Clustering Spikes in Multi-unit Neural Signals into Dominant and Remainder Classes

David Joseph Rodriguez

Advisor: Dr. Patrick Wolf
Pratt School of Engineering
Duke University
April 27, 2006

Submitted in partial fulfillment of the requirement for Graduation with Distinction in the Department of Electrical and Computer Engineering

Abstract

Spike sorting is a critical component in the brain machine interface. The traditional objective of spike sorting is to classify each spike, or action potential, in a multi-unit waveform into a separate class representing the neuron that fired the spike. A major problem in spike sorting is the variable and unknown number of classes. This paper investigates the efficacy of sorting neural signals into two classes - a dominant class, which ideally includes spike occurrences of the most prominent neuron, and a remainder class, which ideally includes spike occurrences of all other distinguishable neurons - rather than sorting into the 'correct' number of classes. This results in fewer classification errors, since classifications amongst neurons in the remainder class do not have to be made. A simulated neural dataset consisting of 30 multi-unit signals each with 4 distinguishable, or near-field, neurons was created using Matlab. The signals were represented as orthogonal 'features' by performing principle component analysis of the spike waveforms. Two clustering algorithms, k-means and expectation maximization using a Gaussian mixture model (GMM EM), were used on the first two principle component scores of each signal. Clustering was performed with the number of classes initialized with values 2, 3, 4, and 5. After sorting, the most prominent cluster was assigned to the dominant class and all other clusters were assigned to the remainder class. The correct and false classifications were determined and analyzed. Initializing the sorting algorithms to sort the data into the number of near-field neurons or one more than the number of near-field neurons and then determining the dominant and reminder classes had the least classification errors for both algorithms. Initializing the number of classes to one less than the number of near-field neurons caused a marginal increase in errors. Initially sorting the data into two classes resulted in much greater classification error. Overall, the k-means algorithm performed better on average, but the GMM EM performed better when initialized with appropriate seeds.

Introduction

Brain machine interfaces use extracellular electrodes to acquire neural signals. The electrodes record a signal that is the summation of the potentials of multiple neurons surrounding an electrode. Multiple signals are then processed and used to actuate a prosthetic device. It is desirable to have fully implantable systems with wireless data acquisition hardware instead of transcutaneous wires to communicate between the electrodes and the prosthesis. However, telemetry in all wireless systems is limited by bandwidth. Consequently, instead of transmitting the raw data recordings, spike detection is used to transmit only the action potential waveforms and their arrival times, which compresses the data sent. Ideally, only the spike occurrences need to be transmitted. However, this requires sorting the detected spikes into separate groups, or classes, for each distinguishable neuron surrounding the extracellular electrode (Lewicki 1998).

An in vivo, real-time system must implement an autonomous spike sorting algorithm. Traditionally, the goal of spike sorting is to classify each distinguishable action potential in the multi-unit signal into separate classes representing the neuron that fired the action potential. A major problem in spike sorting is the variable and unknown number of distinguishable neurons that contribute to a recorded signal. This corresponds to number of classes in which the signal should be sorted. The number of classes is important since it is used to initialize most sorting algorithms. Unfortunately, there is no general and satisfactory solution to determine the number of classes. Bayesian statistical approaches can be used to determine the probable number of classes; however these models are limited in spike sorting applications by their underlying assumptions (Cheeseman and Stutz 1988, Chickering and Heckerman 1997, DasGupta and Raftery 1998, Lewicki 1998).

This paper investigates sorting neural signals into two classes: a dominant class which ideally includes spike counts of the most prominent neuron, and a minor class which ideally includes spike counts of all other distinguishable neurons. The classification results in a single-unit dominant class and a single-unit or multi-unit remainder class depending on the number of distinguishable neurons in the signal. The overall classification errors are lower because the conventional sorting errors amongst neurons in the remainder class are no longer errors, since the previously disparate classes are grouped into one.

Although one might suspect that grouping of classes results in loss of information, there is evidence that an unsorted multi-unit signal may contain as much or more information as a sorted signal with a reasonable 15% discrimination error (Wheeler and Heetderks 1982, Harris et al 2000) under certain conditions (Won and Wolf 2003). Having two classes also has the benefits of limiting and fixing the bandwidth per electrode, which facilitates implementation of the BMI. Furthermore, since the average number of distinguishable neurons is between 2 and 3 (Nicolelis et al 1997, Reich et al 2001), it is reasonable to select 2 as the number of classes.

The question that is addressed here is: What is the best way to group action potentials in multi-unit signals into dominant and remainder classes? To explore this, both k-means and an expectation maximization algorithm using a Gaussian mixture model were used to cluster the first two principle component scores of neural signals. Both algorithms were initialized with a different number of classes ranging from 2 to 5. After the clustering algorithms were run, the resulting clusters were grouped into dominant and remainder classes.

Methods

Creation of Data Sets:

Matlab was used to generate all signals and perform all spike sorting. The stimulated signals were composed of three parts: action potentials from near-field neurons, action potentials from far-field neurons, and thermal noise. The near-field neurons produced the distinguishable and large action potentials in a neural signal. The background noise was composed of far-field neuron action potentials and thermal noise. 4 near-field neurons each with unique amplitudes and 40 far-field neurons were modeled to create each signal. Each simulated near-field and far-field neuron was assigned a unique action potential waveform from a database of waveforms recorded in vivo from owl monkeys and macaques. All action potential waveforms consisted of 46 data points sampled at 31.25 KHz, were aligned by their minimum peak, and were normalized using their minimum peak amplitude.

The firing times of each near-field and far-field neuron were assumed to occur according to a Poisson random process with a mean firing parameter between 20 and 30 Hz. Thus, a unique sequence of firing times was generated for each neuron with a Poisson number generator. For each modeled neuron in a signal, identical action potentials from the database were placed in the signal at the times determined by the Poisson generator. Subsequently, all spike times failing to meet a minimum refractory period of 2 milliseconds were eliminated and all overlapping near-field action potentials were removed from the signal.

After the separate near-field and far-field neuron signals were created, both were up-sampled to a rate of five times their original frequency. A thermal noise signal, modeled by white Gaussian noise with standard deviation of 1 microvolt, was summed with the far-field neuron signal to comprise the overall noise signal. Next, both the noise and near-field neuron signal was band-passed through a two pole Butterworth filter with poles of 450 and 6500. The signals were then down-sampled back to the original sampling rate and summed together to create the final signal. The resulting signals were one second long and had a sampling rate of 31.25 KHz. Spike times and classes of all near-field neurons for each signal were recorded during signal generation.

Thirty signals were generated using the method described above with varying near-field and far-field amplitude ratios. Three different sets of ratios were used to produce three groups each comprised of 10 signals named ‘Low,’ ‘Medium,’ and ‘High’ after their resulting signal to noise ratio(SNR). The signal to noise ratio of each near-field neuron was calculated using the formula:

$$SNR_i = \frac{RMS(s_i)\,|_{s_i \neq 0}}{RMS(noise)} \quad (1)$$

where $RMS(s_i)\,|_{s_i \neq 0}$ refers to the root mean square of the non-zero terms of the i^{th} near field neuron. The mean and standard deviation of the SNR for each near-field neuron in each of the three groups of signals are described in Table 1. Example signals from each of the groups of data along with their firing patters are shown in Figure 1.

	Neuron 1		Neuron 2		Neuron 3		Neuron 4		Overall	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Low SNR	1.96	0.26	1.76	0.30	1.27	0.14	1.01	0.15	1.50	0.21
Medium SNR	2.49	0.42	2.18	0.43	1.88	0.30	1.22	0.17	1.94	0.33
High SNR	3.19	0.46	2.54	0.66	1.88	0.25	1.43	0.22	2.26	0.40

Table 1: The mean and standard deviation SNR for low SNR, medium SNR, and high SNR datasets.

