Proceedings of the 3rd International
IEEE EMBS Conference on Neural Engineering
Kohala Coast, Hawaii, USA, May 2-5, 2007

Automatic Spike Sorting For Real-time Applications

Daniel J. Sebald[1] and Almut Branner[2]
[1]Consultant, Madison, WI, USA
[2]Cyberkinetics Neurotechnology Systems, Inc., Salt Lake City, UT, USA

***Abstract*— Real-time applications of spike sorting, *e.g.*, neural decoding, generally require high numbers of channels, and manual spike sorting methods are extremely time consuming, subjective and, generally, do not perform well for low signal-to-noise ratio (SNR) signals. Hence, an automatic method is sought which is efficient and robust in both detecting neural spikes and constructing a classification model of spikes arriving with underlying statistics that are time-varying.**

We present such a system under study for application with a microelectrode array of 96 channels with typically three or four units (*i.e.*, neurons) per channel. There are several novel elements of the system including filtering the neural signal to a frequency band having better SNR for spike detection, a fixed feature space for simple implementation yet adequate resolving capabilities, a Gaussian statistics model also for simple implementation as a log-likelihood classifier, a systematic approach to determining the number of clusters in a pattern recognition problem, and a robust linear discriminant, histogram-based technique for determining boundaries between feature space clusters.

***Index Terms*— Neural spike sorting, unsupervised classification, pattern recognition, clustering methods, brain-machine interface.**

I. Introduction

Penetrating microelectrodes used to extracellularly record neural activity in the brain or other parts of the nervous system frequently record spikes from multiple cells or units [1]. Because of the information contained in the spatial and temporal firing patterns of individual neurons, separating mixtures of neurons—a process called *spike sorting*, which is a form of classification—is essential for neurophysiological studies as well as for other neural prosthetic, diagnostic or therapeutic applications. Much of the automatic spike sorting research has been done either offline or has yet to be integrated efficiently and effectively into online, real-time processing systems [1–4]. Furthermore, one eventual goal is to create wireless, fully implantable devices for long term research as well as human clinical applications. Portable and implantable devices have limited computational resources and memory. For this reason, seeking efficient signal processing and lower bandwidth solutions is critical. As with many signal processing problems, there is a balance sought between reduced signal processing and adequate information retention.

Spike sorting is the process of discerning neural action potential waveforms from multiple sources (neurons) detected by a single electrode based on waveform shape and amplitude [5, 6]. There is typically significant noise present on microelectrodes because of the small electric field potential a single neuron generates upon discharge. The SNR generally can vary from 0 to 20 dB over the duration of one action potential, which is approximately 1–2 ms in duration.

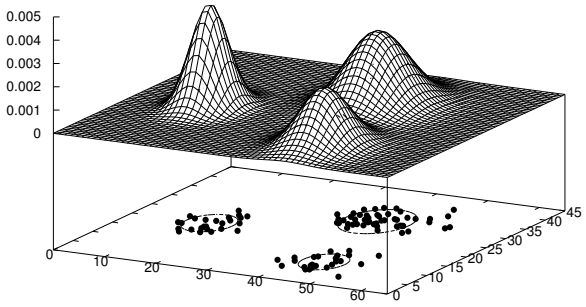

Fig. 1. Gaussian mixture model.

The process of sorting involves building a model with little a priori information about what lies near the tip of an electrode. We are only equipped with the notion of a typical neural spike waveform. The number of neurons and statistical firing patterns are unknown. We have sought an efficient method that we believe is adequate for classifying neural waveforms in such a noisy setting and have decided upon a fixed feature-space based, Gaussian mixture model.

