Multiunit Neural Spike Sorting using a Discrete Gaussian Approximation of Bayes Classifier.

J. P. Stitt*, R. P. Gaumond*, J. L. Frazier**, and F. E. Hanson***

* Bioengineering, Pennsylvania State University, University Park, PA
** Entomology, Pennsylvania State University, University Park, PA
*** Biology, University of Maryland, Baltimore, MD

Abstract
We describe the implementation of a Bayes classifier for identifying the activity of multiple neural units recorded by a single electrode. The implementation involves the discrete Gaussian approximation of a Bayes classifier. We show that the two classic methods of spike sorting, template matching and the method of principal components, are special cases of this implementation of a Bayes classifier. The sorting accuracy and computational complexity of the classical techniques are compared with those of the Bayes classifier. The latter is shown to outperform these classic methods.

Keywords: Bayes classifier; action potentials; template matching; principal components

Introduction
The problems associated with classifying action potentials (APs) in a single-channel multunit recording have been studied for several decades (1-4). A number of classification algorithms have been proposed which will be discussed in the following paragraphs. Many approaches involve simplifying assumptions; these assumptions either facilitate the derivations or reduce the computational complexity required by the algorithm. The most common assumptions concern the characteristics of the noise sources that contribute variation to AP shape. The common assumptions are: (1) the noise is independent of the unit that generates the AP; and (2) the noise is uncorrelated from sample to sample and has an infinite bandwidth (i.e., white noise). These simplifying assumptions lead to the two classic algorithms that have been employed to implement AP classifiers: (1) template matching; and (2) the method of principal components.

Early multiunit classifiers, using analog electronics, based decisions upon a set of features such as amplitude and slope values specified by the investigator. Laboratory computer workstations equipped with analog to digital converters enable the implementation of more complex algorithms that can potentially increase the classification accuracy at a cost of increased computational complexity. These digital identification algorithms typically have distinct modules for the detection, extraction, and storage of the discrete-time samples of the AP, and subsequent classification of the set of extracted APs.

Early digital spike classifiers compared a set of spike shape features to preset values. The feature set consisted of the sampled signal itself, or a small number of derived features (1, 4) e.g., peak-to-trough amplitude (5). Reducing the size of the feature set reduces computational cost but eliminates some information, thereby reducing the accuracy of the classification (6, 7).

The method of template matching (TM) refers to a classifier algorithm that associates a spike waveform with a particular neural unit by comparing a set of data samples X, representing the discrete-time amplitude samples of an unknown spike, with a set of patterns Ti i ∈ {1,2,..., L}. Vector X is assigned to the class for which ||X-Ti|| is minimal.

Several authors have applied linear filter theory to the spike-sorting problem (8-10). These methods involve designing a bank of filters, one filter for each unit in the recording. The unit specific filter should respond maximally only when APs of the associated neuron are present in the spike train. However, results are not always satisfactory since in many cases the spikes produced by all of the units are found to have very similar frequency spectra. The classification results of linear matched filters are inversely related to those of the TM method if the optimal filters that are implemented are matched filters (11).

Another method of AP identification is the method of principal components (PC) (1). The PC method simultaneously reduces the dimensionality of the TM classifier and transforms the covariance matrix, associated with the noise source, to a diagonal colored-noise matrix.

More recently, artificial neural network (ANN) theory has been applied to the spike-sorting problem (12-14).
ANNs can be employed to estimate any nonlinear relationship (15). An appropriately designed ANN can "learn" any relationship between the set of input arguments and the corresponding set of desired outputs. Our implementation of an ANN spike classifier has been shown to outperform both TM and PC classifiers (14). A major drawback to the use of ANNs is the large amount of time that is required to determine the appropriate architecture and to train the ANN. Another problem is that of formulating a theoretical basis for the classifications that result from the ANN classifier. Unless the ANN is particularly simple, it is difficult or impossible to interpret what function the ANN is implementing. Despite these drawbacks, ANNs can be implemented easily in software to serve as improved spike classifiers, and in hardware (both digital and analog VLSI) where they may serve as nonlinear filters and overcome some of the shortcomings of the linear filters discussed above.

