$ [14].

Assume that all the waveforms in C were generated by exactly M neurons. Then the probability, under our Gaussian assumption, that $\vec{c}^i$ was generated by neuron j, $1 < j < M$ is given by:

$$P(\vec{c}^i|j) = \frac{1}{(2\pi)^{\frac{d}{2}}\sqrt{det(\Sigma_j)}} exp(-\frac{1}{2}(\vec{c}^i - \mu_j)^T \Sigma_j^{-1}(\vec{c}^i - \mu_j))$$

where μ_j and Σ_j are the mean and covariance of the PCA coefficients of the waveforms generated by neuron j.

The probability of the channel according to the mixture model is:

$$P(C) = \prod_{i=1}^{N} \sum_{j=1}^{M} \alpha_j P(\vec{c}^i|j)$$

where the $\alpha_j > 0$ are mixing coefficients and $\sum_{j=1}^{M} \alpha_j = 1$.

The EM algorithm was used to maximize $P(C)$ through optimization of distribution parameters and membership weights (the set of which is usually represented by Θ). The EM algorithm alternates between computing the cluster membership weights $P(j|\vec{c}^i)$ and maximizing the likelihood of the data given these weights. To account for noise we introduced a uniform outlier process with a fixed likelihood $\nu = P(\vec{c}^i|noise)$. This outlier process is included in the membership weight calculations in the following manner:

$$P(j|\vec{c}^i) = \frac{P(\vec{c}^i|j)}{\sum_{l=i}^{M} P(\vec{c}^i|l) + \nu}.$$

When initializing EM there is always the issue of how to initialize Θ, given that different values of Θ can produce remarkably different clusterings due to local maxima in the log likelihood function. Further, in the case of clustering spikes we found that simple k-means or random initializations often led to solutions that were non-intuitive. To combat this we initialized Θ using spectral clustering [10]. The application of the algorithm to this problem is straightforward with two exceptions. The algorithm spectrally decomposes an ‘affinity matrix’, A, defined by $A_{ij} = \exp(-||\vec{c}^i - \vec{c}^j||^2/2\sigma^2)$ if $i \neq j$ and $A_{ii} = 0$ to find groups of waveforms that are ‘closely bunched’. The first exception is the free affinity parameter σ which we set empirically. The second exception is the fact the size of the affinity matrix grows as the square of the number of waveforms. On most channels there were too many waveforms to create an affinity matrix between all of them, so a random subset of 1000 waveforms was selected and used. The resulting cluster assignments were used to initialize the mixture means, covariances, and membership weights.

Model selection is an issue in mixture model clustering and corresponds to determining the number of clusters in the data; in this context determining the number of units on a channel. To accomplish this we employed the greedy

Algorithm 1 Greedy Sorter

```
for all channels $C$ do
    $\mathcal{R}[C] \leftarrow$ sort($C, 1$) // Assume only 1 neuron present.
end for
$error \leftarrow$ kalmanfilter($\mathcal{R}$) // Get the Kalman filter error.
for all channels $C$ do
    for $j = 0, \ldots, 5$ do
        $C_{temp} \leftarrow \mathcal{R}[C]$
        $\mathcal{R}[C] \leftarrow$ sort($C, j$) // Assume $j$ neurons present.
        $newerror \leftarrow$ kalmanfilter($\mathcal{R}$)
        if $newerror < error$ then
            $error \leftarrow newerror$
        else
            $\mathcal{R}[C] \leftarrow C_{temp}$
        end if
    end for
end for
```

approach outlined in Algorithm 1 where the subroutine sort(C, j), runs our extended EM mixture model sorting procedure on channel C assuming j mixture components. The algorithm iterates through all the channels of a recording, sorting each channel potentially many times, assumes a different number of mixture components each time, and chooses the number of mixtures that maximizes decoding accuracy. Note that the algorithm can pick a zero component mixture model, which effectively corresponds to excluding channels that do not improve decoding.

C. Human and Control Sorting

In [7] we found that expert human sorters produced different spike trains given the same recording, however we did not study how this effected decoding results. By asking volunteers to sort the same dataset we were able to both ascertain how subjective variability in spike trains effects decoding results and to establish a baseline against which to judge our algorithm.

Four subjects, all graduate students or postdoctoral researchers and all expert spike sorters, were given the recording and asked to sort it using any tool at their disposal. They were instructed to sort it in the way they thought would maximize decoding performance. All subjects used Plexon's Offline sorter software [16] to sort the dataset. Units were identified and spikes were assigned to them by manually cluster cutting waveforms projected into a 2 dimensional PCA space. The resulting spike trains were decoded and the results are displayed as A,B,C, and D in Table I.