II. Gaussian Mixture Model

Figure 1 illustrates the feature space (base) and the Gaussian mixture model (top) [7, 8]. The x/y-plane of the feature space are parameter values representing some discernible aspect of the neural spikes, discussed later. The z-dimension is the joint probability density function for the sample space comprised of a finite sum of conditional Gaussian densities

$$f_n(\mathbf{x}) = \sum_{i=1}^{N} P_n(c_i) f_n(\mathbf{x}|c_i). \tag{1}$$

where n is the time index, $\mathbf{x}$ the feature space variable, N the number of units, $P_n(c_i)$ the probability that a neural spike originated from class (*i.e.*, neuron or unit) c_i, and $f_n(\mathbf{x}|c_i)$ the conditional probability density of feature-space transformed neural spikes for class c_i. The probabilities and densities are time-varying and, in fact, $P_n(c_i)$ conveys a portion of the information sought for decoding neural activity, the other important information being firing rate [9]. In this unsupervised and time-varying setting, we have little option but to assume equiprobable outcomes when

1-4244-0792-3/07/$20.00©2007 IEEE.

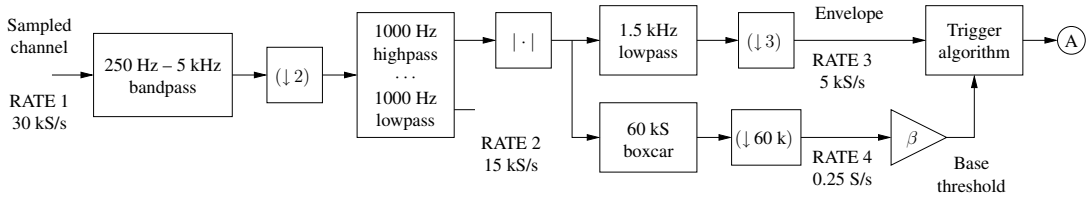

Fig. 2. Subsystem for detecting spikes in noise. All filters used are second order Butterworth filters for efficiency.

classifying, *i.e.*, $P_n(c_i) = P(c_i) = 1/N$ and

$$f_n(\mathbf{x}|c_i) = f(\mathbf{x}|c_i) = \frac{1}{(2\pi)^{\frac{N}{2}}|\mathbf{\Sigma}_i|^{\frac{1}{2}}} e^{-(\mathbf{x}-\boldsymbol{\mu}_i)^T\mathbf{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)/2}. \tag{2}$$

Further information may be available in the form of intended movement in the case of brain-machine interfaces via training exercises or long term statistics, but these topics are for future research.

The method we use to classify feature space vector samples $\mathbf{x}_k$ is the log-likelihood of (2) which results in a fairly efficient squared Mahalanobis distance problem

$$\begin{aligned} \xi_k &= \arg\max_i f(\mathbf{x}_k|c_i) && (3)\\ &= \arg\max_i \Big(-\frac{n}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{\Sigma}_i| \\ &\quad - \frac{1}{2}(\mathbf{x}_k\boldsymbol{\mu}_i)^T\mathbf{\Sigma}_i^{-1}(\mathbf{x}_k - \boldsymbol{\mu}_i)\Big) && (4)\\ &= \arg\min_i \left(\log|\mathbf{\Sigma}_i| + (\mathbf{x}_k - \boldsymbol{\mu}_i)^T\mathbf{\Sigma}_i^{-1}(\mathbf{x}_k - \boldsymbol{\mu}_i)\right). && (5) \end{aligned}$$

where k indexes spike triggers (*i.e.*, not the same as time index n). Based on second order statistics, (5) implies decision boundaries which are piecewise conic sections. The classifier output ξ_k in conjunction with timing information can be used for neurophysiological studies and applications such as real-time neural decoding and device control [9].