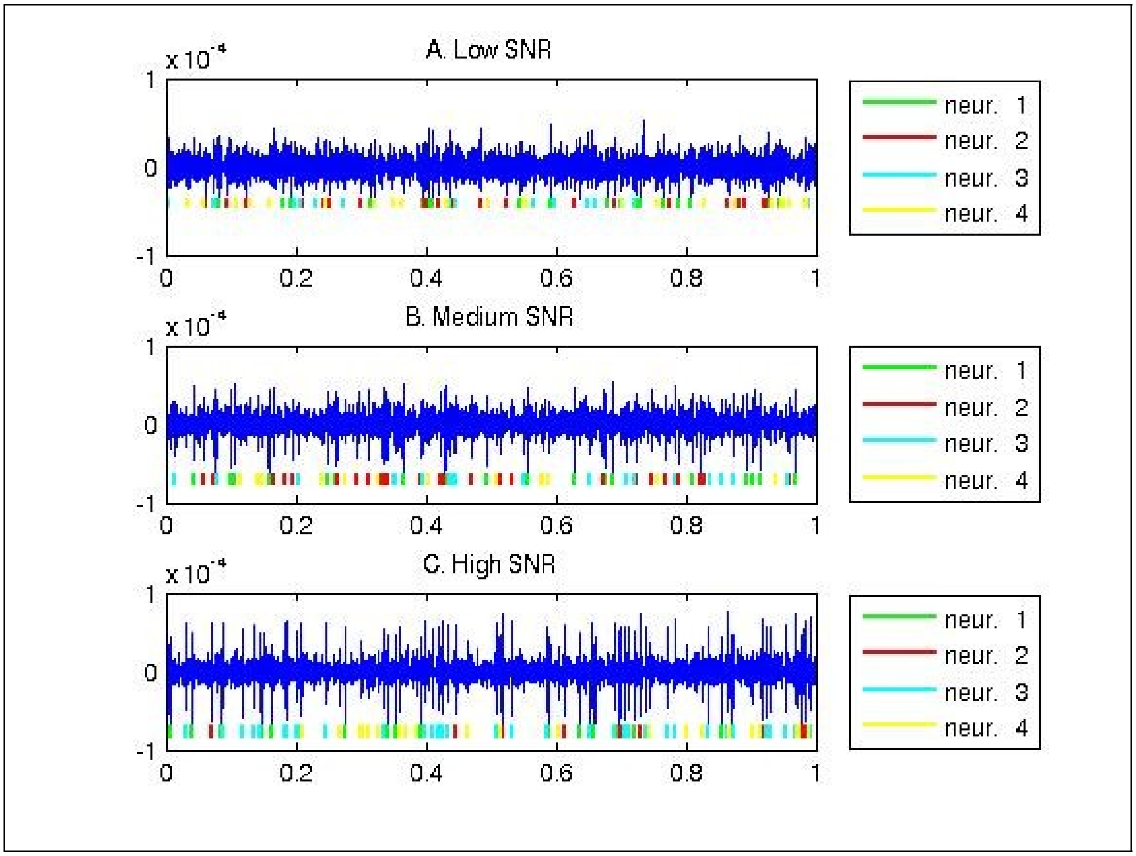

Figure 1: Signal waveform with firing pattern for low SNR, medium SNR, and high SNR datasets.

Data preparation: Detection and PCA

Rather then using a spike detection algorithm, the known spike times from signal generation were used to isolate all spikes in a signal. The peaks of the spikes were then aligned on the known negative peak of the original action potential (before the noise signal was added to the near-field signal). The action potential waveforms were isolated by taking 15 data points to the left of the negative peak and 30 data points to the right of the negative peak, totaling 46 data points or about 1.5 milliseconds. The detected and aligned spikes are shown in Figure 2 for a Low SNR signal and in Figure 3 for a High SNR signal. After the detection and alignment of the spikes, principle component analysis (PCA) was performed on the detected action potential waveforms.

Cluster Analysis:

Cluster analysis was performed on the first two principle component scores using two different clustering algorithms: k-means and expectation maximization using a Gaussian mixture model. The k-means algorithm was initialized with data points selected at random from the dataset. The covariance and mean matrices for the Gaussian mixture model were initialized using a k-means algorithm that was also initialized with random seeds. Since random seeds were used, each method of clustering was performed 20 times for a given number of initial classes.

The Gaussian mixture model represents each class with a multivariate Gaussian distribution. Given a specific class, the probability that spike x was generated from neuron c_k, the likelihood, was modeled by:

$$p(x \mid c_k, \Theta_k) = \frac{1}{2\pi^{d/2}\sqrt{\det(\Sigma_k)}} \exp\left(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right) \quad (2)$$

where $\Theta_{1:k} = \{\mu_1, \Sigma_1 ..., \mu_K, \Sigma_K\}$ and μ_k and Σ_k are the mean and covariance matrix of the PCA coefficients for waveforms in class c_k. The probability that a data point was a member of a given class was determined by:

$$p(c_k \mid x, \Theta_{1:k}) = \frac{p(x \mid c_k, \Theta_k) p(c_k)}{\sum_k p(x \mid c_k, \Theta_k) p(c_k)} \qquad (3)$$

where $p(c_k)$ is the probability of class k, or the relative firing frequency of class k. The class parameters were maximized by maximizing the overall likelihood of the data:

$$p(x_{1:n} \mid \Theta_{1:k}) = \prod_{n=1}^{N} p(x \mid c_k, \Theta_{1:k}) \qquad (4)$$

In this study, the expectation maximization algorithm proposed in Dempster et al (1977) was used to find the parameters of the Gaussian mixture model. After the algorithm was run, hard cluster boundaries were determined by assigning a data point to the class with the greatest probability of membership.

Cluster analysis of each of the 30 generated signals was performed using both the k-means and expectation maximization clustering algorithms initialized with a different number of classes, ranging from 2 to 5. The cluster that had a centroid with the closest Euclidean distance to the mean of the data points representing the most prominent neuron was assigned to the dominant class. The most prominent neuron was defined as the neuron in the signal with the greatest amplitude. All other clusters were then grouped into the remainder class. Correct classifications were defined when a spike originating from the most prominent neuron (known *a priori* from data generation) was classified as belonging to the dominant class, or when a spike from any other near-field neuron was classified as belonging to the remainder class. Each signal was sorted with each clustering algorithm and each initial class number 20 times with different initial seeds. The mean of the classification results of the 20 trials and the classification results for the trial with the minimum number of classification errors were both recorded.

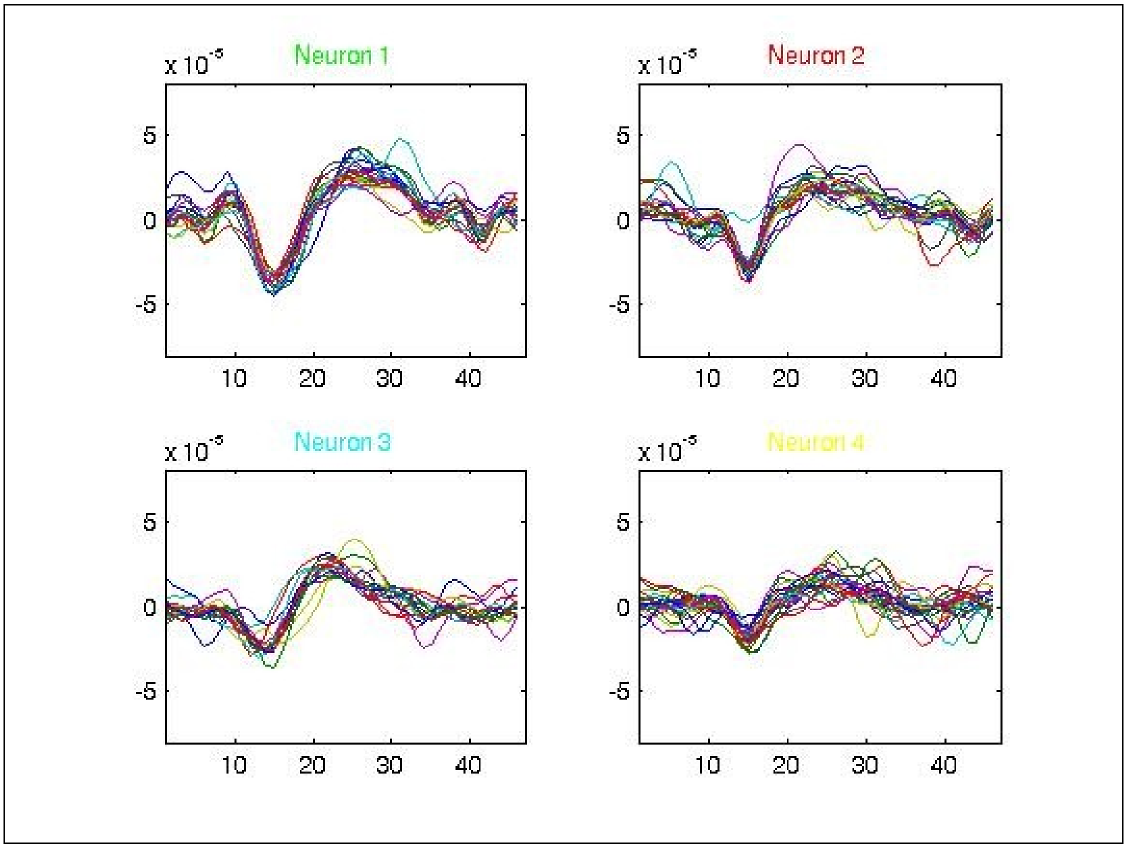

Figure 2: Detection and alignment of action potential for each near-field neuron with low SNR.