To better understand how both our volunteers and our algorithm performed we also compared the results to spike trains produced by both randomly sorting or not sorting at all. No sorting ('None' in Table I) means that all waveforms on each channel were attributed to a single neuron. Randomly sorting ('Random' in Table I) means that on each channel three neurons were posited and waveforms were randomly attributed to them with equal probability.

D. Rate Estimation

A common approach to neural decoding exploits a relationship between the firing rates of individual neurons and some stimulus or motor variable (hand kinematics in our case). It is common to 'bin' spikes within fixed time windows to produce an estimate of firing rate for use in decoding.

We chose to examine decoding performance given two different estimates of the firing rate. One is the usual maximum likelihood clustering interpretation, the results for which are given as 'Auto Max' in Table I. The second, 'Auto Weighted' in the same table, attempts to account for some of the spike train ambiguities in a probabilistically sound way.

Let $B = \sum_{i=1}^{C} M_i$ be the number of cells identified on all channels where M_i is the optimal number of units identified for each channel. Let $\mathbf{z}_k = [z_k^1, z_k^2, \ldots, z_k^B]$ be the firing rate of all B cells identified in the recording in time bin k where z_k^j is the firing rate of cell j. Let $\vec{c}^{\,i}, 0 < i < K$ be the PCA representation of the K waveforms recorded between the start and end times of bin k.

Then, under the mixture model, the maximally likely firing rate, 'Auto Max', is:

$$z_k^j = \sum_{i=0}^{K} \begin{cases} 1 \text{ if } j = \underset{m}{\text{argmax}} P(m|\vec{c}^{\,i}); \\ 0 \text{ otherwise.} \end{cases}$$

A motivation for our adoption of a probabilistic, mixture model sorting algorithm was that we could leverage the characterization of uncertainty in our decoding algorithm. The 'Auto Weighted' decoding used the same Kalman decoding framework but the firing rate was computed as:

$$z_k^j = \sum_{i=0}^{K} P(j|\vec{c}^{\,i})$$

.

In both estimates of firing rate $P(j|\vec{c}^{\,i}) = 0$ if $\vec{c}^{\,i}$ was not recorded on the same channel where neuron j was identified.

This weighting scheme accounts for spikes that are difficult to attribute to any single neuron. It does so by contributing a 'partial spike' or 'weight' to the firing rates of every neuron that could have generated them. In the clustering context, this is equivalent to eschewing a maximum likelihood clustering criteria, and instead finding a way to cope with the fact that a point may be inherently ambiguous and should be treated is if it belongs to several clusters simultaneously. In this 'auto weighted' rate estimation, we used the mixture likelihood given a spike $P(j|\vec{c}^{\,i})$ as the 'partial spike' weight contribution to neuron j.

III. RESULTS

Table I summarizes the decoding results. In the table, 'None' posits one noiseless unit per channel. 'Random' posits three units per channel and assigns waveforms to each unit uniformly at random. 'Ave. Human' is the average of subjects 'A', 'B', 'C', and 'D'. 'Auto Max' is the result for the algorithm described in this paper with the normal, maximum likelihood estimate of firing rate. 'Auto Weighted' uses the same clustering as 'Auto Max' except that a probabilistically 'weighted' estimate of firing rate is used instead.

TABLE I

DECODING RESULTS

Subject	Neurons	Spikes	MSE ± std. (cm^2)
A	107	757674	11.45 ± 1.39
B	96	335656	16.16 ± 2.38
C	78	456221	13.37 ± 1.52
D	88	642422	12.37 ± 1.22
Ave. Human	92	547993	13.46 ± 2.54
Random	288	860261	13.28 ± 1.54
None	96	860261	12.78 ± 1.89
Auto Max	114	625861	11.31 ± 1.33
Auto Weighted	114	625861	11.30 ± 1.15

'None' and 'Random' are shown for comparison. Reported are the total number of neurons and spikes identified and the mean square error between the true and decoded x, y hand position. We found that our automatic sorter did as well or better than the best human sorter.

IV. DISCUSSION

We found it surprising that human spike sorters produced spike trains that are worse for decoding than no sorting at all. With the exception of subject 'A', the subjects generally erred toward missing valid spikes in lieu of attributing non-spikes to neurons. This sorting strategy might result in problematically low firing rates for decoding.

For decoding it might be more important to find units that are well correlated to the kinematic variables than to find precisely the unique units being recorded. This differs substantially from the objectives of spike sorting for neuroscience at large, and resulted in some unusual but effective clusterings. Consider a channel containing clearly separable clusters of waveforms from two different neurons. If they have similar tuning properties they may be grouped together and treated as a single unit.