With model parameters known (or assumed), further estimates of the Gaussian mean and covariance are computed after collecting a block of K samples for individual model components

$$\hat{\boldsymbol{\mu}}_i[m_i] = \frac{1}{|B_{m_i}|}\sum_{k\in B_{m_i}} \mathbf{x}_k \tag{6}$$

and

$$\hat{\mathbf{\Sigma}}_i[m_i] = \frac{1}{|B_{m_i}|-1}\sum_{k\in B_{m_i}} \left(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i[m_i]\right)\left(\mathbf{x}_k - \hat{\boldsymbol{\mu}}_i[m_i]\right)^T \tag{7}$$

where m_i indexes the evolution of the *block* statistics for class i, B_{m_i} is the set of feature space samples belonging to class i for block m_i, and $|B_{m_i}|$ is the cardinality of set B_{m_i}. We use the general notation for statistics in (6) and (7) because the set B_{m_i} may not necessarily be the exact group of K block samples for unit i as output by classifier (5). This is the case because cluster boundaries may move as new samples are gathered and model statistics are updated according to (6) and (7). Hence, the algorithm will adapt to small changes in waveform averages due to electrode movement over time and will rectify the incorrect placement of the boundaries that may occur occasionally as seen in Fig. 7 between the red and green clusters. The histogram technique discussed later is the global data algorithm which achieves this adaptive mechanism.

III. Detection and Transformation

A. High Frequency Triggering

The method of detecting neural spikes chosen here is different than conventional thresholding methods of triggering upon the waveform passing below or above a predetermined amplitude value. Figure 2 shows the detection subsystem components, which include elements similar to envelope detection used in communication systems.

Typically there is very strong local field potential (LFP) energy in the monitored neural signal below 250 Hz. Above 250 Hz there is important waveform content useful for classification, yet there remains noticeable LFP noise which tapers downward with increasing frequency. Above 1 kHz there remains transient neural spike energy up to approximately 5 kHz while the LFP noise energy continues its decreasing trend with respect to increasing frequency. Naturally there is a white noise component given the large amplification of the signals.

Because of the above spectral characteristics, the initial filtering stage is between 250 Hz and 5 kHz, followed by decimation for sake of computation reduction. The neural signal is then split into a high frequency subband for spike detection and subsequent classification and a low frequency subband which is only used for spike classification. This is done for two reasons: 1) The higher frequency range has improved SNR for triggering, *i.e.*, we seek to trigger on high frequency transients at the leading edge of the nominal neural spike. 2) Removing the lower frequency transient produces shorter duration spikes for improved time resolution when there is the possibility of temporal overlap of spikes from multiple units.

In addition to the restricted frequency band for triggering, we have employed envelope detection as a means to capture both negative and positive leading edge spikes. Another feature of the subsystem in Fig. 2 (the lower path) is an estimate of l_1 signal power used to generate a trigger threshold that is adapted every four seconds. Additional multirate processing is done to decrease computation consumption in these stages.

B. Fixed Feature Space

Several researchers have sought the best feature space for discerning between spike waveforms [2–4]. Here we take a

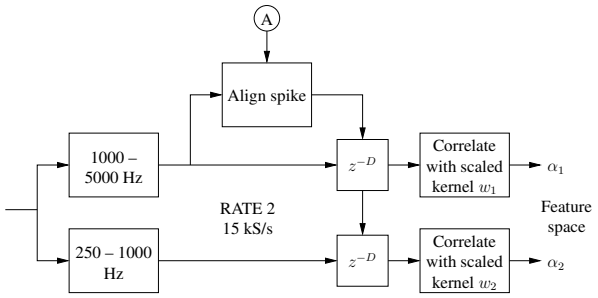

Fig. 3. Subsystem for correlating bandpass filtered spike waveform with nominal waveform filtered to same band.