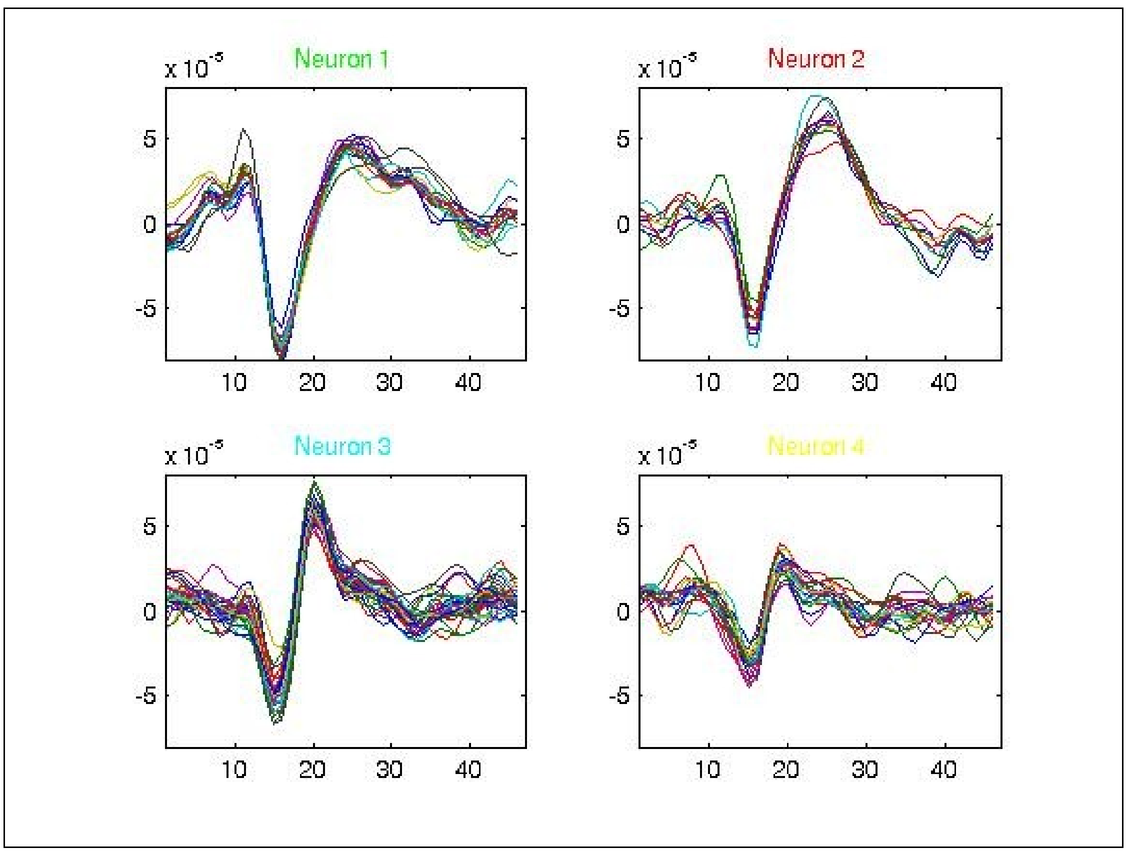

Figure 3: Detection and alignment of action potential for each near-field neuron with high SNR.

Results

The results are presented in three parts. First, the clustering results of two example signals, one signal with a low SNR and one signal with a high SNR, are graphically displayed to better demonstrate the signals used, clustering procedures, selection of the dominant and remainder classes, and determination of correct and incorrect classifications. Second, the overall mean results obtained from taking the mean of the correct classifications (CC) and incorrect classifications (IC) over the 20 trials performed on each signal are presented. Third, the overall minimum error results are presented. The minimum error results were obtained by choosing the classification out of the 20 trials that resulted in the minimum number of errors.

Figure 4 and 5 show the classification results for an example low SNR signal from the k-means and expectation maximization algorithm with 2, 3, 4, and 5 classes. Figures 6 and 7 show the classification results for an example high SNR signal. The different color data points in all figures correspond to the actual near-field neuron responsible for firing the action potential, with the green color always representing the most prominent neuron. This is known from signal generation. The black diamonds show the means of the data points for each near-field neuron. The crosses show the centroids of the clusters determined by the sorting algorithms.

All data points outlined in a dark circle were determined by the sorting algorithm to be in the dominant class. All data points not outlined by a dark circle were determined by the sorting algorithm to be in the remainder class. Correct classifications result when a green data point is outlined in a dark circle (an action potential from the most prominent neuron is determined to be in the dominant class) or a blue, yellow, or red data point is not outlined by a dark circle (an action potential not from the most prominent neuron is determined to be in the remainder class). False classifications result when a green data point is not outlined by a dark circle or a blue, yellow, or red data point is outlined by a dark circle. The number of correct classifications and false classification resulting from the sorting is summarized in Table 2.

		2 classes		3 classes		4 classes		5 classes	
		CC	IC	CC	IC	CC	IC	CC	IC
Low SNR	K-Means (fig 4)	75	6	77	4	77	4	69	12
	GMM EM (fig 5)	71	10	75	6	71	10	68	13
High SNR	K-Means (fig 6)	75	10	85	0	77	8	85	0
	GMM EM (fig 7)	75	10	85	0	75	10	75	10

Table 2: Classification results summary from example signal classifications shown in Figures 4-7.

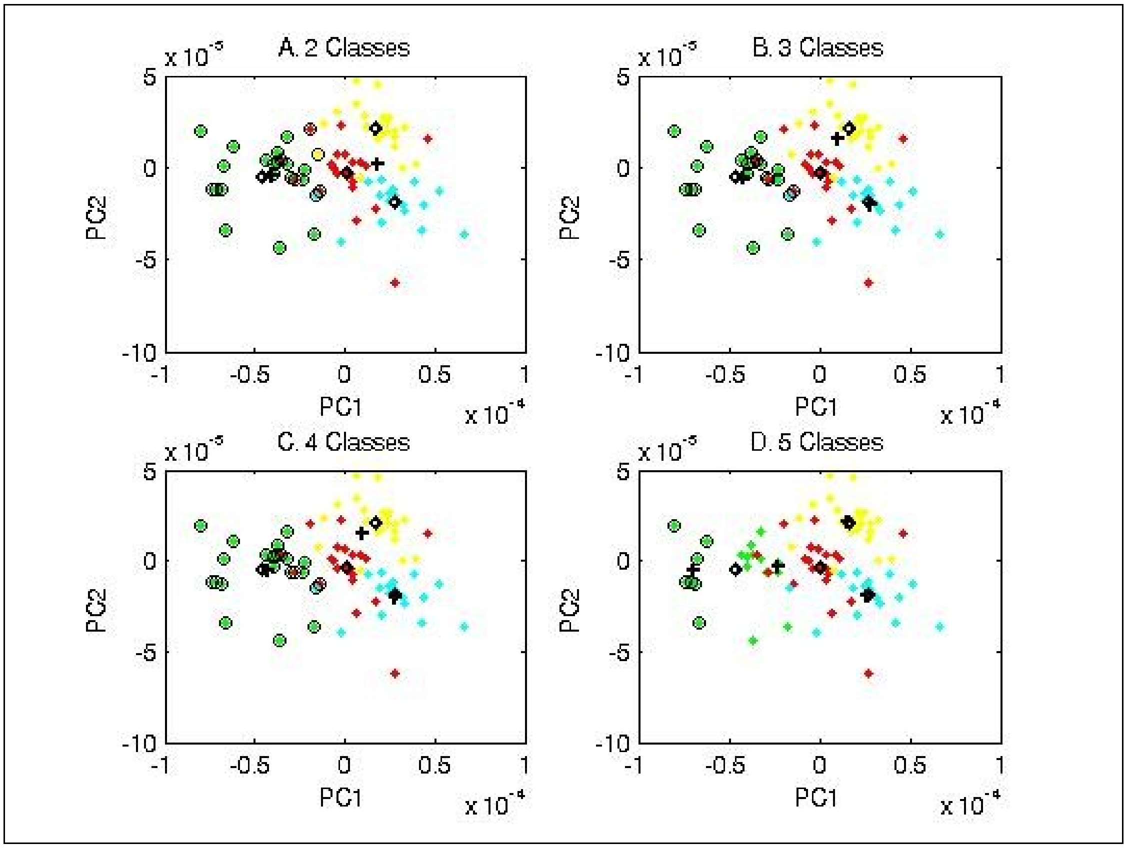

Figure 4: K-means clustering of a low SNR signal with 2, 3, 4, and 5 classes.

