

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

ABSTRACT

ANALYSIS OF CLUSTERING ALGORITHMS FOR SPIKE SORTING OF MULTIUNIT EXTRACELLULAR RECORDINGS

by
Jayesh S Rege

Various techniques have been considered in the past to identify distinct spike shapes from multiunit extracellular recording. These techniques involve adaptive filtering techniques or template matching techniques or hierarchical clustering techniques.

In this investigation, we have used Principal Component Analysis followed by various clustering techniques to identify distinct spike shapes. The amplitude filter is used to separate spikes from background neuronal activity. The correlation matrix of the spike data is used to compute principal component wave forms. Each spike is thus represented by the coefficients of principal components. Then, We have used agglomerative hierarchical clustering algorithm to perform the initial clustering of the data set. The clustering results are then refined by the application of the Estimation Maximization Algorithm. The Bayesian Information Criteria(BIC) is used to find out best fit of the model to the data set.

**ANALYSIS OF CLUSTERING ALGORITHMS FOR SPIKE SORTING OF
MULTIUNIT EXTRACELLULAR RECORDINGS**

by
Jayesh S Rege

**A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science**

Department of Computer and Information Science

January 2000

Blank Page

APPROVAL PAGE

**ANALYSIS OF CLUSTERING ALGORITHMS FOR SPIKE SORTING OF
MULTIUNIT EXTRACELLULAR RECORDINGS**

Jayesh S Rege

Dr. Michael Recce, Thesis Advisor Date
Associate Professor of Computer and Information Science, NJIT

Dr. James A. M. McHugh, Committee Member Date
Professor of Computer and Information Science, NJIT

Dr. Farzān Nadim, Committee Member Date
Assistant Professor of Mathematics, NJIT

BIOGRAPHICAL SKETCH

Author: Jayesh S Rege
Degree: Master of Science in Computer and Information Science
Date: January 2000

Undergraduate and Graduate Education:

- Master of Science in Computer and Information Science
New Jersey Institute of Technology, Newark, NJ, 2000
- Bachelor of Engineering in Computer Engineering,
University of Mumbai, Mumbai, India, 1995

Major: Computer and Information Science

To my parents

ACKNOWLEDGMENT

I would like to express my deepest appreciation to Dr. Michael Recce, who not only served as my research advisor, providing valuable and countless resources, but also constantly gave me support, encouragement, and reassurance. I am very thankful to Dr. Kenneth Harris for provided me with initial data set for analysis and also for giving valuable guidance throughout this research. I also wish to thank Dr. Farzan Nadim for answering many of my questions related to neuroscience. Special thanks are also given to Dr. James McHugh and Dr. Farzan Nadim for actively participating in my review committee.

TABLE OF CONTENTS

Chapter	Page
1. INTRODUCTION	1
1.1 Objective	1
1.2 Background Information	2
2. ORIGIN OF THE ACTION POTENTIAL	4
2.1 Why Neurons Produce Action Potentials?	4
2.2 Data Collection	5
2.3 Spike Detection and Types of Detection Errors	7
3. FEATURE ANALYSIS	12
3.1 Spike Shape Characteristics	12
3.2 Principal Component Analysis	13
4. CLUSTERING ALGORITHMS	19
4.1 Hierarchical Clustering	20
4.1.1 Removal of Outliers	22
4.1.2 Agglomerative Clustering	23
4.2 Estimation Maximization algorithm	26
4.2.1 Stochastic Processes	27
4.2.1.1 Multivariate Gaussian Process	27
4.2.1.2 Poisson Process	29

TABLE OF CONTENTS
(Continued)

Chapter	Page
4.2.2 Application of EM algorithm to Multivariate Gaussian Mixture	29
4.4.2.1 Estimation Step.	31
4.4.2.2 Maximization Step	32
4.4.2.3 Estimation Classification Maximization(ECM) Algorithm	33
4.4.2.4 Modifications to the EM Algorithm	34
4.2.3 How Many Clusters?	36
4.2.4 Disadvantages of EM Algorithm	38
5. EXPERIMENTAL RESULTS	37
REFERENCES	43

LIST OF FIGURES

Figure	Page
2.1 Action potential measured at various electrodes of tetrode.	7
2.2 a) Snapshot of original data wave form collected on one of the electrodes of the tetrode. b) spikes identified from data.	8
2.3 Use of amplitude filter to identify spikes	10
3.1 Principal component analysis of spike shape	18
4.1 Sample spike clusters	21
4.2 Distance of points from their nearest neighbor	22
4.3 Histogram showing the distribution of points in the cluster	25
4.4 Flow-chart of hierarchical clustering algorithm	26
4.5 BIC Vs. number Of Clusters	37
4.6 Interspike histogram	37
5.1 PCA Output of the input data	40
5.2 Output of the hierarchical clustering algorithm	41
5.3 Output of the clustering algorithm on various principal component axis . .	42
5.4 Improvement in the clustering by application of EM algorithm	43
5.5 Representative spike wave forms for 9 clusters identified in data set	44

CHAPTER 1

INTRODUCTION

1.1 Objective

The objective of this thesis is to analyze various clustering techniques for spike sorting of neuronal data. Over the years many techniques have been developed to perform the spike sorting of single unit recording. We wish to extend these algorithms and apply them to multiunit recording.

Our nervous system contains millions of interconnected neurons. Each neuron consists of long tapering structure called axon that originates from its cell membrane. These axons then terminate in to a mesh or tree of 'dendrites' which in turn are connected to cell membrane of the next neuron. Each neuron acts like an integrator of the nerve impulse received at its dendrites and produces an output at its axons. The nerve impulse from one neuron to another consists of electric signal in the form of a voltage spike. This neuronal activity can be collected by using the multielectrode recording from the brain. Since the recording is done using extracellular electrodes, the recording contains noise as well as the overlapped traces of the individual neuronal activity. We wish to apply various clustering techniques to separate and identify the activity of individual neurons. We will also use Principal Component Analysis to perform the feature analysis. The statistical analysis is done using the hierarchical clustering algorithm and the estimation maximization (EM) algorithm.

1.2 Background Information

Advances in neurophysiology have enabled us to get a glimpse of the working of the brain from the study of neurons. We now know that neurons interact with each other by means of bursts of voltage spikes which is known as action potential. If we can understand this language of neurons we might be able to understand the behavior of the nervous system as a whole. Until a few years ago it was thought impossible to understand the overall working of the brain from studying single neurons. However many new discoveries in recent years in the field of neurophysiology have confirmed the fact that the understanding of the working of the basic building block of the nervous system i.e. the 'single neuron' is essential to the understanding of the overall working of the brain.

It is believed that if we can accurately measure and understand the electrical activity amongst the neuronal population it will greatly enhance our understanding of the brain. Our brain is one of the most complex and fastest distributed computing machines ever designed. This is possible because of the ability of our brain to perform fast parallel computation and encoding of a distributed representation of the data. Although the activity mechanism of each individual neuron may be simple, it is the temporal activity of multiple neurons in the neuronal circuitry that gives rise to immense robustness and speed of our nervous system. Hence, in recent years considerable attention has been focused on the recording and analyzing of this neuronal activity. It is hoped that such understanding will provide much needed insight into how various neurons interact amongst each other. In order to correlate the activities of multiple neurons we need to collect the firing patterns from multiple

neurons and study their relationships. We can measure the change in voltage in an area surrounding the neuron by placing an electrode near the cell. Such recording will contain the sum total of signals from the activity of various neurons. We need to separate the individual spikes and assign them to individual neurons to identify the global pattern of the behavior.

CHAPTER 2

ORIGIN OF THE ACTION POTENTIAL

In this chapter we will look at the physiology of how and why neurons produce action potentials. We will review various techniques for measuring electrical activity from cells. We will also describe the data that we are using in these experiments. In the final section we will enumerate various sources of error that can occur during measurement of the action potential.

2.1 Why Do Neurons Produce Action Potentials?

Understanding how neurons produce action potentials was a major research problem until Hodgkin and Huxley described the concept of gated ionic channels in 1952. The cell membrane of the neuron consists of a bilayer of phospholipid molecules. This lipid bilayer effectively isolates the cytoplasm of the neuron from its surroundings. The cell membrane also consists of peculiar arrangements of the protein molecules across its surface. These protein molecules create ionic channels that span the cell membrane. Ionic channels effectively allow only certain ions to pass through them while blocking the movement of other types of ions. Thus sodium channels will allow the movement of only Na^+ ions and block K^+ ions. These channels are also gated in the sense that they open and close depending on the electric potential gradient across the cell membrane. At equilibrium, the cell has a large concentration of K^+ inside, while the surroundings of the cell includes a large concentration of Na^+ ions. However, the diffusion gradient is offset by the electrical field across the cell

membrane and, therefore, there is no net current across this membrane. The potential difference between the inside and outside of the cell membrane is close to the ionic equilibrium potential of K^+ which is around -70 mV.

During the action potential the sodium-gated channels open and cause the influx of Na^+ ions from the surroundings of the neuron. This makes inside of the cell positive and takes it to the ionic equilibrium potential of the Na^+ ions. This process is known as the depolarization of the cell. Thus, the rising edge of the action potential is caused by the depolarization of the cell. After the initial influx of Na^+ ions, the Na^+ channels start closing and at the same time the K^+ channels start opening. This causes the efflux of the K^+ ions. This process is known as the hyperpolarization of cell and will take the cell back to its resting potential. In a recording this will be seen as falling edge of the action potential. Due to the gating mechanism it is difficult for a neuron to produce the next action potential before a certain interval of the time. This is called as the refractory period and it has a duration of about 2 ms.

2.2 Data Collection

Initial attempts at recording neuronal activity were done using single electrode.(Abeles and Goldstein.1977). The experimenter would place an electrode amongst the nerve cells or fibers being studied. The electrode has a low resistance and is capable of measuring micro-voltage changes in the potential of the surrounding environment. The electrical activity recorded consists of nerve spikes with a duration of 1 ms and peak-to-peak amplitude of less than 1 mV. The measurement also includes embedded background noise. Background noise is a white noise introduced

by the measuring instrument and voltage variations due to background activity of the neuronal population. If the signal to noise ratio of the waveform is good enough, then the waveform could be converted into series of spikes by using a threshold detector. The threshold detector could be a simple shmitt trigger, which can be set at a particular threshold voltage.

However, since cells in hippocampus fire action potential in rapid bursts the shape and amplitude of these spikes can change to a certain extent. Such changes in spike shape can lead to identification of 2 or more units from the activity of a single cell.

In order to overcome such problems, multiple tip electrode recording techniques such as tetrode was developed. The tetrode(Recce and O'Keefe 1989) consist of twisted bundles of four strands of wire electrodes. The tips of the electrodes are slightly displaced by about $20 \mu\text{m}$. The lower end of the wire strands is cut at a slight angle with sharp scissors to form a tip with four exposed surfaces. The upper ends of the electrodes are electrically attached to measuring interments. Theoretically, tetrode can uniquely identify the anatomical position of the neuron in space.

In the case of a tetrode even if there is a slight change in spike waveform on the 1st electrode, the corresponding spike on the 2nd electrode also changes. This however still maintains the relative proportionality of the spike waveforms. It has also been found experimentally [6] that the tetrode improves the number of neuronal units identified from the waveform. Figure 2.1 shows sample voltage waveform measured across various electrodes of a tetrode.