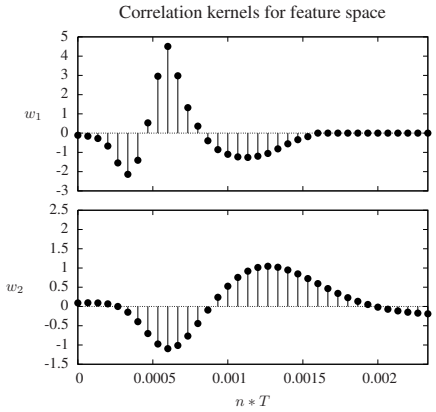

Fig. 4. Kernels used to transform captured, aligned waveforms into the feature space; a high frequency component and a low frequency component of a nominal neural spike as passed through frequency splitting, Butterworth filters.

less optimal but more efficient approach which is to seek a fixed, two-dimensional feature space that in most cases is adequate for separating units into clusters. The most obvious advantage of a fixed feature space is that there is no need for constructing the projections on each channel separately and adaptively. This reduces processing requirements as well as eliminates an optimization algorithm. Also, a fixed projection space is something that a layman (*e.g.*, clinician) can grow accustomed to and can have confidence that the fixed projection is in fact adequate in most cases, whereas an always changing feature space may introduce doubt about whether the projection is over or under-sensitive in one direction or another.

The two dimensional projection chosen utilizes some processing we have already done. That is, we have filtered the waveform to a high frequency band to increase SNR and temporal resolution, so we choose this as one signal basis of our feature space, and the low frequency waveform serves as another basis of our feature space as seen in Fig. 3. The two dimensional, random feature space vector is constructed as $\mathbf{x}_k = \begin{bmatrix}\alpha_1[k] & \alpha_2[k]\end{bmatrix}^T$ where

$$\alpha_i[k] = \sum_{j=0}^{J-1} x_i[n_k + j]w_i[j] \tag{8}$$

is the projection (correlation) of the captured, filtered, and aligned neural signal x_i, starting at time n_k onto the component basis w_i. In this case, fixed kernels, Fig. 4, were calculated as the ensemble average of previously recorded, filtered waveforms. Note the shorter temporal duration of the high frequency kernel w_1 compared to w_2 and its association with the triggering mechanism mentioned earlier.

C. Alignment

One cannot under-emphasize the importance of proper alignment when computing α_1 and α_2, for otherwise small vestigial clusters appear in the feature space. Ideally, one would use a matched filter (correlation) and select the index as peak output in the trigger region. However, this is wasteful, computation-wise. Hence, we approximate with a three point correlation near the peak in the captured high frequency component of the waveform. This approach eliminates vestigial clusters that we have otherwise seen.

IV. Classification and Model Construction

A. Pairwise Cluster Dividing and Combining

To determine the number of clusters in an unsupervised fashion, we propose a novel approach called pairwise cluster dividing and combining (PCDC) which systematically examines a single spike cluster and divides it into two clusters, if warranted, and vice versa examines a pair of spike clusters and combines them into one cluster, if warranted. The technique includes building a histogram based on projections onto either principle component or linear discriminant axes. The concept is to systematically examine individual clusters, *i.e.*, peaks in the histogram, and eventually converge to an appropriate number of clusters.

The inherent questions of the PCDC method are 1) Does the data presented constitute one cluster, or more than one cluster? and 2) If more than one cluster, where is a good place to split the data for an acceptable boundary? Given methods for addressing both questions, as described later in this section, we may add/remove a unit to/from the model and compute its associated statistics $\hat{\boldsymbol{\mu}}_i$ and $\hat{\boldsymbol{\Sigma}}_i$ in (6) and (7) based upon the reclassification according to the boundary.

We begin by assuming only one cluster, or detectable firing neuron, is present simply because it is an easy place to start in terms of building an initial mean and covariance for a group of clusters. After the collection of a block of feature samples, typically between 200 and 300, the initial single cluster may be split as a result of the histogram peak hypothesis test. The resulting subclusters may later split themselves and so on. At some point in time a random, aberrant splitting may occur resulting in too many clusters. However, a combining mechanism, which acts analogous to the division mechanism, will eventually correct such an occurrence.

For the combination test, we pick the cluster nearest the one of interest and consider their combined histogram structure. The inherent question then is once again, Does the data presented constitute one cluster, or more than one cluster? If the histogram peak profile suggests only one cluster is present then the algorithm combines those clusters and computes statistics for the single unit.