In the following discussion we consider an approximation of the Bayes Classifier allowing for a unique non-diagonal noise covariance matrix to be associated with each unit. We show that the TM and PC methods are special cases of the Bayes classifier. And we compare accuracy and computational complexity for all 3 schemes using action potential data from a chemosensory preparation of tobacco hornworm (Manducca Sexta) where multiple neural units are recorded at a single electrode when complex chemosensory excitation is applied, but where the individual units can be identified and characterized using simple chemosensory stimuli.

Classifier Methods

Each AP, or spike, is a random signal; however, they all possess common features which can be used to detect the occurrence of a spike within the time sequence. Once an AP has been detected, it is extracted from the time sequence and classified as one of the L possible units.

The occurrence of an AP in unit \(i \in \{1, 2, ..., L\}\) generates a vector \(X\), a sample function of a random process associated with that unit. If the distributions of these random processes are known for each of the \(L\) units, then the theory of statistical hypothesis testing can be used to develop a classifier that results in the minimum probability of error. The optimal classifier is termed the Bayes Classifier for minimum error and the minimal error is termed the Bayes error (11, 16).

The distributions are in general, unknown and must be approximated by statistical measures derived from a set of sample functions. The characteristic parameters of the \(i\)th distribution can be estimated when it is possible to identify sample functions belonging to a particular unit. In the biological application we discuss below (chemosensory neurons in Manducca Sexta) it is possible to apply stimuli which elicit AP’s from one unit only, then to consider the problem of identifying units when more complex chemosensory stimuli are applied and multiple units respond at closely spaced intervals.

If we designate \(S\) as the space of all possible spike waveform shapes, then any given waveform \(X\) is associated with a probability density function \(p_{\text{Mixture}}(X)\). The \(L\) classes of spikes form a partition of \(S\) and the \(i\)th class \(C_i\) is associated with a class-conditioned density function denoted \(p(X|C_i)\). The value of \(p(X|C_i)\) is the probability that \(X\) belongs to the \(i\)th class. The mixture density function \(p_{\text{Mixture}}(X)\) (Eq. 2.1) is the weighted sum of all of the class-conditional density functions that can occur.

\[
p_{\text{Mixture}}(X) = \sum_{i=1}^{L} P_i p(X|C_i) \quad (2.1)
\]

The weighting factor \(P_i\) is the a priori probability for \(C_i\). In the absence of information to the contrary, it is assumed to be the same for all units (i.e., \(P_i = 1/L\)).

Maximum A Posteriori Probability (MAP) Classifier.

The decision rule of the multihypothesis Bayes Classifier assigns the waveform \(X\) to the class associated with the largest value of a posteriori probability \(q(C_i|X)\). The a posteriori probability is the conditional probability of the \(i\)th class given the sample vector \(X\). Applying Bayes Rule, the \(q(C_i|X)\) is equal to the \(i\)th class's conditional density \(p(X|C_i)\) weighted by the ratio of its a priori probability \(P_i\) and the mixture probability.

\[
q(C_i|X) = \frac{P_i p(X|C_i)}{p_{\text{Mixture}}(X)} \quad (2.2)
\]

Since \(p_{\text{Mixture}}(X)\) is identical for all neurons, and \(P_i\) will be assumed to be equal for all neurons, maximizing \(q(C_i|X)\) is equivalent to maximizing \(p(X|C_i)\).

The conditional density for each class can be assumed Gaussian because the “noise” sources are numerous, as discussed in Appendix 1. The Central Limit Theorem (CLT) is invoked to argue that the joint amplitude distribution for the elements of \(X\) can be modeled as an \(N\)-dimensional joint Gaussian distribution. Each distribution is completely characterized by two parameters, the expected value \(M_i\) and the covariance \(\Sigma_i\), and the values of these parameters can be estimated from the sample mean and covariance of the \(N_i\) waveforms of known origin associated with the \(i\)th unit.