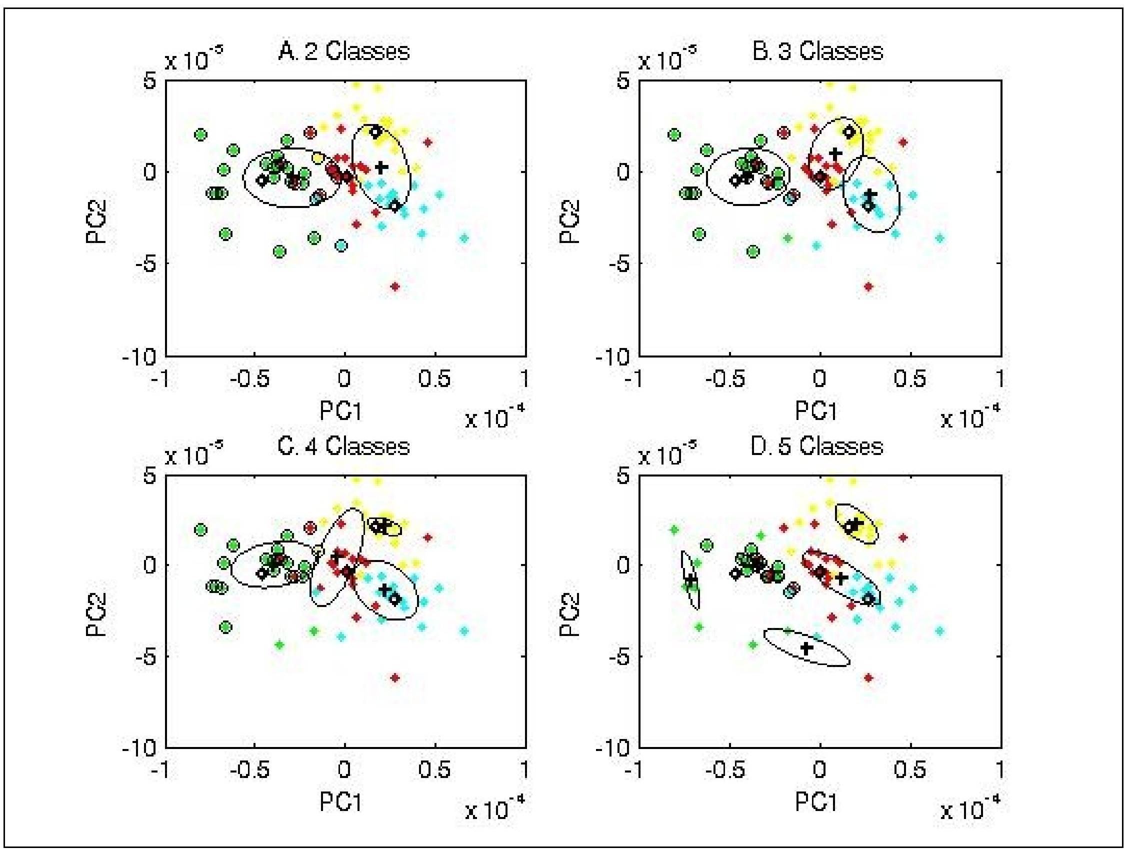

Figure 5: GMM EM clustering of a low SNR signal with 2, 3, 4, and 5 classes.

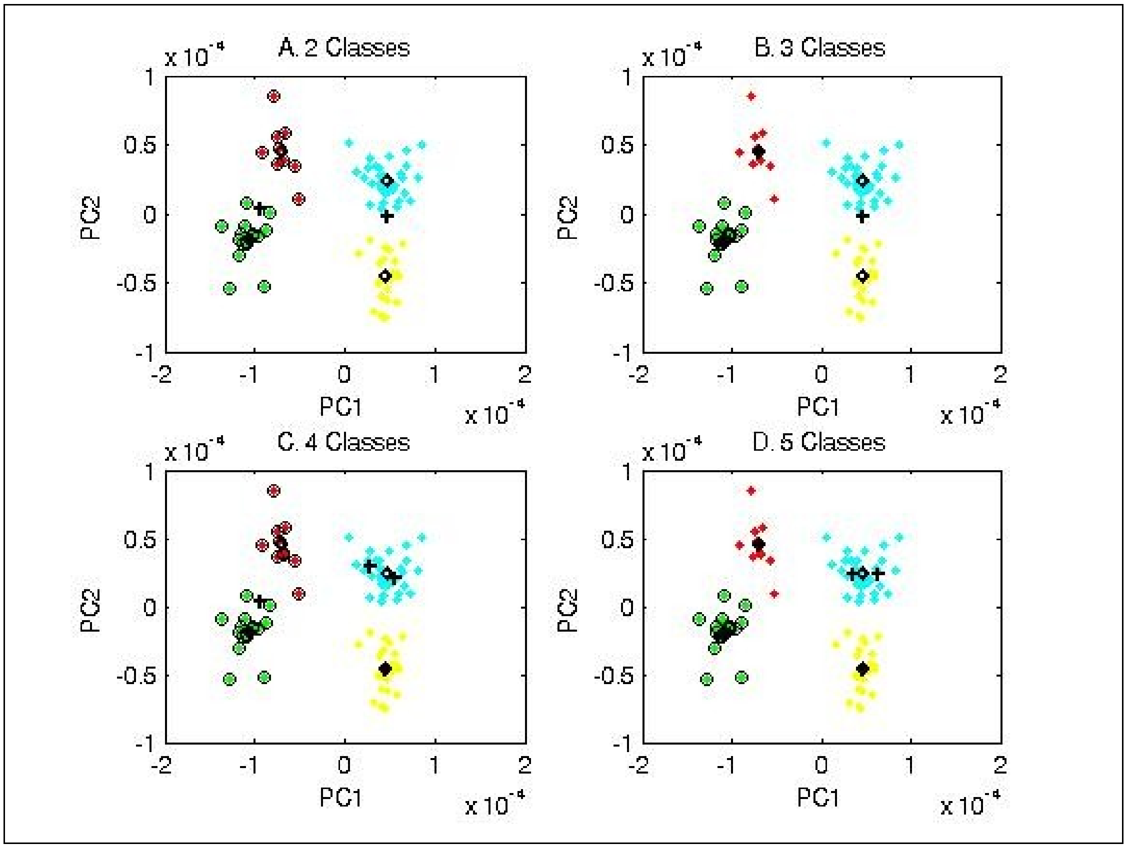

Figure 6: GMM EM clustering of a high SNR signal with 2, 3, 4, and 5 classes.