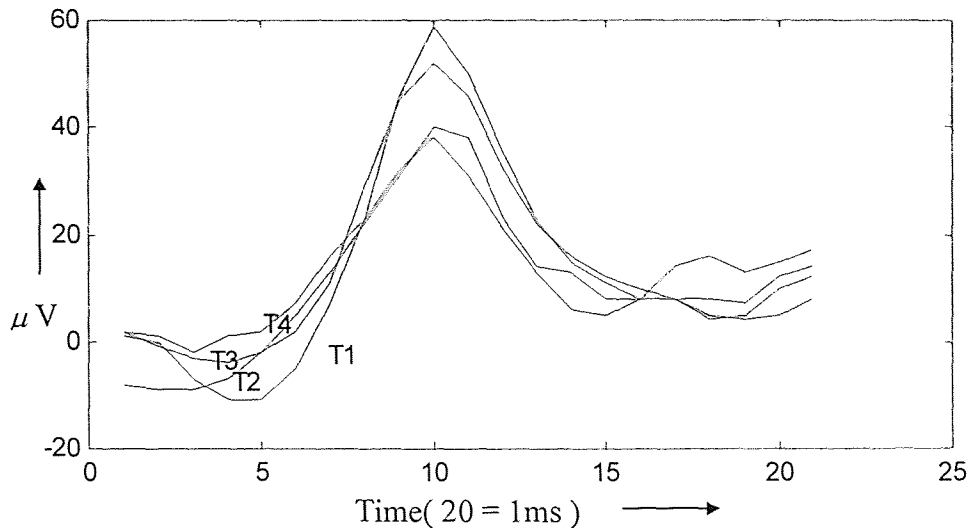


Figure 2.1 Action potential measured at various electrodes of tetraode.

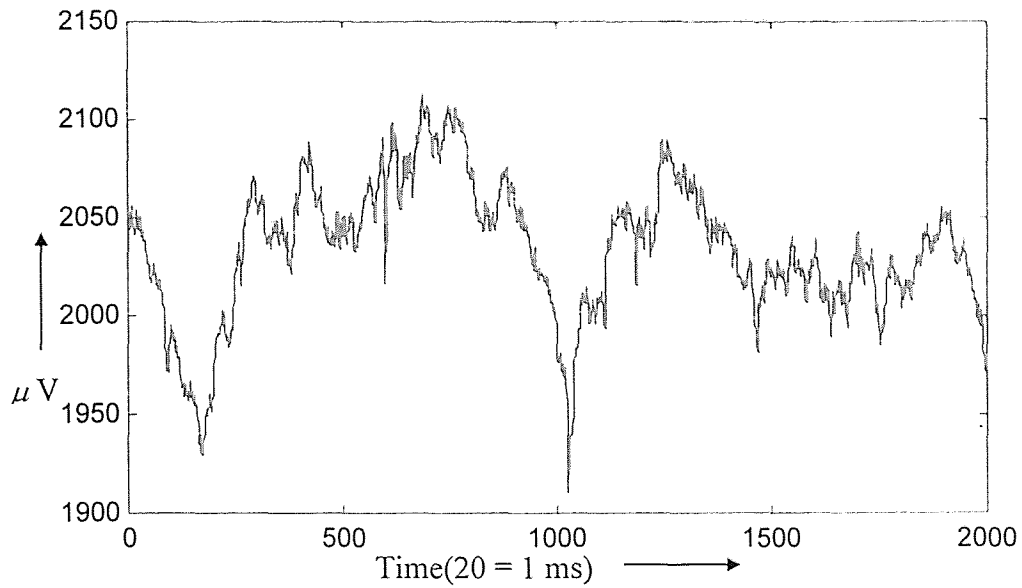
2.3 Spike Detection and Types of Detection Errors

The first step in the spike detection is separating spikes from the data collected at the tetraode. Consider the figure 2.2(a), which shows a general snapshot of the signal collected at one of the electrodes of a tetraode. We need to separate spikes from the background activity of neurons.

Figure 2.2 shows the original waveform and identified spikes from the data.

We define occurrence of a spike when voltage at any of the electrodes crosses a fixed threshold value. Thus it is inherent in our argument that an estimate of presence of spike activity is susceptible to variations in this threshold value. The number of detected spikes will increase if we lower the threshold value but that will increase the number of false alarms. A false alarm occurs when we identify a spike when it is actually noise. On the other hand if we increase this threshold value we might reduce the number of false alarms but it will increase the probability of a false dismissal. A false dismissal occurs when we classify a valid spike as noise.

a)]



b)]

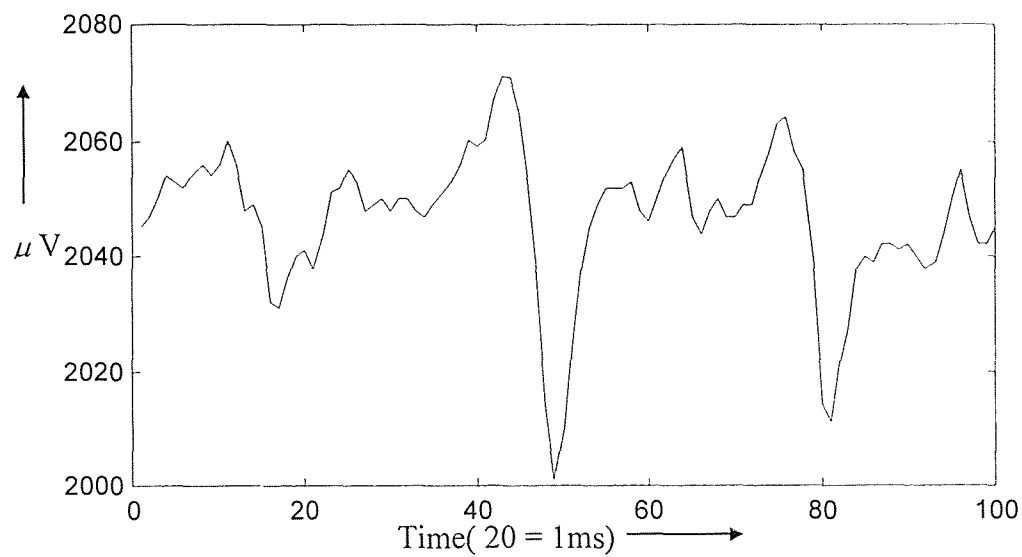


Figure 2.2 a) Snapshot of original data waveform collected on one of the electrodes of the tetrode. b) spikes identified from data.

Other sources of errors in detection of spike activity include:

1. Cross-electrode stray capacitance.
2. Variations in the ground potential.
3. Background activity of the neuronal population.
4. Overlapping of the spikes from more than one neuron.
5. Drift in the anatomical position of the electrode.

The capacitance of electrode tips to solution is several times bigger than the cross-electrode capacitance. Hence we can assume that the cross-electrode stray capacitance is very low. It has been found that the coefficient of correlation across the channels of two tetrodes placed near each other is very low[6]. However the coefficient of correlation across the channels of same tetrode is very high. Hence we can assume that the noise introduced by variations of the ground potential is very small. The remaining sources of errors will result in noise outliers and will be removed by the clustering algorithm.

One of the simplest methods for spike identification is amplitude filter. In its simplest form an amplitude filter consists of a Schmitt trigger and produces a spike if the maximum amplitude of the spike lies between the maximum and minimum of the filter window. We can modify this simplest Schmitt trigger circuitry to take into account the waveform on all four of the electrodes. In case of a tetrode, an amplitude filter finds a spike if the spike amplitude crosses the threshold at any one of the four electrodes. The threshold selected could be characteristic of the background process or it could also be set by the user. This simple scheme gives rise to hypercubical decision surface. However due to cross channel correlation, the noise spikes are also

correlated across various channels of the tetrode. Hence more suitable approach[15] could be to use a hyperellipsoidal decision surface to identify spikes.

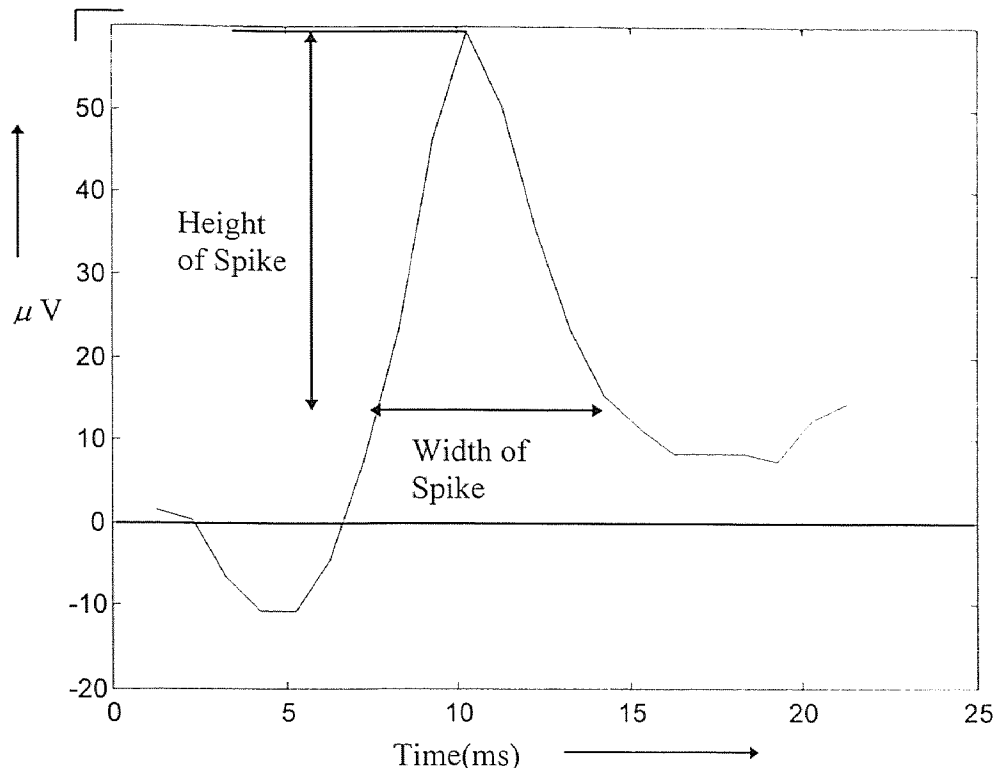


Figure 2.3 Use of amplitude filter to identify spikes. θ_H = spike height and θ_w is Spike width.

A limitation to the performance of the amplitude filter is that, in most experimental situations, the spike being filtered is immersed in noise and this will cause peak amplitude to fluctuate from spike to spike. Also occasionally noise amplitude will cross the threshold for spike detection and will result in a false spike detection. The amplitude filter is also susceptible to small movements of electrodes or to slight changes in the spike generating mechanism of neurons. Also since amplitude filter is

checking for height of the spike, it is highly sensitive to noise at this particular instant of time rather than noise spread over the total duration of spike.

CHAPTER 3

FEATURE ANALYSIS

The process of classification of spikes from the recorded signal mainly involves following phases.

1. Detection of the spikes from recording.
2. Feature Extraction: Estimation of shapes of various spike wave forms.
3. Feature Analysis(Clustering): Establishment of decision criteria for classification, based upon the differences in shapes of spikes. Testing of each spike to identify the neuronal group to which the spike belongs.