\[
M_i = \frac{1}{N_i} \left[ \sum_{n=1}^{N_i} X_{i,n} \right] \quad (2.3)
\]

\[
\Sigma_i = \frac{1}{N_i} \left[ \sum_{n=1}^{N_i} (X_{i,n} - M_i)(X_{i,n} - M_i)^T \right] \quad (2.4)
\]
For an N-dimensional vector $\mathbf{X}$, therefore, the PDF associated with the $i$th class is

$$p(X|\theta) = \frac{1}{\sqrt{2\pi^N |\Sigma|}} \exp\left(-\frac{1}{2} \cdot \frac{d^2}{d^2_{\text{mal}}} \right)$$  \hspace{1cm} (2.5.a)

$$d_{\text{mal}}^2(X) = (X-M_i)^T \Sigma_i^{-1}(X-M_i)$$  \hspace{1cm} (2.5.b)

The conditional density function of (2.5) is applied to each unknown spike waveform of an electrophysiological recording and is assigned to the class which has the greatest value of $p(X|C_i)$. A reduction in the number of computations required to calculate $p(X|C_i)$ (called the likelihood function) can be achieved by applying the natural logarithm to Eq. 2.5.

Elimination of the additive constant $(N/2 \cdot \log 2\pi)$ which is common to all $p(X|C_i)$ results in equation 2.6, where multiplication by -2 has reversed the sense of the inequality. These operations convert the PDF into a distance metric.

$$d_{\text{MAP}} = (X-M_i)^T \Sigma_i^{-1}(X-M_i) + \ln(|\Sigma_i|)$$  \hspace{1cm} (2.6)

The first term on the right hand side of Eq. 2.6 is the square of the Mahalanobis distance when the covariance matrix is an identity matrix.

The MAP classifier assigns $X$ to $C_i$ if $d_{\text{MAP}}$ is a minimum for $C_i$. Note that the Euclidean distance is a special case of the Mahalanobis distance when the covariance matrix is an identity matrix.

We next consider template matching and principal component classifiers as special cases of this MAP classification scheme.

**Template Matching.**

Template matching has a long history in the classification of APs obtained from single-channel multunit recordings (17, 18). The template associated with the $i$th class is found by calculating the ensemble average of a group of observations known to be associated with the class. We can designate these templates $M_i$. The method of template matching assigns X to the class for which $M_i$ has the minimum Euclidean distance from $X$ that is, $C_i$ that is associated with the minimum value of $\|X - M_i\|$. To see how this condition relates to the MAP metric $d_{\text{MAP}}$ of equation (2.6) consider the following assumptions. Let all the elements of $(X - M_i)$ be independent and identically distributed random noise samples, with variance $n$. Under this assumption $\Sigma = nI$ where I is the N-dimensional identity matrix. The distance metric of (2.6) becomes

$$d_{\text{MAP}} = (X-M_i)^T I^{-1}(X-M_i) + \ln(|I|)$$  \hspace{1cm} (2.8)

If the factor $n$ is identical for all classes $C_i$, it will have no impact on the test, and the template matching algorithm thus reduces to assignment of $X$ to the class which minimizes the value of $d_{\text{TM}}$ equal to the Euclidian distance between $X$ and $M_i$.

$$d_{\text{TM}} = (X-M_i)^T (X-M_i)$$  \hspace{1cm} (2.9)

**Principal Components.**

The method of principal components (PC) is a feature extraction technique that reduces the dimensionality of the classification problem. The essence of the PC method is the transformation of the linearly independent N-dimensional basis of the measurement space to an orthonormal basis. This transformation is accomplished through the application of an orthogonalization algorithm such as the Gram-Schmidt (19). Transforming the covariance matrices and selecting the N most significant eigenvectors converts Eq. 2.6 into Eq. 2.10.

$$d_{\text{PC}} = (X'-M_i')^T \Sigma_i^{-1}(X'-M_i') + \ln(|\Sigma_i|)$$  \hspace{1cm} (2.10)

The primed signal vector indicates that it has been diagonalized and that the number of columns and subsequently the dimensionality have been reduced from the original N to a lower number $N'$. Similarly, the mean vector $M_i$ and the waveform vector $X$ are primed to indicate their reduction in dimension. The transformation operator is a matrix that has as its columns the eigenvectors of the orthonormal basis. The eigenvalues associated with each eigenvector serve as a measure of the contribution of the eigenvector to the nature of the sample $X$. Typically, only a small number $N'$ of the eigenvalues (typically two or three for APs) are of any significant value and as a result the remaining $(N-N')$ dimensions are ignored.