Additionally, it has been observed [17] that 'discarding' certain units sometimes improves decoding results. Our algorithm discarded some units, and indeed some channels wholly, by labeling all activity on poorly correlated channels to be noise. The sequential greedy nature of our algorithm may remove more units from consideration than is necessary. It is possible that the observed improvement in decoding results may have come from the elimination of superfluous data in this way.

We stress that the particular algorithm given in this paper is a 'proof of concept' approach and that both doing a better job of accounting for spike train uncertainty in the decoding process and building a better formulation of spike sorting as a decoding optimization problem may ultimately improve neural decoding significantly.

V. CONCLUSIONS

We discovered that spike train variability affects decoding results and demonstrated that for neuroprosthetic applications simple automated sorting techniques are as good or better than human manual sorting. We extended Gaussian mixture model sorting to identify non-spike waveforms as noise and exploited a spectral cluster initialization technique that works well in this context. We also demonstrated that decoding performance can be used to solve the problem of model selection for neural decoding applications. Likewise we suggested that decoding results may be a valuable tool to evaluate sorting algorithms. Areas for future research include developing a better noise model and evaluating basis transformations other than PCA.

ACKNOWLEDGEMENT

We wish to thank D. Morris of Stanford University for his well organized software contribution. We also wish to thank C. Vargas-Irwin and W. Truccolo for serving as expert sorters.

REFERENCES

[1] M. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network*, vol. 9, no. 4, pp. R53–78, 1998.

[2] S. Helms Tillery, D. Taylor, R. Isaacs, and A. Schwartz, "Online control of a prosthetic arm from motor cortical signals," *Society for Neuroscience Abst.*, vol. 26, 2000.

[3] M. Serruya, N. Hatsopoulos, L. Paninski, M. Fellows, and J. Donoghue, "Brain-machine interface: Instant neural control of a movement signal," *Nature*, vol. 416, pp. 141–142, 2002.

[4] D. Taylor, S. Tillery, and A. Schwartz, "Direct cortical control of 3D neuroprosthetic devices," *Science*, vol. 296, no. 5574, pp. 1829–1832, 2002.

[5] J. Wessberg, C. Stambaugh, J. Kralik, P. Beck, M. Laubach, J. Chapin, J. Kim, S. Biggs, M. Srinivasan, and M. Nicolelis, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, pp. 361–365, 2000.

[6] E. Maynard, C. Nordhausen, and R. Normann, "The utah intracortical electrode array: A recording structure for potential brain-computer interfaces," *Electroencephalography and Clinical Neuophysiology*, vol. 102, pp. 228–239, 1997.

[7] F. Wood, M. Fellows, I. Vargas-Irwin, M. Black, and J. Donoghue, "On the variability of manual spike sorting," *IEEE TBME*, to appear June 2004,

[8] K. Kim and S. Kim, "Neural spike sorting under nearly 0-db signal-to-noise ratio using nonlinear energy operator and artificial neural-network classifier," *IEEE TBME*, vol. 47, pp. 1406–1411, 2000.

[9] S. Shoham, M. Fellows, and R. Normann, "Robust, automatic spike sorting using mixtures of multivariate t-distributions," *J. Neuro. Methods*, 127(2):111–122, 2003.

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

[11] W. Wu, M. Black, Y. Gao, E. Bienenstock, M. Serruya, A. Shaikhouni, and J. Donoghue, "Neural decoding of cursor motion using a Kalman filter," *Advances in Neural Information Processing Systems 15*, pp. 133-140, 2003.

[12] L. Paninski, M. Fellows, N. Hatsopoulos, and J. Donoghue, "Spatiotemporal tuning of motor cortical neurons for hand position and velocity," *J. Neurophysiology*, vol. 91: pp. 515-532, 2004.

[13] Y. Gao, M. Black, E. Bienenstock, S. Shoham, and J. Donoghue, "Probabilistic inference of arm motion from neural activity in motor cortex," *Advances in Neural Information Processing Systems 14*, pp. 221–228, 2002.

[14] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object detection," *ICCV*, vol. 14, pp. 786–793, 2002.

[15] G. J. McLachlan and K. E. Basford, *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, 1988.

[16] (2003) Plexon Inc. Dallas, TX.

[17] J. Sanchez, J. Carmena, M. Lebedev, M. Nicolelis, J. Harris, and J. Principe, "Ascertaining the importance of neurons to develop better brain-machine interfaces," *IEEE TBME*, Accepted.