In chapter 2, we considered the problem of the detection of spikes from measured voltage wave form. In this chapter we will consider the problem of feature analysis and various techniques to represent spike data. During feature analysis we will represent each spike shape to facilitate clustering of the spike shapes in the next phase.

3.1 Spike Shape Characteristics

The goal of feature extraction process is to reduce original data by measuring certain properties of data that can be useful in the final classification of data. Over the years various attempts to classify spike wave form have used characteristics of shape of the spike wave form. It is assumed that the spikes from different neurons will have different spike shapes. We can use this variability of spike shape from one neuron to

other to classify various spikes. The question now arises is how do we compare two spike shapes?

One approach is to measure the spike height or the maximum amplitude of spike wave form. However the amplitude of the spike wave form is susceptible to fluctuations due to background activity as well as due to noise introduced by measuring instrument. Some of the other characteristics of the spike shape that can be considered are width of the spike or the slope of the spike wave form. However all these features are susceptible to noise and are indirectly dependent on each other. The positive overlap of spikes from two or more neurons will distort the resultant spike measured at electrodes and makes identification of individual spike shapes even more difficult. Hence we need some kind of additional independent features in order to enrich our feature vector.

The other approach is using Principal Component Analysis to represent each spike shape. We will discuss PCA and its advantages in the following section.

3.2 Principle Component Analysis

It is clear from above discussion that in order to maximize our chances of classification we need to use full spike wave form characteristics. However processing of full wave form characteristics increases computational complexity and memory requirements.

One of the more popular ways to reduce wave form is by using the fourier transformation. Processing of spike wave form in frequency domain results in a lot of high frequency components with low amplitude[1]. Hence, number of coefficients

required to represent original wave form will exceed number of data points. This will anyway make it difficult to differentiate individual spike shapes from each other. Hence we need to look at alternative techniques to represent spike wave form which will reduce the number of coefficients and at the same time will represent the signal to a high degree of accuracy. Here we consider Principal Component Analysis technique for dimension reduction of the original signal.

Consider the linear representation of set of N data wave forms $S_n(t)$ where n ranges from 1 to N .

$$S_n(t) = \sum_{m=1}^M c_{nm} f_m(t).$$

Each $f_m(T)$ is an individual basic wave form and c_{nm} is the coefficient of individual wave form. There are M basic wave forms and $f_m(T)$ represents m^{th} one of them. Each of these basic wave forms are orthonormal to all other basic wave forms. Using these basic wave forms we can represent the original signal by using $m \times n$ coefficients as opposed to $n \times n$ data points.

We need to determine these basic wave forms and the corresponding coefficients in such a way that the residual least-mean-square error is least. Our goal is to find series of orthonormal components such that each component is the best fit for the residual signal. Thus the first basic wave f_{1t} is the least mean square error fit of a single wave form to the entire data set. The second basic wave is orthonormal to the first one and is least-mean-square fit to the residual from the fit of the first wave. The process is continued until the $S_i(t)$ data signal space is specified by the set of basic wave forms to a sufficient degree of accuracy.

In order to compute the Principle Components of the data we use correlation matrix R of the signal. Let S be the signal matrix of N signals each with T dimensions. Let C be the PCA coefficients. Then

$$S = C F \quad [3.1]$$

where C is the N by M matrix of the PCA coefficients.

F is the M by T matrix and represents the basic orthonormal wave forms.

If R is the correlation matrix of the signal then R can be represented as

$$R - \lambda I = 0 \quad [3.2]$$

where, λ is diagonal matrix of eigenvalues of R .

$$\lambda = \begin{bmatrix} \lambda_1 & 0 & \dots\dots\dots & 0 \\ 0 & \lambda_2 & 0 & \dots\dots\dots \\ 0 & \dots\dots\dots & \dots\dots\dots & \dots\dots\dots \\ 0 & \dots\dots\dots & \dots\dots\dots & \dots\dots\dots \\ 0 & 0 & \dots\dots\dots & \lambda_N \end{bmatrix}$$

The set of the solutions $(\lambda_1, \dots, \lambda_N)$ of the above equation represent the eigenvalues of matrix R . These eigenvalues can be real or imaginary numbers. Substituting eigenvalues back in the equation [3.2] we will get a set of eigenvectors (U_1, \dots, U_N) . Each eigenvector is of dimension $N \times 1$.

The eigenvectors and eigenvalues are related to their parent correlation matrix in the following way:

$$R U_m = \lambda_m U_m$$

These N eigenvectors taken together will form $N \times N$ an orthogonal matrix U called as eigenmatrix of the R . Thus,

$$R U = U \lambda \quad [3.3]$$

Since eigenvectors are orthonormal to each other,

$$UU^T = I \quad [3.4]$$

Postmultiplying equation [3.3] by U^T and using equation [3.4] we get

$$\begin{aligned} R &= U \lambda U^T \\ R &= (U\sqrt{\lambda})(\sqrt{\lambda}U^T) \\ R &= (U\sqrt{\lambda})(U\sqrt{\lambda})^T \text{ as } \lambda = \lambda^T \end{aligned} \quad [3.5]$$

but since R is correlation matrix, R is also equal to

$$R = S S^T \quad [3.6]$$

Using equation [3.1] we get

$$\begin{aligned} R &= (C F) (C F)^T \\ R &= C F F^T C^T \end{aligned} \quad [3.7]$$

Since F represents vectors which are orthonormal to each other $F F^T = I$,

$$R = C C^T \quad [3.8]$$

Comparing equations [3.8] and [3.5]

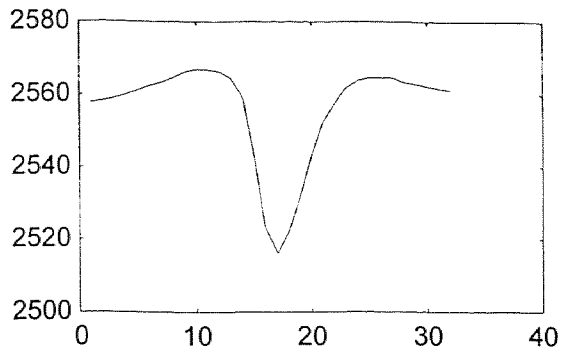
$$C = U \sqrt{\lambda}$$

The eigenvalue λ_m represents the contribution of m^{th} component of the basic wave form to the total power of the original signal. The optimal representation of the signal using M basic wave forms is obtained by using the first M eigenvalues of the correlation matrix R.

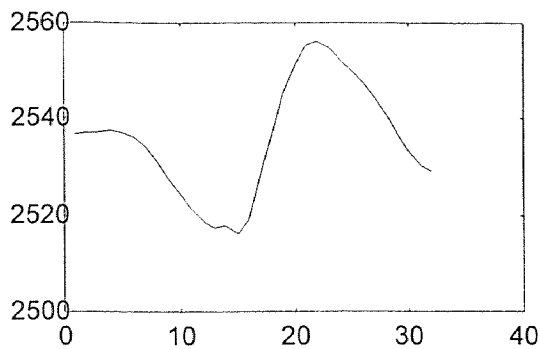
The advantage of processing data by using PCA is that it reduces the dimensions of the data and in turn reduces the degrees of freedom for the statistical processing of data. Another advantage of PCA is improvement in the signal to noise ratio of the

wave form. Let us assume that the background noise perturbs the signal to a small extent, so that original action potential from the neuron is still the dominant portion of the recorded spike. In this case, the contribution of noise will be higher in the lower principal components. Since we are taking only first M basic wave forms, in effect we are also removing the contribution of noise to the signal and will improve signal to noise ratio.

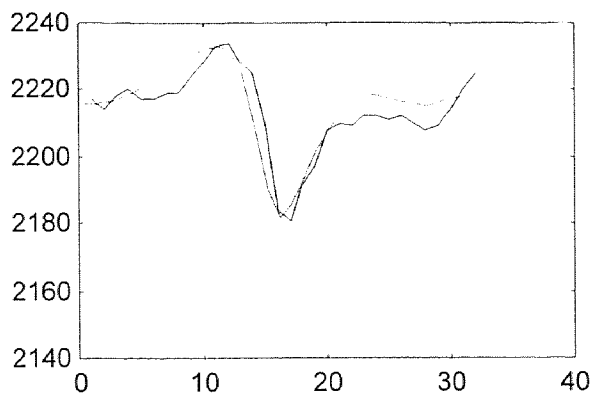
We are losing some characteristics of wave shape when we use principal component analysis as compared to template matching techniques. However reduction in the degrees of freedom and subsequent reduction in memory requirements for analysis, makes PCA an extremely attractive alternative. In our case each spike consists of 32 samples over 4 electrodes. We have applied PCA to the signal on each electrode and reduced the data set to 3 dimensions using the first 3 principal components. Each spike is now represented by a 12 dimensional vector. Figure 5.1 shows output of PCA analysis in two dimensions. We have used Xgobi software to visualize data in multiple dimensions.



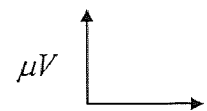
a) First basic wave form obtained by applying PCA to spikes on channel 1.



b) Second basic wave form obtained by applying PCA to spikes on channel 1.



c) The solid line indicates the original spike while the dash line indicates the approximate spike using principal components



Time (20=1ms)

Figure 3.1 Principal Component Analysis of Spike Shape.

CHAPTER 4

CLUSTERING ALGORITHMS

Clustering is finding groups in the data. We can easily identify clusters in two dimensions but it is very difficult to identify clusters when data consists of more than two dimensions. As described in earlier section we have reduced the data set to 12 dimensions using principal component analysis. However identifying exact clusters in so many dimensions is very difficult, hence we use clustering algorithms to perform automatic clustering of data. In the current chapter we will consider various clustering algorithms to perform automatic clustering of the data set.

Clustering algorithms can be divided in two categories, namely nonparametric and parametric clustering. In case of parametric clustering we assume that the data set follows certain statistical criteria and in turn we can successfully model the data as one of the well-known statistical distributions. The goal of parametric clustering algorithms is to identify parameters of the distribution of data. However, success of the parametric techniques depends upon the choice of the selected model and the fitness of data to the selected statistical distribution. The advantage of the parametric techniques is that we can use analytical and statistical tools to predict the behavior of the clustering algorithm. The EM algorithm, which we have used, falls in this category.

On the other hand nonparametric clustering techniques try to use properties of the data to identify classes of the data. Thus nonparametric clustering techniques rely on heuristics of the data or the process which has created data. The hierarchical

clustering and the K means clustering algorithms are examples of nonparametric techniques for clustering data.

4.1 Hierarchical Clustering

Hierarchical clustering algorithms is a collection of algorithms that use the dissimilarity distance between clusters to make decisions during clustering process. If we map properties of objects to coordinate space then each object will be represented by a point in this space. Since a cluster is a group of objects with similar properties, the region of high concentration of points in the coordinate space will represent a cluster of objects. The process of mapping properties of objects to a coordinate space is known as normalization. The number of axis in the resultant coordinate space is equal to the number of properties of the objects. Thus when we are representing each spike by 12 principal components, we are normalizing each spike to a 12 dimensional coordinate space represented by individual principal components. The position of the spike in this space will be determined by the coefficients of principal components required to represent the original spike. If we have two spikes of similar shape then they will be represented by two points, which are closer to each other in the coordinate space.