PCDC is an adaptable technique constantly readjusting the number of units. Some applications, *e.g.*, neural decoding algorithms, may require the number and order of units to be constant, in which case the occasional appearance of a new unit poses a problem. To deal with this, we have the option to freeze the number of units, or additionally freeze the update of statistics after a training period which depends on the firing rate characteristics of the data. For example, decoding and control applications utilizing movement related signals generally require two to three minutes of training while epilepsy applications utilizing interictal signals from infrequently firing neurons may require up to 20 minutes of training.

B. Clusters as Peaks in Histogram

The question we have posed in the previous section about the number of clusters is reminiscent of work originally done by Fisher [10] in choosing the best direction to project multidimensional data onto a linear subspace. That is, for linear discriminant inner product

$$y = \mathbf{w}^T \mathbf{x}, \tag{9}$$

Fisher had shown, given two classes known a priori, the optimum cluster separation projection is [11]

$$\mathbf{w}_{\mathrm{f}} = \mathbf{S}_{\mathrm{W}}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2). \tag{10}$$

where scatter matrix $\mathbf{S}_{\mathrm{W}} \sim \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2$. Our problem is similar, but involves data not classified a priori and possibly more than two clusters. Initially, when the number of clusters is unknown, we turn to principle components, which gives the direction of most variation, on the premise that if the group of data represents separate clusters then most variation will appear along the direction connecting the means of individual clusters, provided the variance of feature space projections α_1 and α_2 are of approximately equal scale.

Let $\mathbf{v}_{\mathrm{maj}}$ be the principle major eigenvector for the eigen-decomposition of the covariance matrix $\boldsymbol{\Sigma}$, *i.e.*, that which is associated with the larger of the two eigenvalues of the decomposition for two dimensional data. Simply choose

$$\mathbf{w}_{\mathrm{p}} = \mathbf{v}_{\mathrm{maj}}. \tag{11}$$

as the projection direction, and construct univariate data $y_i = \mathbf{w}_{\mathrm{p}}^T \mathbf{x}_k$ for $k \in B_{m_i}$. Figure 5 illustrates the idea of projecting feature space data onto the principle component lying along the line between clusters.

Given the univariate data, the next step is to normalize data along the linear subspace so that it maps to the resolution of the histogram, N_{bin}. We leave out the outer four samples (two on each end) as potential outliers then map according to the affine relationship

$$z_i = m y_i + b \tag{12}$$

where scale $m = N_{\mathrm{bin}}/(y_{\max} - y_{\min})$, inverse of the bin width, and offset $b = (N_{\mathrm{bin}} - m(y_{\min} + y_{\max}))/2$. With this translation, rounding downward to the nearest integer gives the bin in which the vector sample $\mathbf{x}_k$ should be associated. The resulting histogram for the data of Fig. 5 appears in Fig. 6 where multiple peaks are evident. The hypothesis test for determining whether a histogram represents one or more than one cluster needs to be robust to a variety of possible distribution characteristics such as overlapping Gaussian clusters or well separated clusters. For example, the currently implemented level-crossing based strategy is to declare a valley when the bin count falls at least, say, 30% below a peak and a peak when the bin count rises, say, 130% above a valley. If a valley descends to zero, the next peak has to rise above at least, say, 1.6% of the total number of samples in the histogram.

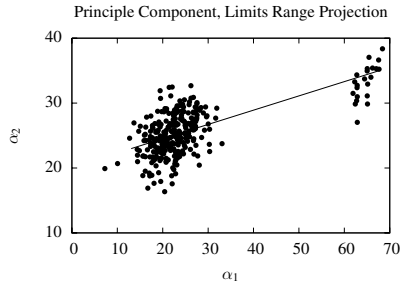

Fig. 5. Projection direction and histogram span based upon principle component linear discriminant.