In addition to reducing the dimensionality of the classification problem, the PC method results in a diagonal covariance matrix for each class. The covariance matrix is diagonal because an orthogonal basis is inherently uncorrelated. If the sample $X$ is prewhitened before calculating the $d_{\text{PC}}$ then all of the $\Sigma_i = I$. By prewhitening the covariance matrices and assuming that $\Sigma_i$ are identical for all classes, the resulting classifier is an example of template matching in a lower dimensional space. Template matching has already been shown to be a special case of the MAP classifier and, since the PC method is an example of template matching, it is also a special case of the MAP classifier.

**Experimental Methods.**

We apply the classifier methods discussed in the previous section to a biological system in which there are a known number of neurons generating APs and the shapes of the APs are distinct and can be used to determine the origin of each AP. In this preparation it is important to determine
which neurons are active when they are exposed to a complex chemical compound.

Figure 1 contains plots of spikes produced by four different neurons that were recorded by a single electrode in our preparation (14). These spikes are ensemble averages of several hundred spikes evoked by selectively activating each neuron individually. The spikes produced by these four neurons can be classified easily by comparing amplitude values at a number of sample points. The clear difference between amplitude values suggests either that each neuron is located at a different distance from the electrode or that they differ in diameter or a combination of these two properties (20).

The preparation is considered here is the tobacco hornworm (*Manduca Sexta*). In the medial styloconic taste receptor of this animal, four chemosensory neurons are encased within a cuticular structure having a porous tip. A micropipette, containing the stimulating compound, is placed over the tip and used to record the responses of all four receptors.

Figure 1 shows the average response to each of the four compounds that are known to evoke activity from one receptor. Each waveform is the average of between 100 and 250 responses. Waveforms were digitized at a 10KHz rate. Spikes were detected by an algorithm which looks for occurrences of an energy peak in a narrow frequency band. Spike peaks are aligned and the average waveshapes determined as shown in Figure 1.

Figure 1 - The four spikes associated with the four neurons that are present in the single-channel multiunit recording of our preparation.

Spike amplitudes clearly differ for the mean waveforms shown. However amplitude variation is such that there was significant overlap between adjacent classes for all classes. This is especially true for classes 2 and 3.

The time course of spikes in each class differs subtly from that in all other classes. An optimal classifier should make use of both the amplitude and the time course variations in order to distinguish between neural classes.

The data set considered consists of recordings in a single animal in which four reference compounds were applied, one corresponding to each class of neural response shown. The data set included 775 spikes (249 Class 1; 116 Class 2; 177 Class 3; 233 Class 4). Two-thirds of this data set formed the training set that was employed to parameterize the classifiers. The classification algorithms were applied to the remaining one-third of the data set. It may be assumed that spikes elicited by these reference compounds are from a single class of neuron (5). The accuracy of the classifier algorithms will be judged on this basis.

Results

Table 1 contains the results of applying the test set of spikes to each of the classifiers that are listed in the first column. The class-conditional means $M_i$ used in equations 2.6 and 2.9 are obtained directly from the waveforms shown in Figure 1 by selecting the first 32 samples. The second column lists the percentage of spikes that were correctly classified by the corresponding classifier. The third column contains the number (in thousands) of floating-point operations (kFLOPS) that was required by each classifier to complete the classification of all of the spikes in the test set.

The MAP classifier is the discrete Gaussian approximation of Bayes classifier and none of the other techniques outperformed MAP in correctly classifying the ensemble of spikes. The MAP classifier correctly classified 99% of the test set. Under the assumptions that the covariance matrices of all classes are equivalent and equal to the identity matrix, both the TM and the method of PCs (see PC - Linear) correctly classified 96% of the test set spikes. The TM requires only about one tenth the number of kFLOPS required by the MAP to classify the test set. The PC method reached the same percentage of correct classifications as TM when employing four PCs; however, the number of operations required to map $X$ into the orthogonal space and then to classify each spike meant that the PC method required roughly the same number of operations as the MAP to classify the test set.