Next step is to determine the dissimilarity distance between the two clusters. The dissimilarity distance indicates how close two clusters are in the coordinate space. If two clusters are very close to each other then their dissimilarity distance is lower. There are many metrics(Hartigan(1974) and Kaufman(1990)) that can be used to identify the dissimilarity distance between two clusters. We have defined

dissimilarity distance as the sum of distances between the points of each cluster.

Consider two clusters R_1 and R_2 then dissimilarity distance D is given by

$$D(R_1, R_2) = \sum_{i=1}^n \sum_{j=1}^m d(R_{1i}, R_{2j})$$

The dissimilarity distance that we have chosen gives more weight to intercluster distance as well as the shape of the cluster. As seen in Figure 4.1 we find that in case of spike clusters, more number of points are located near the surface of clusters. Also these two clusters almost overlap. Hence we feel that using the above metric we are

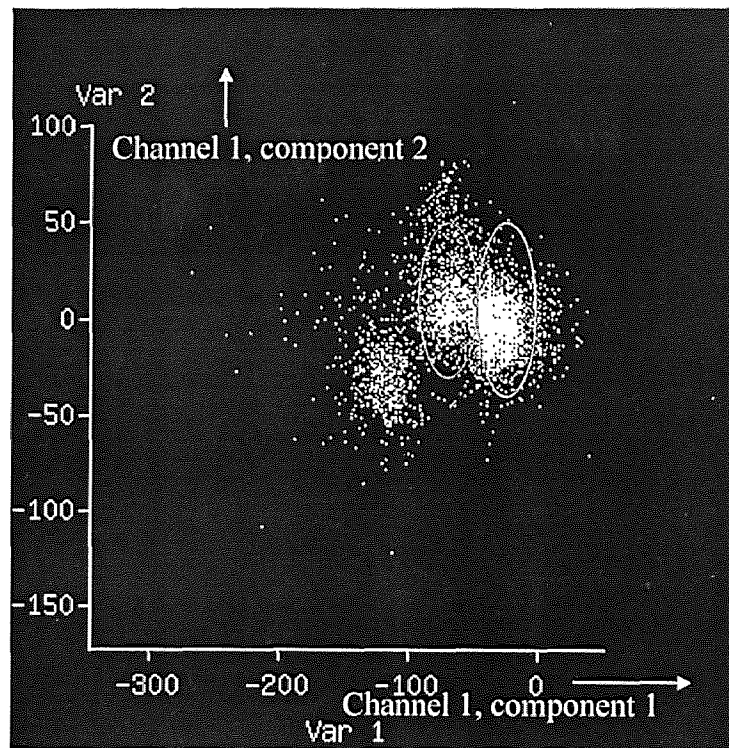


Figure 4.1. Sample spike clusters.

giving importance to both the intercluster distance and the size of the clusters.

4.1.1 Removal of Outliers

Any clustering technique is susceptible to the presence of outliers or noise points. In order to remove outliers, the first technique that we use is to filter points that are far away from their nearest neighboring points. Figure 4.2 shows the graph of points sorted according to their distance from their nearest neighbor. The graph shows that points after 4800 clearly show an increase in their distance from their nearest neighbor. These points are long distance away from their nearest neighbor, hence we consider these points as outliers and remove them from our data set.

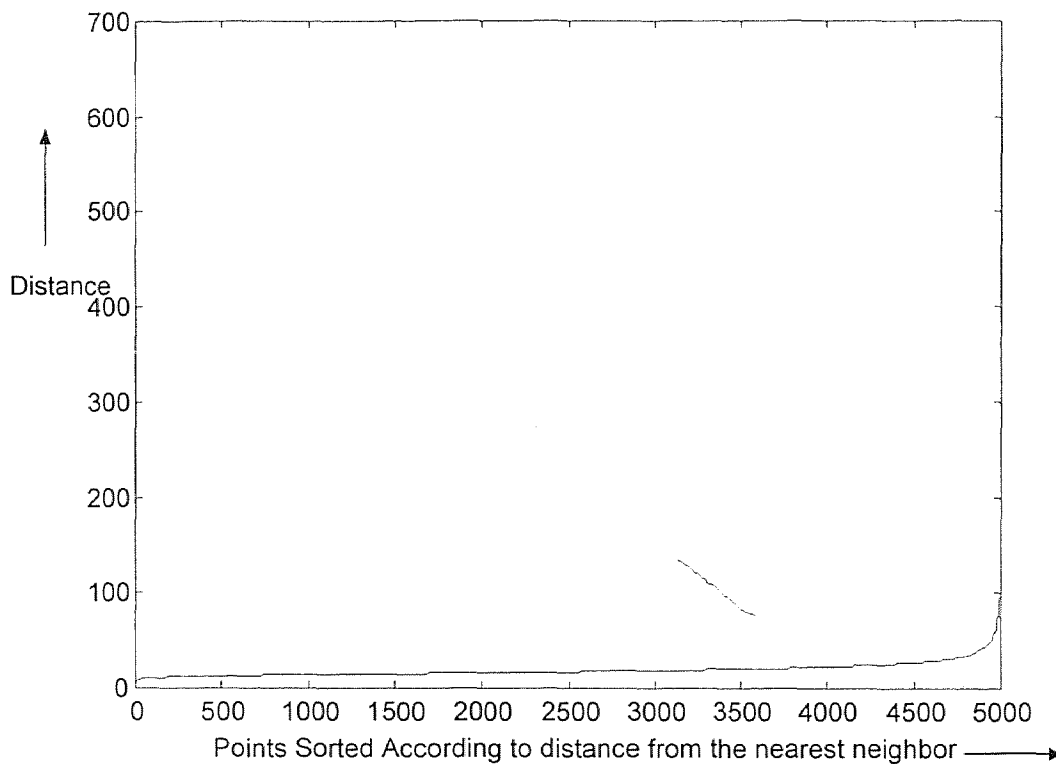


Figure 4.2 Distance of points from their nearest neighbor. The dashed line indicates number of points removed from the intracellular cluster(cluster identified by simultaneous intracellular recording). If we select 4800 points from this data set, we are losing only 3 points from the intracellular cluster.

4.1.2 Agglomerative Clustering

Hierarchical clustering algorithms can be divided in two categories namely agglomerative and divisive clustering. In case of divisive clustering techniques we start assuming that all n points belong to same cluster. We then split this single cluster to 2 cluster and then into 4 clusters and so on.

In case of agglomerative clustering techniques, we initially start with n clusters, where each cluster contains a single point. We partition these n clusters to $n-1$ clusters, the next a partition into $n-2$ and so on. During each stage of clustering we merge clusters that are closest to each other. Thus after one cycle of the algorithm we will be left with $n-1$ clusters. The process is repeated until the required number of clusters are remaining.

Hierarchical clustering techniques result in the formation of tree called as dendogram tree, which indicates how samples are grouped together. One of the limitations of the hierarchical clustering algorithms is their memory requirements. Since we need to store intercluster distances, the amount of memory required in the worst case for n clusters in $O(n^2)$.

We have used agglomerative clustering techniques for clustering of the spike data. We initially assume every point to be a single cluster. (i) For every cluster we calculate the dissimilarity distance of the cluster with every other cluster. Thus if there are m clusters we will get an $m \times m$ dissimilarity matrix. (ii) We find clusters which are closest to each other. (iii) We then merge these clusters which we have found in step(ii). Thus the number of remaining clusters after step(iii) is reduced by 1. The above steps (i) to (iii) are then repeated until we are left with only 2 clusters. Since we

are also interested in identifying the optimal number of clusters, we write the results of each cycle of clustering when we are reducing clusters from 10 to 2. Hence in the end, we will have results for 10 clusters, 9 clusters and so on.

We are reducing the number of clusters by one during each pass of the algorithm. We can improve the speed of algorithm if we can merge more than one cluster in the same pass. Hence in order to improve the speed of clustering we have divided the whole algorithm into two phases depending upon the number of remaining clusters. Since initially, number of remaining clusters is high, it will take a lot of computational time if we decide to merge two clusters which are closest to each other. Also during this phase, we are merging clusters which are smaller than 5 to 6 points and we do not require the exact granularity in our decision about two closest clusters. Hence, even if we make few wrong decisions about merging two most optimal clusters, it will not affect the final outcome of the algorithm. During the first phase we merge every cluster with its closest another cluster. Thus in effect in one cycle, we are reducing the number of clusters by $m/2$ instead of one. However during phase 2, we want better accuracy in our clustering decisions and hence we use steps (i) to (iii) as described earlier. We continue with phase 1 until we have found clusters with around 6 to 7 points each. At the end of phase1, we get around 70 clusters, each of which contain 7 points. We then revert to phase 2, and use the steps (i) to (iii) to reduce number of remaining clusters to 2. Figure 4.4 shows the complete flow chart of the algorithm.

Even though we have reduced the number of outliers as described in section 4.2.1, we still find clusters with a small number of points after running the hierarchical clustering algorithm. Figure 4.3 shows the histogram of 'number of clusters' Vs.

'number of points' in each cluster. The histogram shows that there are many clusters with the less than 50 points. As we can see out of 30 clusters, only 6 clusters contain more than 50 data points. We consider these remaining clusters as noise clusters, and filter these points from data set. The cut off that we have selected is around 0.01 times the original size of the data set. The 6 clusters selected contain around 82% of the original data set. We then again apply the hierarchical clustering algorithm to identify the clusters in the reduced data set. The output of the hierarchical clustering algorithm is shown in figure 5.2

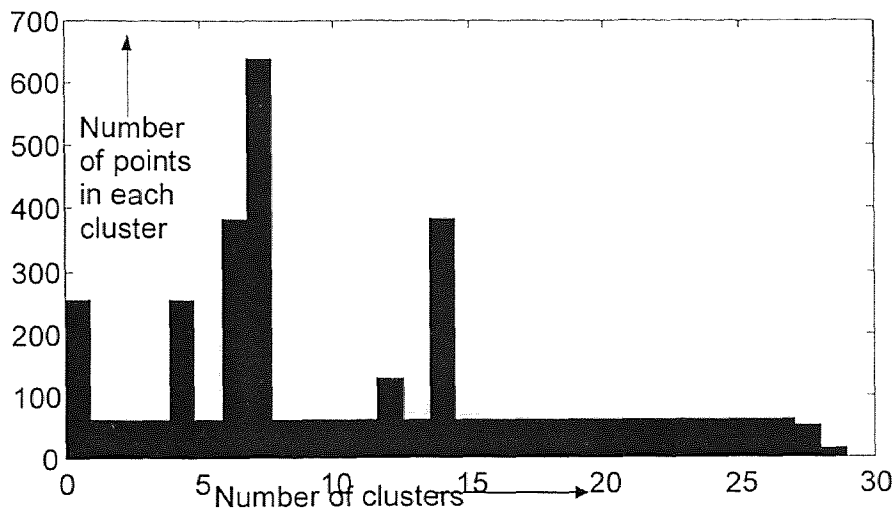


Figure 4.3. Histogram showing number of points in each cluster.