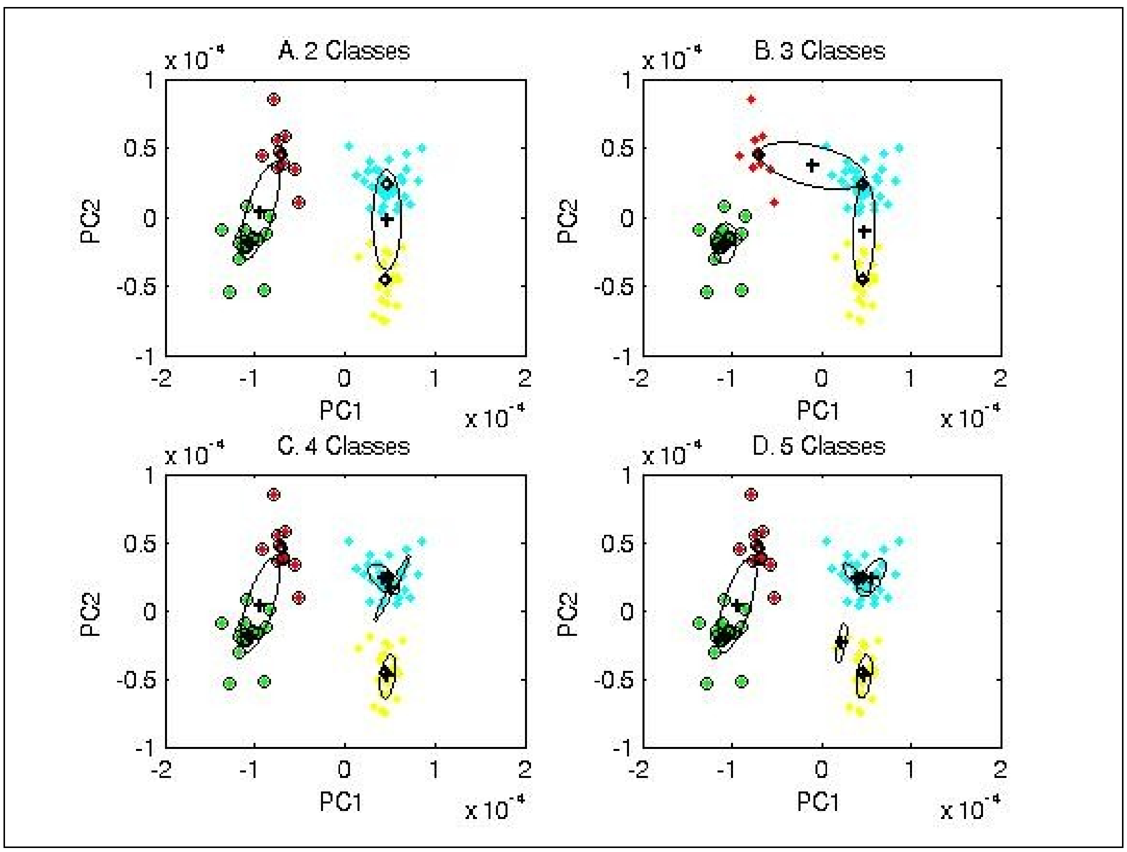

Figure 7: GMM EM clustering of a high SNR signal with 2 , 3, 4, and 5 classes.

Each of the four plots in Figures 4-7 demonstrates the result of a single classification with random seeds. Similar classifications were performed 20 times for each signal. The results of classification are presented in two ways: (1) the mean of 20 classifications (2) the one classification out of the 20 classifications that had the minimum errors. The summary of mean results is displayed graphically in Figure 8 and Figure 9 and numerically in Table 3. Likewise, the summary of the minimum error results are displayed in Figure 10, Figure 11, and Table 4.

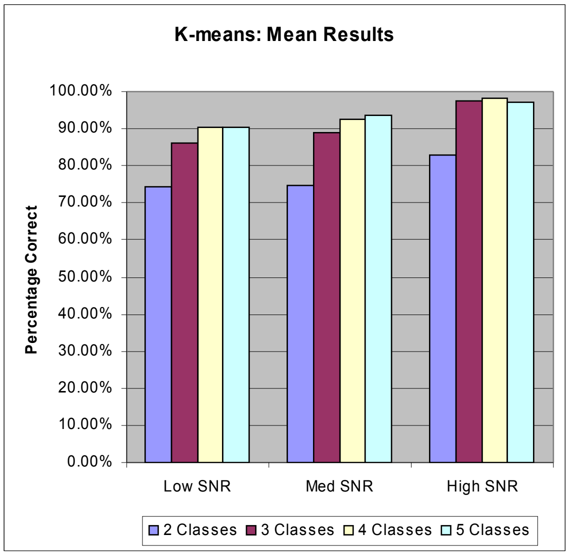

Figure 8: The mean classification results using the k-means algorithm.

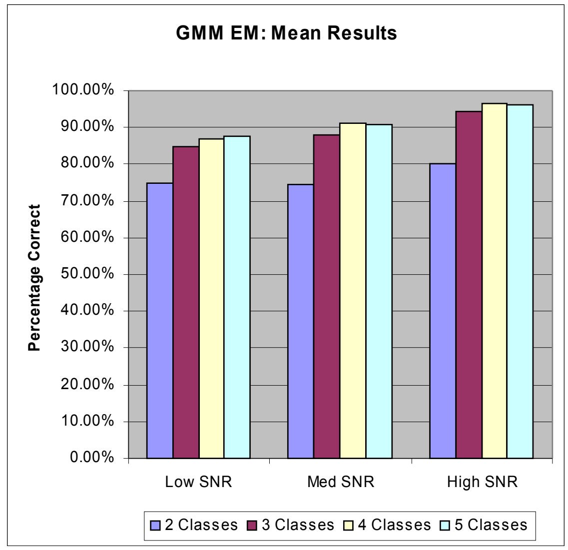

Figure 9: The mean classification results using the GMM EM algorithm.

		2 classes CC	2 classes IC	3 classes CC	3 classes IC	4 classes CC	4 classes IC	5 classes CC	5 classes IC
Low	K-means	74.24%	25.76%	86.20%	13.80%	90.48%	9.52%	90.32%	9.68%
	GMM EM	74.99%	25.01%	84.58%	15.42%	87.05%	12.95%	87.57%	12.43%
Med	K-means	74.85%	25.16%	88.86%	11.15%	92.64%	7.36%	93.61%	6.39%
	GMM EM	74.55%	25.45%	87.94%	12.06%	91.15%	8.85%	90.87%	9.13%
High	K-means	83.08%	16.92%	97.58%	2.43%	98.18%	1.82%	97.22%	2.78%
	GMM EM	79.98%	20.02%	94.19%	5.82%	96.53%	3.47%	96.06%	3.95%
Average	K-means	77.39%	22.61%	90.88%	9.12%	93.77%	6.23%	93.72%	6.29%
	GMM EM	76.51%	23.49%	88.90%	11.10%	91.58%	8.42%	91.50%	8.50%

Table 3: Summary of correct classifications (CC) and incorrect classifications (IC) for the mean classification results.

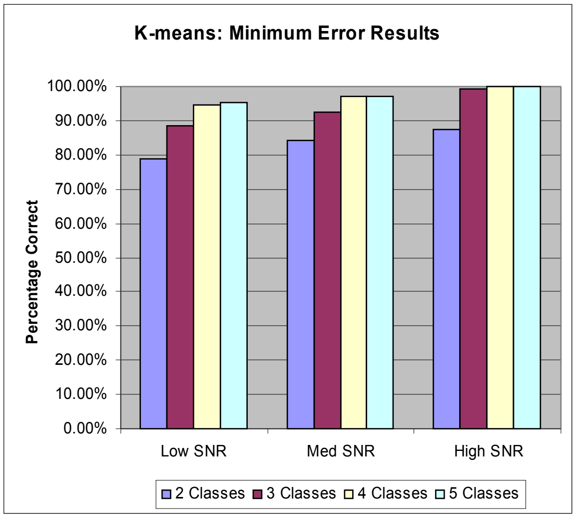

Figure 10: The minimum error classification results using the k-means algorithm.

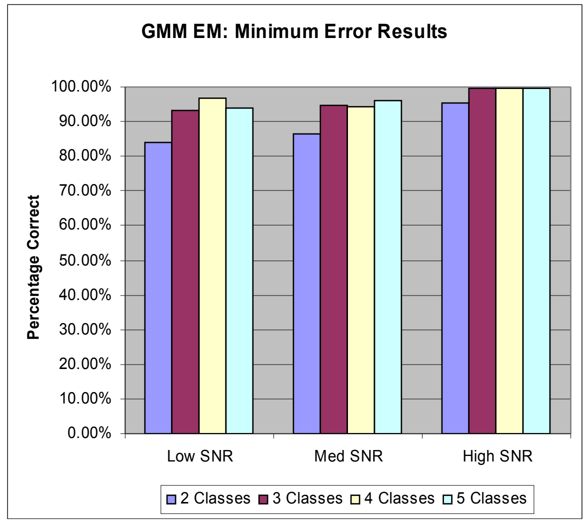

Figure 11: The minimum error classification results using the GMM EM algorithm.