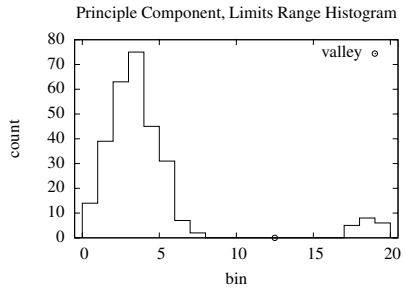

Fig. 6. Resulting 20 bin histogram for data projected onto principle component.

The histogram allows choosing an inherent boundary as the valley separating peaks, as shown in Fig. 6. Hence, herein lies the means alluded to in Section II of reclassifying sample points before adjusting the Gaussian mixture model statistics and why (6) and (7) have general notation.

V. Preliminary Results

The methods presented in this paper have been implemented on a commercially available real-time neural processor and have been tested using 30 kHz sampled human data acquired from a microelectrode array placed in the motor cortex [9]. Figure 7 shows the resulting clusters for a very active channel having four units. Settings were configured for a 250 sample point block size and the system was run for 60 s at which point the number of clusters was frozen.

The boundaries of Fig. 7 are as we might expect for such a mixture given the feature space data. Some related captured spikes for this model are shown in Figs. 8 and 9, which illustrate some interesting details. Several points are worth noting:

1) The temporal waveforms of Fig. 8 illustrate how very active channels often have spikes landing in the tail

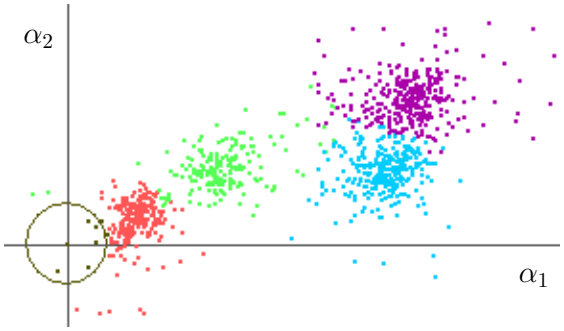

Fig. 7. Pattern space for a single electrode channel after sorting. The circle represents a user-defined noise cluster.

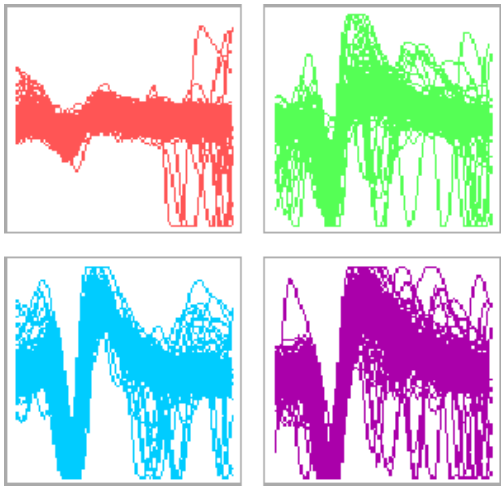

Fig. 8. Individual, sorted unit traces corresponding to cluster model illustrated in Fig. 7.

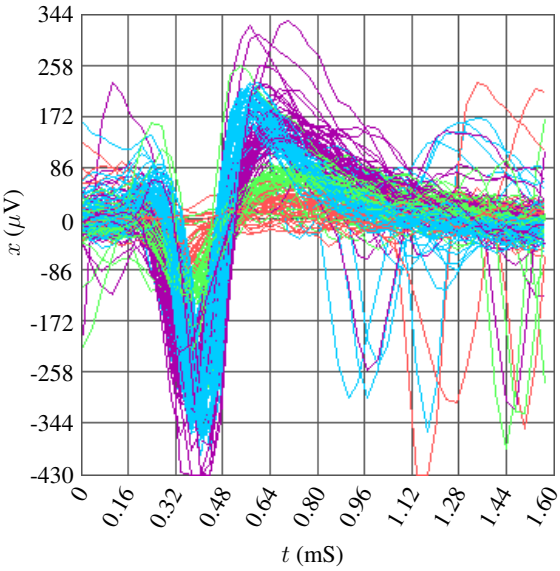

Fig. 9. Ensemble of sorted unit traces corresponding to cluster model illustrated in Fig. 7.

of previous spikes. Yet, detection and classification does not seem to be significantly effected by such occurrences.