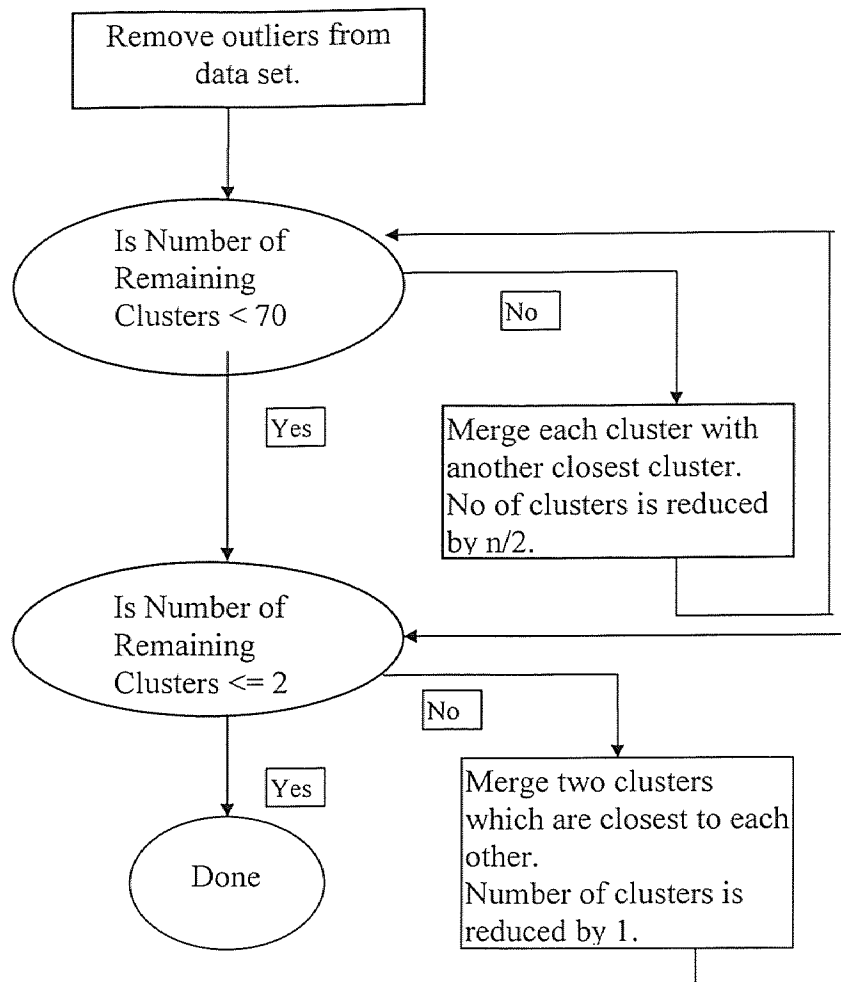


Figure 4.4. Flow-chart of the hierarchical clustering algorithm.

4.2 Estimation Maximization Algorithm

Expectation Minimization algorithm is a recent parametric technique for optimizing Maximum Likelihood of the model. It was introduced by Dempster et in year 1977. Since then it has shown great promise in solving complex problems in Bayesian statistics. We plan to model data as multivariate mixture and apply EM algorithm to perform clustering. We will use the output of the hierarchical clustering algorithm to

start the EM algorithm. We will also evaluate improvements in the clustering due to the EM algorithm.

4.2.1 Stochastic Processes

Before we apply the Bayesian approach to sorting of the spike data we need to model firing pattern of the neuron as a stochastic process. Our goal is to model total spike output as a sum of the stochastic processes. In this section we will describe characteristics of various stochastic processes that we will use in the later sections to actually perform modeling of the data. A stochastic process is a random process that generates sequential signals such as noise or speech. The variation of the signal is partially or completely random. Also occurrence of an event is random and is completely independent of earlier events.

4.2.1.1 Multivariate Gaussian Process: Gaussian process is an example of stochastic process and has been used extensively to describe many physical phenomena. The probability density function of the gaussian random variable is described by the following probability density function.

$$f(x) = \frac{1}{\sqrt{2\pi} * \sigma} e^{-\frac{(x-\mu)^2}{2*\sigma^2}} \quad [4.1]$$

μ and σ^2 are respectively mean and variance of the random variable x . The gaussian process of equation [3.6] decreases exponentially with increasing distance of x from the mean value μ . The probability distribution function $F(x)$ is given as

$$F(x) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^x \exp\left(-\frac{(x-\mu_x)^2}{2\sigma_x^2}\right) dx$$

Multi-variate Gaussian Process is an mixture of multiple vector valued gaussian Processes. Consider a d-variate Gaussian vector process $\{X = [x(m_0), x(m_1), \dots]^T\}$ with mean vector μ and covariance matrix Σ . The probability density function of the multivariate gaussian process is given by

$$f(x) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma|} \exp\left(-\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu)\right)$$

Where the mean vector μ is defined as

$$\mu = \begin{pmatrix} E[x(m_0)] \\ E[x(m_1)] \\ E[x(m_2)] \\ \dots \\ \dots \\ E[x(m_d)] \end{pmatrix}$$

The covariance matrix Σ is defined as

$$\Sigma = E[(x - \mu)(x - \mu)^T]$$

$$\Sigma = \begin{pmatrix} c_{xx}(m_0, m_0) & \dots & c_{xx}(m_0, m_1) & \dots & c_{xx}(m_0, m_2) & \dots & c_{xx}(m_0, m_d) \\ c_{xx}(m_1, m_0) & \dots & c_{xx}(m_1, m_1) & \dots & c_{xx}(m_1, m_2) & \dots & c_{xx}(m_1, m_d) \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ c_{xx}(m_d, m_0) & \dots & c_{xx}(m_d, m_1) & \dots & c_{xx}(m_d, m_2) & \dots & c_{xx}(m_d, m_d) \end{pmatrix}$$

where $c_{ab} = (x(m_a) - \mu_a) * (x(m_b) - \mu_b)$.

The mixture multivariate gaussian process is a weighted sum of a number of individual multivariate gaussian processes.

Thus mixture multivariate gaussian distribution = $\sum_{i=1}^p \pi_i g(x)$

where $g(x)$ is individual multivariate gaussian process and is given by

$$g(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x - \mu)\Sigma^{-1}(x - \mu)^T\right] \quad [4.2]$$

π_i = mixing proportion or weight of the individual gaussian Process.

The multivariate gaussian distribution is represented by $N(\mu, \sigma)$.

4.2.1.2 Poisson Process: The poisson distribution is a continuous time, integer valued counting process used for modeling of the occurrences of random events in varying time interval. The poisson distribution is generally used to model the occurrences of noise or sharp noise pulses in the input signal.

The probability density function of the poisson distribution is given by

$$P(k, t) = \frac{\lambda t^k}{k!} e^{-\lambda t}, \text{ where } k \text{ is a constant dependent on the process.}$$

4.2.2 Application of EM algorithm to Multivariate Gaussian Mixture

The EM algorithm consists of two steps, the Estimation step and Maximization step.

The estimation step consists of calculation of the posterior likelihood of the data. The

Maximization step follows the estimation step and involves maximizing the

likelihood computed in the earlier step. The process the repeated until the results

converge to some stable value. In essence, the EM algorithm tries to find the local maxima for the log likelihood of the model in the vicinity of the initial values.

In order to use EM for clustering of spike data, we assume that spike density due to each neuron is a multivariate gaussian process. In a sense we are assuming that spikes due to an individual neuron follows a random sequence and every spike occurrence is independent of any earlier spike occurrence. The signal that we collect at electrodes consists of foreground spike activity of multiple neurons and a noise component introduced due to various components as explained in chapter 2.

We consider the signal $s(t)$ to be

$$s(t) = \sum_{i=1}^p \pi_i g(t) + \eta(t)$$

where $\eta(t)$ is the noise introduced in the measurement.

Since we are assuming that noise in our measurement is a white noise, the spikes introduced due to such noise can be considered as random. Hence we consider background noise to be purely random and assume that it follows a poisson distribution.

Let us consider the d dimensional observation \mathbf{x} from a multivariate gaussian distribution of G classes.

The probability distribution of the mixture density will be

$$f(\mathbf{x}; \theta) = \sum_{k=1}^G \pi_k f(\mathbf{x}; \theta_k)$$

where π_k is the mixing proportion and $0 \leq \pi_k < 1$ also $\sum \pi_k = 1$.

And $\theta = \pi_1, \dots, \pi_G, \mu_1, \dots, \mu_G, \sigma_1, \dots, \sigma_G$ are unknown parameters.

The likelihood is given by

$$L(x; \pi; \theta) = \sum_{i=1}^n \left[\sum_{k=1}^G \pi_k f(x; \theta) \right]$$

where n = number of total points

G = total number of clusters.

In order to apply EM to clustering of data, we will add another classification matrix Z so that

$$L(x; \pi; \theta) = \sum_{i=1}^n \left[\sum_{k=1}^G z_{ik} \pi_k f(x; \theta) \right]$$

where z_{ik} is the posterior probability that the i^{th} observation belongs to cluster G and

$$\sum_k z_{ik} = 1.$$

4.2.2.1 Estimation Step: The Estimation step involves computing of posterior likelihood probability, given the current data set. Since $L(x; \theta)$ is difficult to differentiate analytically we will consider log likelihood of the data as $\lambda(x; \theta) = \ln(L(x; \theta))$.

Since the log likelihood is a linear function of likelihood, any maximum in L will also be reflected in $\lambda(x; \theta)$.

$$\lambda(x; \theta) = \sum_{i=1}^n \ln \left[\sum_{k=1}^G z_{ik} \pi_k f(x; \theta) \right] \quad [4.3]$$

Now according to Bayes rule we have

$$P(\theta|X) = \frac{P(X|\theta) * P(\theta)}{\sum P(\theta)}$$

$$P(\theta|X) \approx P(X|\theta) * P(\theta)$$

We can compute $f(x; \theta)$ by using the above Bayes rule. Using equation [4.1], [4.2] and [4.3] we get

$$\lambda(x; \theta) = \sum_1^n \sum_1^G \ln(z_{ik}) + \ln(\pi_k) + \ln(f(x; \theta))$$

In order to compute initial estimate of log likelihood $\lambda_0(x; \theta)$, we use the initial estimate of the parameters θ_0 . We have used the output of hierarchical clustering algorithm to compute initial estimates of \sum_0 and π_0 . However, θ_0 can also be computed by using hand clustering or by using the K means clustering algorithm.

The initial classification matrix \mathbf{Z} is computed as

$$z_{ik} = \frac{\pi_k f(x; \theta)}{\sum_{i=1}^G \pi_k f(x; \theta)}$$

4.4.2.2 Maximization Step: During the maximization step we try to maximize log-likelihood that we have obtained in the estimation step.

The maxima of $\lambda(x; \theta)$ will occur when

$$\frac{\partial}{\partial \theta} \lambda(x; \theta) = 0; \text{ where } \lambda(x; \theta) \text{ is given by}$$

$$\lambda(x; \theta) = -\frac{d}{2} \log(2\pi) - \frac{1}{2} \log \left| \sum \right| - \frac{1}{2} \left[(x - \mu) \sum^{-1} (x - \mu)^T \right]$$

During the maximization stage we try to maximize λ by computing a new value of

$\hat{\theta}$. Since θ is indicated by n, π and μ , the new value of parameter $\hat{\theta}$ is given by

$$\hat{n}_k = \sum_{i=1}^n z_{ik}$$

$$\hat{\pi}_k = \frac{n_k}{n}$$

$$\hat{\mu}_k = \frac{\sum_{i=1}^n z_{ik} x_i}{n_k}$$

$$\hat{\sigma}_k = (x - \mu_k)^* (x - \mu_k)^T$$

The new value of the parameter $\hat{\theta}$ is used in the calculation of the classification matrix during the subsequent estimation step.