		2 classes		3 classes		4 classes		5 classes	
		CC	IC	CC	IC	CC	IC	CC	IC
Low	K-means	78.73%	21.27%	88.53%	11.47%	94.48%	5.52%	95.19%	4.81%
	GMM EM	83.81%	16.19%	93.22%	6.78%	94.25%	5.75%	94.02%	5.98%
Med	K-means	84.24%	15.76%	92.53%	7.47%	96.97%	3.03%	97.31%	2.69%
	GMM EM	86.54%	13.46%	94.83%	5.17%	96.86%	3.14%	96.19%	3.81%
High	K-means	87.28%	12.72%	99.11%	0.89%	99.88%	0.12%	99.88%	0.12%
	GMM EM	95.43%	4.57%	99.77%	0.23%	99.77%	0.23%	99.77%	0.23%
Average	K-means	83.42%	16.58%	93.39%	6.61%	97.11%	2.89%	97.46%	2.54%
	GMM EM	88.59%	11.41%	95.94%	4.06%	96.96%	3.04%	96.66%	3.34%

Table 3: Summary of correct classifications (CC) and incorrect classifications (IC) for the minimum error classification results.

Discussion

The results demonstrated that performing cluster analysis on a signal with 4 near-field neurons with the initial number of classes set to 4 or 5 yielded the least errors. Cluster analysis with the initial number of classes set to 2 yielded the most errors. Also, it was demonstrated that the k-means algorithm performed better on average, but the GMM EM performed better when initialized with appropriate seeds.

From the mean results, it was determined that initializing sorting algorithms to sort the data into the number of near-field neurons (4 classes) or one more than the number of near field neurons (5 classes) and then determining the dominant and reminder classes had the least classification errors (K-means error[4] = 6.23%, GMM EM error[4] = 8.42% & K-means error[5] = 6.29%, GMM EM error[5] = 8.50%, respectively). Sorting the data into one fewer class than the actual number of near-field neurons (3 classes) resulted in slightly greater error (K-means error[3] = 9.12%, GMM EM error[3] = 11.10%) Sorting the data into 2 classes initially resulted in much larger error (K-means error[2] = 22.61%, GMM EM error[2] = 23.49%).

The minimum error results showed a similar pattern. Sorting the data into the number of near-field neurons (4 classes) or one more than the number of near-field neurons (5 classes) yielded the least error (K-means error[5] = 2.54%, GMM EM error[5] = 3.34% & K-means error[4] = 2.89%, GMM

EM error[4] = 3.04 %, respectively). Sorting first into 3 and 2 classes resulted in a greater number of errors, with sorting into 2 classes resulting in the most errors (K-means error[3] = 6.61%, GMM EM error[3] = 4.06% & K-means error[2] = 16.58%, GMM EM error[2] = 11.41%, respectively).

The k-means clustering algorithm performed better on average than expectation maximization algorithm using a Gaussian mixture model. However, the GMM EM algorithm performed better when the seeds for algorithm initialization were better chosen. This demonstrates the sensitivity of the GMM EM algorithm to initial seed parameters. It also demonstrates the ability for the GMM EM algorithm to better classify the data than the k-means algorithm when initialized with appropriate seeds.

The least errors resulted when the data was clustered using the GMM EM algorithm with the initial classes set to 4 or 5 classes. This is because 4 Gaussians modeled 4 near-field neurons well. 5 Gaussians also modeled 4 near-field neurons well because in some cases 2 Gaussian distributions model the data for a single neuron. Figure 7D demonstrates this phenomenon. The most errors resulted when the Gaussian mixture model was used with the initial classes set to 2. This occurred because a Gaussian distribution poorly approximates the multiple clusters in the remainder class. Figure 7A demonstrates this phenomenon. Unlike a Gaussian distribution, the data points making up the remainder class would have dearth of data points in the center, since it would likely be between clusters. Perhaps modeling the remainder cluster with a different distribution, such as a uniform distribution, would yield better results.

The results show a general pattern that clustering the data with the initial class parameter set to the true number of distinguishable neurons yields the least amount of classification error when the clusters are regrouped into dominant and remainder classes. Unfortunately, this algorithm would still require the spike sorting system to determine the number of classes, or distinguishable neurons, in the recorded signal. Thus, the problem of determining the correct number of initial classes persists. Although this study examined the best way to group a multi-unit signal into 2 classes, future work must be done to quantify the information gained or lost as a result of classification.

Limitations

The use of simulated datasets provides realistic signals with known spike times, neuron classes, and waveform morphologies. Since the true spike classes are known from signal generation, the results of sorting were compared to this known information to qualify whether a spike is correctly or incorrectly classified. If empirical in vivo data were used, the spike times and neuron classes would not be known *a priori* to compare to the sorting algorithms' results.

Several assumptions were made to generate the signals. First, it was assumed that each neuron induces identical waveforms at the recording electrode each time that it fires. This obviates the need to model membrane dynamics and models a system with no electrode drift. Second, firing times were assumed to occur according to a Poisson process. This provides signals with known mean firing times without predefined firing patterns. Thirdly, no overlaps were included in the signals. Although this does not correctly model a true neural signal, it creates an ideal signal for testing clustering algorithms. When two neurons fire at the same time, the resultant waveform recorded is the summation of the electrical contributions of both neurons. In clustering analysis these waveforms are undesirable since these waveforms do not fit into any class representing one neuron. Furthermore, outliers, which correspond to overlapping spikes, are not sufficiently described by the first two principle component scores (Lewicki 1998).There are several methods to remove overlaps or model overlaps as originating form one neuron by subtraction methods before clustering (Jansen 1990, Chandra and Optician 1997, Lewicki 1994). However, rather than use a specific method to account for overlaps, a signal without overlaps assumes an ideal method that has already accounted for all overlaps, and allows for an independent analysis of the clustering algorithms.

Spike detection is the first step in spike sorting where all spikes of distinguishable neurons are identified and their waveforms isolated. There are numerous spike detection algorithms including threshold, energy, and matched filter with varying degrees of efficacy (Obeid and Wolf 2004). Since all signals were generated, the known spike times were used to locate all spikes of the near-field neurons from the overall signal. This obviates the need to select any of the many detection algorithms. By using known spike times of the signal, ideal spike detection was achieved where all true spike and no false spikes are detected. Thus, all errors can be attributed to the sorting process.

Since both k-means and expectation maximization algorithms are iterative and converge to a local minimum, they are sensitive to initial seeds. There are various initialization techniques for both clustering algorithms, all of which affect the results of clustering (Wood et al 2004). Rather than choose a deterministic method to initialize the algorithms, random seeds were used to initialize the k-means algorithm. The Gaussian mixture model's covariance and mean matrices were initialized with the results of a k-means algorithm that was initialized with random seeds. The use of random seeds precludes a deterministic outcome and allows for a general analysis of the clustering algorithms. To analyze the effects of the initial seeds, each algorithm was run 20 times on each signal with each number of initial classes. Then, the mean of the results of clustering for the 20 trials and the result with the least errors out of the 20 trials was determined. The mean results demonstrate the average clustering results using a random seed algorithm and shows the sensitivity to initial seeds of both algorithms. The minimum error results demonstrate the classification results if the best initial seeds from the 20 trials were used. Thus, if more ideal seed initialization algorithm were used to select the initial seeds, results closer to the minimum error results could be expected.

A priori information from signal generation was used to choose the dominant class; specifically, the mean of data points from the most prominent neuron. However, this information would not be known in an autonomous spike sorting system and another criterion would have to be used. A simple algorithm could choose the class with the greatest SNR ratio or greatest average maximum or peak to peak amplitude. Another algorithm could use the Gaussian mixture model, which quantifies the certainty of classification results. The probability that a spike is classified in a particular class is known (see equation 3). This can be used to determine the class that statistically has the least errors (Lewicki 1998). This class can be chosen as the dominant class, thus minimizing the overall errors.

Conclusion

Several methods to classify spikes in a multi-unit signal into dominant and remainder classes were suggested and tested. Classifying a neural signal into dominant and remainder classes could minimize the overall classification errors and leave a multi-unit remainder class that contains a significant amount of information. It also fixes the bandwidth per electrode making implementation of such an in vivo sorting system easier. It was determined that initializing both k-means and GMM EM algorithms with the number of classes equal to the number of distinguishable neurons or one more than the number of distinguishable neurons yielded the minimum number of errors. Initializing the algorithms with one less class yielded marginally more errors, while initializing the sorting algorithms to two classes yielded the most errors. Thus, a sorting system implementing such an algorithm would have to determine the number of distinguishable neurons to get the best results. Also, it was shown that the GMM EM algorithm could achieve better results when initialized with appropriate seeds, however the k-means algorithm performed better on average.

References

Cheeseman P, Stutz J 1988 AutoClass: A Bayesian Classification system. Proc. 5th Int. Conf. On Machine Learning (San Francisco: Morgan Kaufmann):54-64.