2) Figure 9 shows how similar unit 3 (blue) and unit 4 (purple) are and how difficult this problem might be with temporal techniques like time amplitude windows. Furthermore, the fixed feature space we have chosen yields reasonable cluster separation as illustrated in Fig. 7.
3) In Fig. 7, unit 2 (green) extends slightly into what we assume should be the cluster of unit 1 (red). This is a consequence of the fact unit 2 includes several noisy waveforms resembling outliers and does not precisely fit a Gaussian model in the pattern space. A different classification model, such as [2], could be used with the same automated techniques afforded by PCDC and histogram analysis with little alteration.

VI. Conclusions

The real-time automatic spike sorting methods presented here have shown robust classification on actual human cortical action potentials as implemented on the Cyberkinetics NeuroPort™ System. The method does suffer from the fact that successive dividing occurs only as new blocks of data appear. For low firing rate neurons this can mean slow model building. Yet, this training period is fast compared to manually constructing a model for 96 channels. Furthermore, future research can explore reusing the block of previously captured data if too much time has elapsed. The histogram technique is essential to the algorithm, as it introduces global analysis of data rather than sample-by-sample adaptation. Computing histograms is as efficient as, say, computing statistics. The PCDC method is novel as far as we are aware and may have applications in pattern recognition scenarios outside of spike sorting.

References

[1] M. S. Lewicki, "A review of methods for spike sorting: the detection and classification of neural action potentials," *Network: Comput. Neural. Syst.*, vol. 9, pp. R53–R78, 1998.
[2] S. Shoham, M. R. Fellows, and R. A. Normann, "Robust, automatic spike sorting using mixtures of multivariate t-distributions," *Journal of Neuroscience Methods*, vol. 127, pp. 111–122, 2003.
[3] M. Sahani, "Latent variable models for neural data analysis," Computation and Neural System, California Institute of Technology, Pasadena, 1999.
[4] G. Santhanam, M. Sahani, S. I. Ryu, and K. V. Shenoy, "An extensible infrastructure for fully automated spike sorting during online experiments," in *Proceedings of the 26th Annual International Conference of the IEEE EMBS*, vol. 127, San Francisco, CA, 2004, pp. 4380–4384.
[5] C. M. Gruner and D. H. Johnson, "Comparison of optimal and suboptimal spike sorting algorithms to theoretical limits," *Neurocomputing*, vol. 38–40, pp. 1663–1669, Jun. 2001.
[6] S. M. Bierer and D. J. Anderson, "Multi-channel spike detection and sorting using an array processing technique," *Neurocomputing*, vol. 26–27, pp. 947–956, Jun. 1999.
[7] G. J. McLachlan and D. Peel, *Finite Mixture Models*. New York: Wiley, 2000.
[8] K. H. Kim, "A fully-automated neural spike sorting based on projection pursuit and Gaussian mixture model," in *IEEE Conf. on Eng. in Med. & Bio.*, Arlington, VA, March 16–19 2005, pp. 151–154.
[9] L. R. Hochberg, M. D. Serruya, G. M. Friehs, J. A. Mukand, M. Saleh, A. H. Caplan, A. Branner, D. Chen, R. D. Penn, and J. P. Donoghue, "Neuronal ensemble control of prosthetic devices by a human with tetraplegia," *Nature*, vol. 442(7099), pp. 164–171, July 13 2006.
[10] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of Eugenics*, vol. 7, pp. 179–188, 1936.
[11] R. Duda, P. Hart, and D. Stork, *Pattern Classification*. New York: John Wiley & Sons, 2001.