The process of estimation step followed by maximization step is repeated until changes in parameter values is below certain threshold.

4.2.2.3 ECM or Estimation Classification Algorithm: ECM was proposed by Celeux and Govart[10] in their paper on pattern recognition. The ECM algorithm is much like the general EM algorithm except for an additional step of classification. In case of ECM algorithm the \mathbf{Z} matrix contains the classification of the points to a specific cluster. In the case of the EM algorithm we calculate the posterior estimation of the point being assigned to particular cluster, while in case ECM the point is allocated to a cluster which has maximum probability of observation. In order to do so we add an additional constraint on \mathbf{Z} that

$$Z_{ik} = \begin{cases} 1 & : \text{if } i^{\text{th}} \text{ observation belongs to cluster } k. \\ 0 & : \text{if } i^{\text{th}} \text{ observation does not belongs to cluster } k. \end{cases}$$

Thus, ECM algorithm consists of

Repeat until solution has converged

Estimate the log likelihood

Classify the observations

Maximize the log likelihood.

End.

In case of ECM algorithm we try to maximize the classification log likelihood which is given by

$$\lambda(x; \theta) = \sum_{i=1}^n \left[\sum_{k=1}^G \ln(z_{ik} * \pi_k * f(x; \theta)) \right] \quad [4.4]$$

4.2.2.4 Modifications to the EM Algorithm: The EM algorithm has generally been found to be very sensitive to the presence of large numbers of noise outliers in the data. Hence we need to modify the basic EM algorithm to take into account outliers.

4.2.2.4.1 Removal of Outliers in EM Algorithm: Outliers are set if points that are far away from the center of the cluster. In effect, the mahalanobis distance of these points is $\gg \sigma$. These outliers generally arise due to background activity of the neuronal population or due to nonlinear characteristics of the electrode. The convergence rate of the EM algorithm is inversely affected due to presence of these outliers. In some cases it might even lead to failure of the EM algorithm to reach stable solution. In order to model these outliers we take an approach similar to that used by Raftery and Dasgupta[2] for identification of minefields in the data. We

assume that outliers are distributed evenly throughout the hypervolume V and follow poisson distribution. The probability of point occurring due to noise will be $\frac{p_0}{V}$, p_0 is constant. Hence during each iteration of EM algorithm we compute mahalanobis distance of the point from the center of the cluster. If mahalanobis distance is greater 3σ we assume the point to be outlier and remove it from the data set.

4.2.2.4.2 Incorporating Refractory Period in the EM Algorithm: Until now clustering techniques that we have used have focused on use of the euclidean distance between clusters to classify points. We have not taken into account temporal properties of spikes. From neurophysiological behavior of neurons we know that once a neuron has produced an action potential, it does not produce another action potential during some interval of time. This interval of time is called as the **Refractory Period** of a neuron and is about 2 milliseconds. In order to take refractory period into account we have introduced an additional step after the classification step of ECM algorithm. During this step we calculate number of conflicts that will happen due to current set of classification. If we find that spikes in two clusters overlap then we readjust the classification for those spikes so that the number of conflicts between the two clusters is reduced.

4.2.3 How Many Clusters?

In order to determine the exact number of clusters in the data set we use the bayes factor.

$$\text{Bayes factor} = \frac{P(D|M_1)}{P(D|M_2)}$$

Bayes factor is posterior odds of one model against another. The larger bayes factor indicates higher likelihood of the model M_1 as compared to M_2

But we calculate $P(D/M)$ during each step of EM algorithm and is nothing but maximum likelihood $\lambda(x;\theta)$. Hence,

$$B = \frac{P(D|M_1)}{P(D|M_2)} \approx \frac{\text{Max. Log Likelihood of Model } M_1}{\text{Max. Log Likelihood of Model } M_2}$$

In our case model M_n is a model containing n classes.

In order to select the best fit we use the Bayesian Information Criteria [6] to find most matching model. BIC is basically derived from bayes factor and consists of an additional term to take into account number of independent parameters of the model.

For multivariate gaussian mixture model

$$\text{BIC} = 2 \log p(x|G) \approx 2\lambda(x, \theta, G) - m_c \log(n),$$
 where m_c is number of independent parameters for model c .

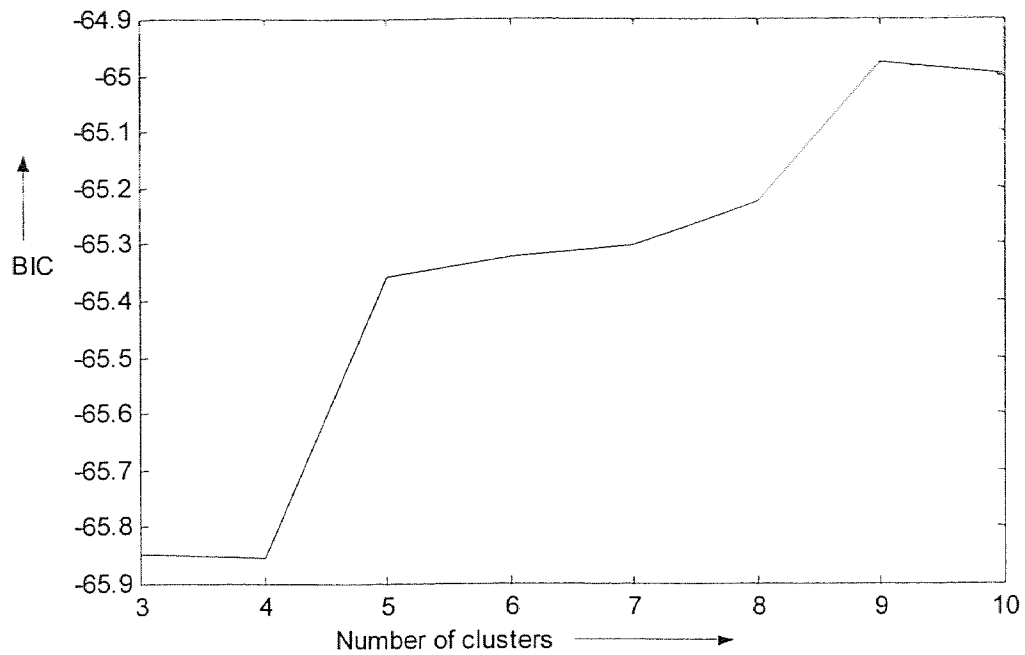


Figure 4.5 BIC Vs. number Of Clusters

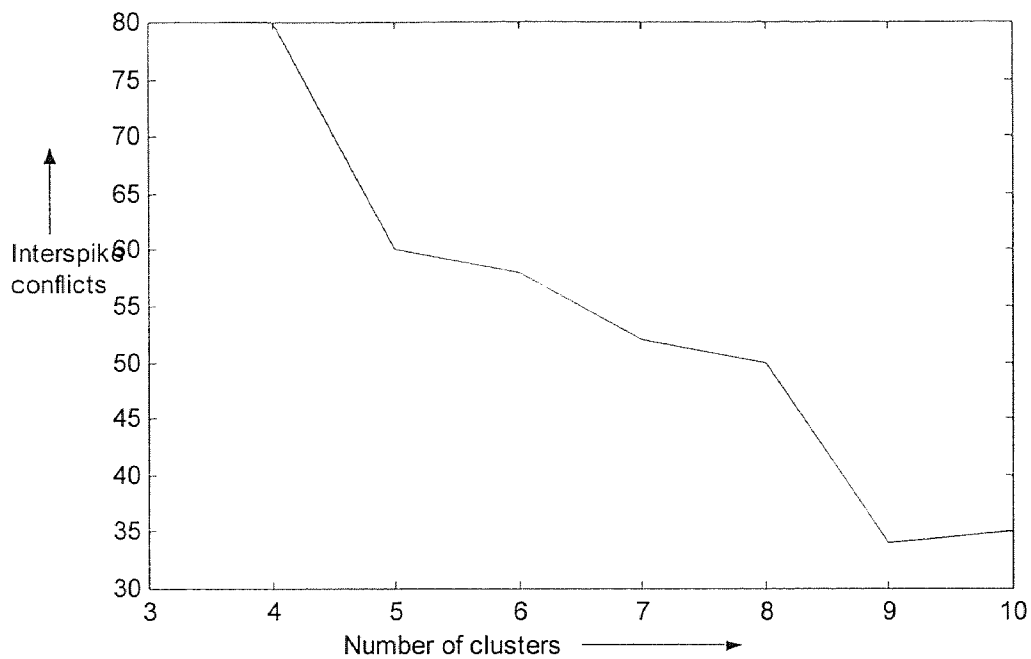


Figure 4.6 Interspike conflicts Vs. Number Of Clusters

Figure 4.5 shows the plot of BIC vs. model. The plot shows that there is a local maximum for $n = 9$ clusters. Hence, We conclude that the model containing 9 clusters is the best fit for the data set. In order to check the validity of our model, we have also computed graph of number of inter spike conflicts Vs number of clusters. As we can see from figure 4.6, when the number of spikes is 9 the number of interspike conflicts is lowest. Figure 5.6 shows the 9 spike shapes identified as a result of the application of the EM algorithm.

4.2.4 Limitations of the EM Algorithm

EM algorithm depends upon the starting values or the seed values of the unknown parameters. Hence the rate of convergence of EM depends upon these starting values and sometimes the algorithm might not even converge. Second problem is that the rate of the convergence a EM is very slow. Also the EM algorithm can not proceed if any of the covariance matrix becomes singular.

CHAPTER 5

EXPERIMENTAL RESULTS

We have focused our experiments on the data set shown in Figure 5.1¹. Figure 5.1 shows output of the principal component analysis of the original data. We have used Xgobi software to visualize data in multidimensional coordinate axis. Xgobi is a multivariate data visualization system. It was designed by Deborah F. Swayne, Di Cook and Andrea Buja and is available from “<http://lib.stat.cmu.edu/general/XGobi/>”. **The red cluster shows the spikes that have been identified by using the simultaneous intracellular recording.** In most cases of spike sorting algorithms we can not validate results, other than relying on statistical inferences. However in our case, we will compare our result set with respect to actual experimental recording.

1. The data used in experiments was collected by Dr. Darryl Agnze. We are thankful to Dr. Kenneth Harris for making this data available to us for analysis.