Chandra R and Optican L M. Detection, classification, and superposition resolution of action-potentials in multiunit single-channel recordings by an online real-time neural-network IEEE Trans. Biomed. Eng. 1997; 44:403-12.

Chickering D M and Heckerman D. Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables. Machine Learning 1997;29:181-212.

Dempster, Laird N, and Rubin D. Maximum-likelihood from incomplete data via the EM algorithm. J. Royal Stat. Soc. 1977;39:1-38.

Detecting features in Spatial Point Process With Clutter via Model-Based Clustering. Journal of the American Statistical Association; Mar1998;93;441:294-303.

Jansen R F. The reconstruction of individual spike trains from extracellular multineuron recordings using a neural network emulation program. J Neurosci. Meth. 1990;35:203-13.

Harris KD, Henze DA, Csicsvari J, Hirase H, and Buzsaki G. Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. J. Neurophysio. 200;84:401-14.

Lewicki, M S. A review of methods for spike sorting; the detection and classification of neural action potentials. Network: Comput. Neural Sys. 1998; 9:R53-R78.

Lewicki M. Bayesian modeling and classification of neural signals. Neural Computation 1994; 6:1005-30.

Nicolelis M A L, Ghazanfar A A, Faggin B M, Votaw S and Oliveira L M. Reconstructing the engram: simultaneous, multisite, many single neuron recordings. Neuron 1997;18:529-37.

Obeid I, Wolf, P. Evaluation of Spike-Detection Algorithms for a Brain-Machine Interface Application. IEEE Transactions on Biomed. Eng 2004;51;905-11.

Reich D S, Mechler F, Victor J D. Independent and redundant information in nearby cortical neurons. Science 2001;294;2566-8.

Wheeler B C and Heetderks W J. A comparison of techniques for classification of multiple neural signals. IEE Trans Biomed. Eng. 1982;29:752-9.

Won D S, Wolf P D. A simulation study of information transmission by multi-unit microelectrode recordings. Neural Syst. 2003;14:1-16.

Wood F, Fellows, M, Donaghue J P, Black M J. Automatic Spike Sorting for Neural Decoding. Proceedings of IEEE EMBS 2004: 4009-12.

Appendix

A1. Mean results for 10 signals with high SNR signal.

	2 classes		3 classes		4 classes		5 classes	
	CC	IC	CC	IC	CC	IC	CC	IC
K-means	88.24%	11.76%	92.35%	7.65%	93.53%	6.47%	95.35%	4.65%
GMM EM	88.24%	11.76%	92.00%	8.00%	90.53%	9.47%	93.24%	6.76%
K-means	100.00%	0.00%	98.65%	1.35%	97.63%	2.37%	96.79%	3.21%
GMM EM	91.35%	8.65%	100.00%	0.00%	99.04%	0.96%	97.69%	2.31%
K-means	83.33%	16.67%	99.52%	0.48%	97.80%	2.20%	95.65%	4.35%
GMM EM	83.81%	16.19%	97.20%	2.80%	95.77%	4.23%	92.38%	7.62%
K-means	73.03%	26.97%	98.88%	1.12%	98.93%	1.07%	98.26%	1.74%
GMM EM	82.08%	17.92%	89.66%	10.34%	95.79%	4.21%	96.63%	3.37%
K-means	53.53%	46.47%	98.82%	1.18%	98.47%	1.53%	98.41%	1.59%
GMM EM	52.59%	47.41%	93.65%	6.35%	96.65%	3.35%	97.65%	2.35%
K-means	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%	95.32%	4.68%
GMM EM	80.91%	19.09%	92.34%	7.66%	95.84%	4.16%	95.78%	4.22%
K-means	76.15%	23.85%	89.67%	10.33%	98.79%	1.21%	98.30%	1.70%
GMM EM	73.13%	26.87%	87.20%	12.80%	97.03%	2.97%	95.71%	4.29%
K-means	72.73%	27.27%	98.86%	1.14%	99.83%	0.17%	98.86%	1.14%
GMM EM	70.23%	29.77%	93.81%	6.19%	97.44%	2.56%	96.48%	3.52%
K-means	100.00%	0.00%	99.00%	1.00%	99.00%	1.00%	98.40%	1.60%
GMM EM	89.80%	10.20%	99.70%	0.30%	99.30%	0.70%	98.70%	1.30%
K-means	83.76%	16.24%	100.00%	0.00%	97.82%	2.18%	96.88%	3.12%
GMM EM	87.65%	12.35%	96.29%	3.71%	97.94%	2.06%	96.29%	3.71%
Average								
K-means	83.08%	16.92%	97.58%	2.43%	98.18%	1.82%	97.22%	2.78%
GMM EM	79.98%	20.02%	94.19%	5.82%	96.53%	3.47%	96.06%	3.95%

A2. Mean results for 10 signals with medium SNR signal.

	2 classes		3 classes		4 classes		5 classes	
	CC	IC	CC	IC	CC	IC	CC	IC
K-means	75.64%	24.36%	100.00%	0.00%	99.10%	0.90%	97.37%	2.63%
GMM EM	91.67%	8.33%	94.81%	5.19%	95.83%	4.17%	95.83%	4.17%
K-means	75.00%	25.00%	93.00%	7.00%	97.38%	2.63%	95.63%	4.38%
GMM EM	69.75%	30.25%	86.75%	13.25%	96.50%	3.50%	97.19%	2.81%
K-means	100.00%	0.00%	99.88%	0.12%	99.52%	0.48%	99.40%	0.60%
GMM EM	92.77%	7.23%	99.76%	0.24%	99.64%	0.36%	99.64%	0.36%
K-means	79.41%	20.59%	79.41%	20.59%	88.68%	11.32%	89.04%	10.96%
GMM EM	68.16%	31.84%	80.07%	19.93%	85.07%	14.93%	82.65%	17.35%
K-means	68.20%	31.80%	83.72%	16.28%	91.86%	8.14%	90.93%	9.07%
GMM EM	65.35%	34.65%	87.15%	12.85%	86.57%	13.43%	89.65%	10.35%
K-means	54.05%	45.95%	65.54%	34.46%	74.66%	25.34%	84.59%	15.41%
GMM EM	54.39%	45.61%	63.99%	36.01%	75.07%	24.93%	72.70%	27.30%
K-means	69.83%	30.17%	81.11%	18.89%	89.28%	10.72%	96.06%	3.94%
GMM EM	67.72%	32.28%	90.22%	9.78%	91.61%	8.39%	93.39%	6.61%
K-means	77.47%	22.53%	97.68%	2.32%	95.20%	4.80%	92.02%	7.98%
GMM EM	87.58%	12.42%	97.07%	2.93%	95.30%	4.70%	92.32%	7.68%
K-means	81.05%	18.95%	88.21%	11.79%	93.05%	6.95%	93.47%	6.53%
GMM EM	80.00%	20.00%	84.63%	15.37%	88.74%	11.26%	88.68%	11.32%
K-means	67.80%	32.20%	100.00%	0.00%	97.68%	2.32%	97.56%	2.44%
GMM EM	68.11%	31.89%	94.94%	5.06%	97.20%	2.80%	96.65%	3.35%
Average								
K-means	74.85%	25.16%	88.86%	11.15%	92.64%	7.36%	93.61%	6.39%
GMM EM	74.55%	25.45%	87.94%	12.06%	91.15%	8.85%	90.87%	9.13%

A3. Mean results for 10 signals with low SNR signal.