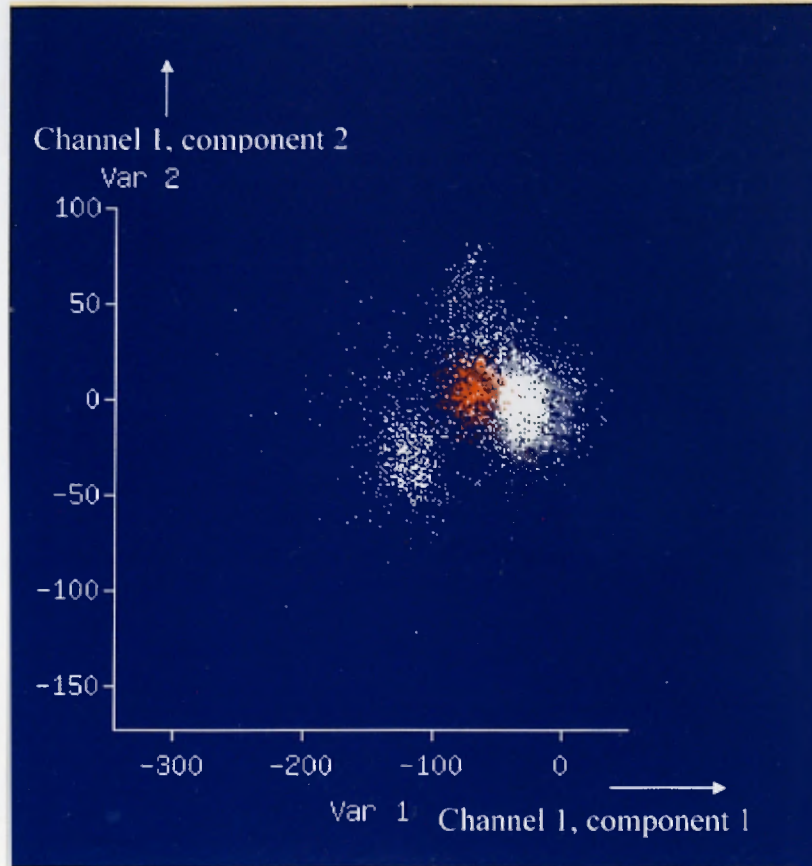


Figure 5.1. PCA output of input data in two dimensions. The graph shows output of first PCA on channel 1 with second PCA on channel 2. The red dots show the intracellular cluster.

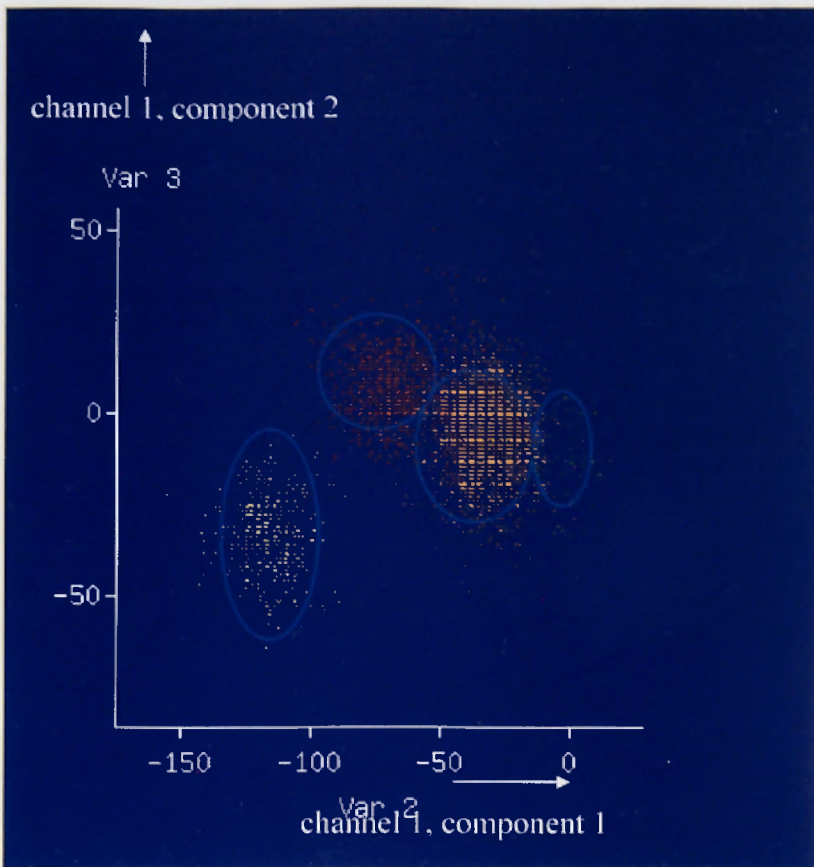


Figure 5.2 Output of the hierarchical clustering algorithm. We can clearly see 4 separated clusters. We have also been able to find out intracellular cluster as shown in Figure 5.1

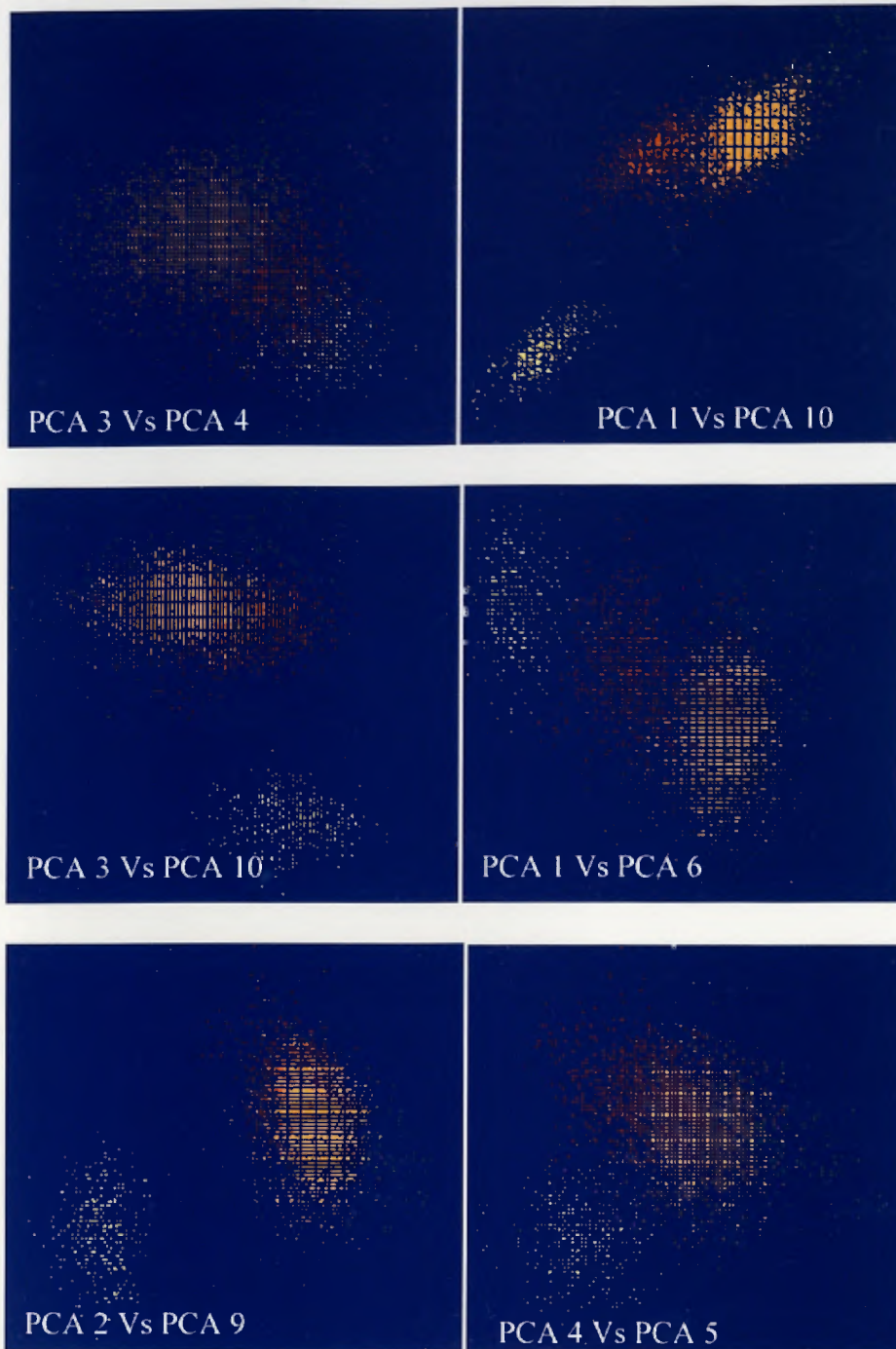


Figure 5.3. Output of hierarchical clustering algorithm on various principal component axis.

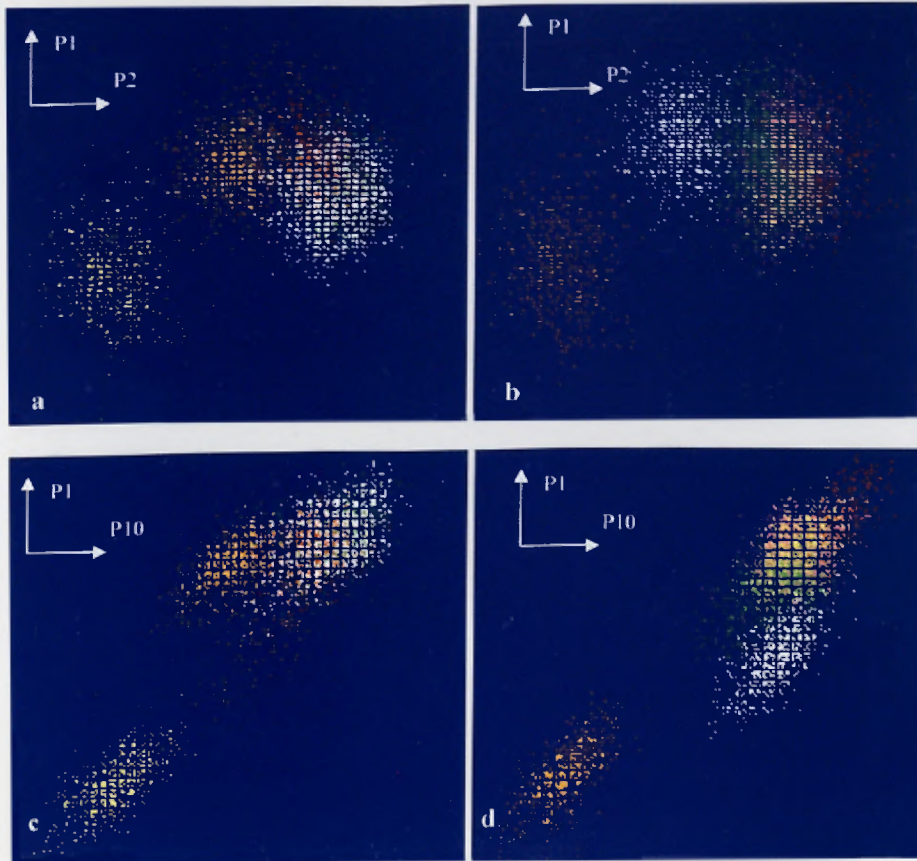


Figure 5.5 Improvement in the clustering by application of EM algorithm. Figure a) and c) indicate the output of the hierarchical clustering algorithm for six clusters. Figure b) and d) show the corresponding improvement in the clustering by application of EM algorithm. The clusters in a) and c) have been refined by the EM algorithm as shown in b) and d).

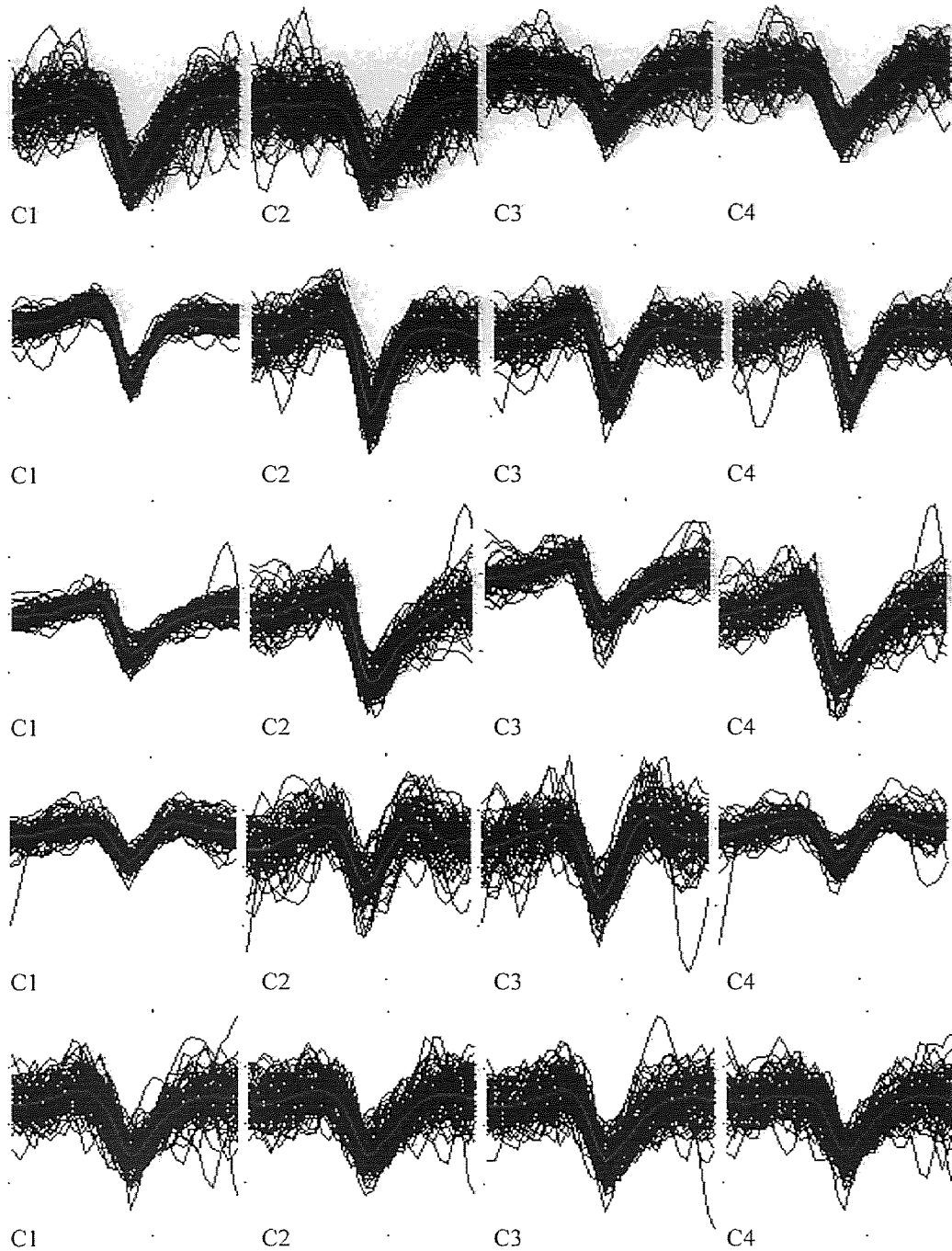


Figure 5.6. a) Representative spike wave forms for 5 of the 9 spikes identified in data set of fig 5.1.

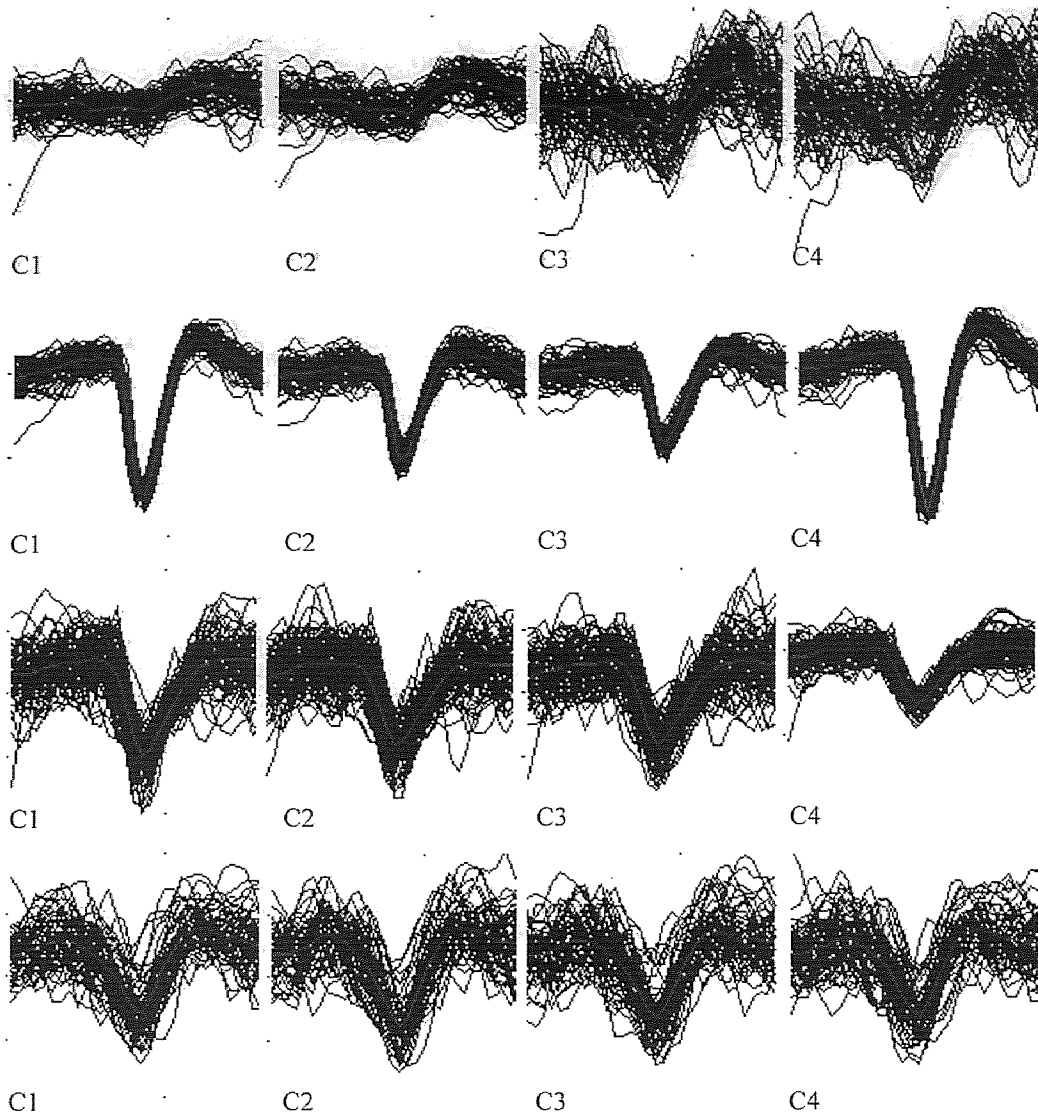


Figure 5.6. b) Representative spike wave forms for remaining 4 of the 9 spikes identified in data set of fig 5.1.

REFERENCES

- [1] Abeles M. and Goldstein M.H. "Multispikes Train Analysis", *Proc. IEEE*, 65: 762-773 (1977)
- [2] Andrew Gelman, John B. Carlin, Hal S. Stern and Donald B. Rubin. *Bayesian Data Analysis*, London: Chapman and Hall (1998)
- [3] C. Fraley. "Algorithms for model-based gaussian hierarchical clustering", *Technical Report 311, Department of Statistics, University of Washington*. Seattle, WA (1996)
- [4] Charles M. Gray, Pedro E. Maldonado, Mathew Wilson. "Tetrode markedly improves the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex", *Journal of Neuroscience Methods* 63, 43-44, (1995)
- [5] Dasgupta A. and A. E. Raftery. "Detecting features in spatial point process with clutter via model based clustering", *Technical Report No. 295, Department of Statistics, University of Washington*. Seattle, WA (October 1995)
- [6] Fraley C. and A. E. Raftery. "How many clusters? which clustering method? answers via model based clustering analysis" *Technical Report No. 329, Department of Statistics, University of Washington*. Seattle, WA (1997). Also available from "<http://www.stat.washington.edu/www/research/reports/>"
- [7] Duda and Hart. *Pattern Classification and Scene Analysis*, New York: Wiley (1973)
- [8] M. Schmidt. "Computer separation of multi-unit neuroelectric data: A review", *Journal of Neuroscience Methods*, 12:95-111 (1984)
- [9] Edmand M. Glaser, Daniel S. Ruchkin. *Principles of Neurobiological Signal Analysis*, New York: Academic Press (1976)
- [10] Gilles Celeux, Gerard Govaert. "Gaussian parsimonious clustering methods", *Pattern Recognition*, Vol. 28, 781-793 (1995)
- [11] Geoffrey J. McLachlan and Thirimbakam Krishnan. *The EM Algorithm and Extensions*, New York: Wiley (1999)
- [12] J. A. Hartigan, *Clustering Algorithms*, New York: Wiley (1975)

- [13] Kaufman and P.J. Rousseeuw. *Finding Groups in Data*, New York: Wiley (1990)
- [14] Recce M. L. and O'Keefe J. "The tetrode: a new technique for multi-unit extracellular recording", *Soc. Neuroscience Abstr.* (15)1250 (1989)
- [15] Sahani M., J. S. Pezaris "On the separation of signals from neighboring cells in tetrode recordings", *Proceedings of NIPS* (1997)
- [16] Sahani M., Pezaris J. S. and Andersen R. A. "Extracellular recording from adjacent neurons, a maximum-likelihood solution to the spike separation problem", *Soc. Neuroscience. Abstr.* 23 1546 (1997)
- [17] Sahani M. Ph.D. Thesis. *Dept. of Computational and Neural Science, California Institute of Technology*. Pasadena CA. (1998)
Also available from "<http://www.gatsby.ucl.ac.uk/~maneesh/thesis/>"
- [18] Mark F. Bear, Barry Connors. *Neuroscience: Exploring the Brain*. Baltimore: Williams and Williams Publications, (1997)
- [19] Michel S. Lewicki. "Bayesian modeling and classification of neuronal signals" *Neural Comp.* 6(5):1005 (1994)
- [20] Michael S. Fee, Partha P. Mitra, David Kleinfeld "Automatic sorting of multiple-unit neuronal signals in the presence of anisotropic and non-gaussian variability", *J. Neuroscience Methods* 69:175 (1996)
- [21] Rebrik, Wright and Miller. "Cross channel correlation in tetrode recordings", *Computational Neuroscience* (in special issue containing proceedings of 1998 Computation and Neural Systems meeting). Software available from <http://mccooy.ucsf.edu>.
- [22] Saeed V. Vaseghi. *Advanced Signal Processing and Digital Noise Reduction*. New York: Wiley and Teubner, (1996)
- [23] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery. *Numerical Recipes in C*, Cambridge, MA: Cambridge University Press (1992)
- [24] Yang X., Shamma S. A. "A totally automated system for the detection and classification of neural spikes", *IEEE Trans. Biomed. Eng.* 35 806–16 (1988)
- [25] Zhang K. C., Sejnowski T. J. "Automatic separation of spike parameter clusters from tetrode recordings", *Soc. Neuroscience Abst.* (23)504 (1997)