	2 classes		3 classes		4 classes		5 classes	
	CC	IC	CC	IC	CC	IC	CC	IC
K-means	92.59%	7.41%	94.51%	5.49%	91.23%	8.77%	89.44%	10.56%
GMM EM	83.95%	16.05%	86.91%	13.09%	87.72%	12.28%	85.99%	14.01%
K-means	44.19%	55.81%	66.28%	33.72%	87.79%	12.21%	93.49%	6.51%
GMM EM	54.65%	45.35%	73.78%	26.22%	81.69%	18.31%	85.12%	14.88%
K-means	73.58%	26.42%	84.72%	15.28%	92.92%	7.08%	90.71%	9.29%
GMM EM	69.95%	30.05%	82.41%	17.59%	89.25%	10.75%	86.70%	13.30%
K-means	48.86%	51.14%	71.52%	28.48%	84.24%	15.76%	88.80%	11.20%
GMM EM	58.16%	41.84%	67.66%	32.34%	75.57%	24.43%	82.47%	17.53%
K-means	67.06%	32.94%	76.41%	23.59%	86.41%	13.59%	84.29%	15.71%
GMM EM	63.18%	36.82%	72.59%	27.41%	74.18%	25.82%	79.76%	20.24%
K-means	64.52%	35.48%	79.57%	20.43%	84.68%	15.32%	86.29%	13.71%
GMM EM	65.75%	34.25%	82.26%	17.74%	84.41%	15.59%	87.26%	12.74%
K-means	98.75%	1.25%	97.50%	2.50%	92.63%	7.37%	91.38%	8.62%
GMM EM	98.25%	1.75%	97.31%	2.69%	95.63%	4.37%	93.38%	6.62%
K-means	72.69%	27.31%	95.00%	5.00%	96.81%	3.19%	97.31%	2.69%
GMM EM	75.88%	24.13%	95.94%	4.06%	96.81%	3.19%	95.56%	4.44%
K-means	81.59%	18.41%	97.73%	2.27%	93.81%	6.19%	89.20%	10.80%
GMM EM	80.51%	19.49%	88.47%	11.53%	88.41%	11.59%	85.06%	14.94%
K-means	98.59%	1.41%	98.80%	1.20%	94.30%	5.70%	92.25%	7.75%
GMM EM	99.65%	0.35%	98.45%	1.55%	96.83%	3.17%	94.37%	5.63%
Average								
K-means	74.24%	25.76%	86.20%	13.80%	90.48%	9.52%	90.32%	9.68%
GMM EM	74.99%	25.01%	84.58%	15.42%	87.05%	12.95%	87.57%	12.43%

A4. Minimum error results for 10 signals with high SNR signal.

	2 classes		3 classes		4 classes		5 classes	
	CC	IC	CC	IC	CC	IC	CC	IC
K-means	88.24%	11.76%	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%
GMM EM	88.24%	11.76%	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%
K-means	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%
GMM EM	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%
K-means	95.24%	4.76%	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%
GMM EM	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%
K-means	73.03%	26.97%	98.88%	1.12%	100.00%	0.00%	100.00%	0.00%
GMM EM	98.88%	1.12%	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%
K-means	65.88%	34.12%	98.82%	1.18%	98.82%	1.18%	98.82%	1.18%
GMM EM	69.41%	30.59%	98.82%	1.18%	98.82%	1.18%	98.82%	1.18%
K-means	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%
GMM EM	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%
K-means	82.42%	17.58%	94.51%	5.49%	100.00%	0.00%	100.00%	0.00%
GMM EM	98.90%	1.10%	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%
K-means	72.73%	27.27%	98.86%	1.14%	100.00%	0.00%	100.00%	0.00%
GMM EM	98.86%	1.14%	98.86%	1.14%	98.86%	1.14%	98.86%	1.14%
K-means	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%
GMM EM	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%
K-means	95.29%	4.71%	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%
GMM EM	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%	100.00%	0.00%
Average								
K-means	87.28%	12.72%	99.11%	0.89%	99.88%	0.12%	99.88%	0.12%
GMM EM	95.43%	4.57%	99.77%	0.23%	99.77%	0.23%	99.77%	0.23%

A5. Minimum error results for 10 signals with medium SNR signal.

	2 classes		3 classes		4 classes		5 classes	
	CC	IC	CC	IC	CC	IC	CC	IC
K-means	75.64%	24.36%	100.00%	0.00%	99.10%	0.90%	97.37%	2.63%
GMM EM	91.67%	8.33%	94.81%	5.19%	95.83%	4.17%	95.83%	4.17%
K-means	75.00%	25.00%	93.00%	7.00%	97.38%	2.63%	95.63%	4.38%
GMM EM	69.75%	30.25%	86.75%	13.25%	96.50%	3.50%	97.19%	2.81%
K-means	100.00%	0.00%	99.88%	0.12%	99.52%	0.48%	99.40%	0.60%
GMM EM	92.77%	7.23%	99.76%	0.24%	99.64%	0.36%	99.64%	0.36%
K-means	79.41%	20.59%	79.41%	20.59%	88.68%	11.32%	89.04%	10.96%
GMM EM	68.16%	31.84%	80.07%	19.93%	85.07%	14.93%	82.65%	17.35%
K-means	68.20%	31.80%	83.72%	16.28%	91.86%	8.14%	90.93%	9.07%
GMM EM	65.35%	34.65%	87.15%	12.85%	86.57%	13.43%	89.65%	10.35%
K-means	54.05%	45.95%	65.54%	34.46%	74.66%	25.34%	84.59%	15.41%
GMM EM	54.39%	45.61%	63.99%	36.01%	75.07%	24.93%	72.70%	27.30%
K-means	69.83%	30.17%	81.11%	18.89%	89.28%	10.72%	96.06%	3.94%
GMM EM	67.72%	32.28%	90.22%	9.78%	91.61%	8.39%	93.39%	6.61%
K-means	77.47%	22.53%	97.68%	2.32%	95.20%	4.80%	92.02%	7.98%
GMM EM	87.58%	12.42%	97.07%	2.93%	95.30%	4.70%	92.32%	7.68%
K-means	81.05%	18.95%	88.21%	11.79%	93.05%	6.95%	93.47%	6.53%
GMM EM	80.00%	20.00%	84.63%	15.37%	88.74%	11.26%	88.68%	11.32%
K-means	67.80%	32.20%	100.00%	0.00%	97.68%	2.32%	97.56%	2.44%
GMM EM	68.11%	31.89%	94.94%	5.06%	97.20%	2.80%	96.65%	3.35%
Average								
K-means	74.85%	25.16%	88.86%	11.15%	92.64%	7.36%	93.61%	6.39%
GMM EM	74.55%	25.45%	87.94%	12.06%	91.15%	8.85%	90.87%	9.13%

A6. Minimum error results for 10 signals with low SNR signal.

	2 classes		3 classes		4 classes		5 classes	
	CC	IC	CC	IC	CC	IC	CC	IC
K-means	92.59%	7.41%	94.51%	5.49%	91.23%	8.77%	89.44%	10.56%
GMM EM	83.95%	16.05%	86.91%	13.09%	87.72%	12.28%	85.99%	14.01%
K-means	44.19%	55.81%	66.28%	33.72%	87.79%	12.21%	93.49%	6.51%
GMM EM	54.65%	45.35%	73.78%	26.22%	81.69%	18.31%	85.12%	14.88%
K-means	73.58%	26.42%	84.72%	15.28%	92.92%	7.08%	90.71%	9.29%
GMM EM	69.95%	30.05%	82.41%	17.59%	89.25%	10.75%	86.70%	13.30%
K-means	48.86%	51.14%	71.52%	28.48%	84.24%	15.76%	88.80%	11.20%
GMM EM	58.16%	41.84%	67.66%	32.34%	75.57%	24.43%	82.47%	17.53%
K-means	67.06%	32.94%	76.41%	23.59%	86.41%	13.59%	84.29%	15.71%
GMM EM	63.18%	36.82%	72.59%	27.41%	74.18%	25.82%	79.76%	20.24%
K-means	64.52%	35.48%	79.57%	20.43%	84.68%	15.32%	86.29%	13.71%
GMM EM	65.75%	34.25%	82.26%	17.74%	84.41%	15.59%	87.26%	12.74%
K-means	98.75%	1.25%	97.50%	2.50%	92.63%	7.37%	91.38%	8.62%
GMM EM	98.25%	1.75%	97.31%	2.69%	95.63%	4.37%	93.38%	6.62%
K-means	72.69%	27.31%	95.00%	5.00%	96.81%	3.19%	97.31%	2.69%
GMM EM	75.88%	24.13%	95.94%	4.06%	96.81%	3.19%	95.56%	4.44%
K-means	81.59%	18.41%	97.73%	2.27%	93.81%	6.19%	89.20%	10.80%
GMM EM	80.51%	19.49%	88.47%	11.53%	88.41%	11.59%	85.06%	14.94%
K-means	98.59%	1.41%	98.80%	1.20%	94.30%	5.70%	92.25%	7.75%
GMM EM	99.65%	0.35%	98.45%	1.55%	96.83%	3.17%	94.37%	5.63%
Average								
K-means	74.24%	25.76%	86.20%	13.80%	90.48%	9.52%	90.32%	9.68%
GMM EM	74.99%	25.01%	84.58%	15.42%	87.05%	12.95%	87.57%	12.43%