Prewhitening involves applying a linear transformation to each resultant vector $(X-M_i)$ prior to classification. The transform for each class is the square root of the inverse of the $i^{th}$ class covariance matrix. A prewhitening filter may be approximated by an autoregressive (AR) parametric model as well (21). The application of the whitening transformation incurs additional computational cost, and this overhead is included in the count of kFLOPS of the corresponding classifier. Prewhitening the spikes of the test set reduced the classification accuracy for both TM and PC methods. For both the TM and PC methods,
the application of the whitening transformation requires more operations to complete the classification of the test set than is required by the MAP classifier. The PC method required all 31 PCs to reach the same level of correct classification as the prewhitened TM method.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Percent Correct</th>
<th>kFLOPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAP</td>
<td>99</td>
<td>893</td>
</tr>
<tr>
<td>Template Matching</td>
<td>96</td>
<td>68</td>
</tr>
<tr>
<td>Principal Components:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC-Linear #1</td>
<td>83</td>
<td>840</td>
</tr>
<tr>
<td>PC-Linear #2</td>
<td>90</td>
<td>842</td>
</tr>
<tr>
<td>PC-Linear #3</td>
<td>95</td>
<td>845</td>
</tr>
<tr>
<td>PC-Linear #4</td>
<td>96</td>
<td>847</td>
</tr>
<tr>
<td>Prewhitened:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Template Matching</td>
<td>88</td>
<td>906</td>
</tr>
<tr>
<td>Principal Components:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PC #1</td>
<td>30</td>
<td>3</td>
</tr>
<tr>
<td>PC #2</td>
<td>30</td>
<td>8</td>
</tr>
<tr>
<td>PC #3</td>
<td>46</td>
<td>14</td>
</tr>
<tr>
<td>PC #4</td>
<td>41</td>
<td>23</td>
</tr>
<tr>
<td>PC #5</td>
<td>55</td>
<td>33</td>
</tr>
<tr>
<td>PC #6</td>
<td>59</td>
<td>45</td>
</tr>
<tr>
<td>PC #7</td>
<td>48</td>
<td>58</td>
</tr>
<tr>
<td>PC #8</td>
<td>61</td>
<td>73</td>
</tr>
<tr>
<td>PC #9</td>
<td>78</td>
<td>90</td>
</tr>
<tr>
<td>PC #31</td>
<td>88</td>
<td>906</td>
</tr>
</tbody>
</table>

| PC-MAP:                  |                 |        |
| PC #1                    | 85              | 840    |
| PC #2                    | 91              | 845    |
| PC #3                    | 91              | 852    |
| PC #4                    | 93              | 859    |
| PC #5                    | 97              | 869    |
| PC #6                    | 97              | 880    |
| PC #7                    | 98              | 893    |

Table 1 Listing of the percent correctly classified and the number (in thousands) of floating point operations (kFLOPs), that each of the classifiers from column number one, required to classify the entire test set.

The final classifier implementation involves a MAP classifier of the transformed signals of the PC method. The PC-MAP method attains its maximum level of correct classifications in a lower dimensional space (7 PCs vs. 31 for MAP) than the MAP. However, when the number of calculations required by the transformation of each X onto the orthogonal space is taken into account, that number of operations is roughly the same as required by the MAP while the accuracy is slightly lower. The calculation of six or fewer PCs results in a MAP-PC classifier that requires fewer calculations than the MAP classifier; however, the percent error is greater than the MAP method alone.

Conclusions

The MAP classifier outperformed all other classifier implementations. This conclusion can be attributed to the fact that the MAP classifier makes use of more of the available information than does any other method. The cost of the higher accuracy is an increase in the computational complexity over the TM method and the lower dimensional PC implementations. The method of PC does reduce the dimensionality of the problem, but in this case seven PCs are required to reach the maximum percent correct. The number of operations required to map and classify the test set exceeds the number of operations required by the MAP and thus there is no advantage to the method of PC.

As discussed above, prewhitening each X before classification should eliminate the need for second and higher order statistics. In both the TM and PC cases, use of the optimal whitening transformation (i.e., square root of the inverse of \( \Sigma \)) requires more computations than does the MAP classifier. In addition, the percent correct is less than that obtained from the MAP classifier. This reduction in accuracy is inherent to the limitations of obtaining estimates of the prewhitening process. An even greater reduction in classification accuracy can be assumed if an AR parametric approximation of the prewhitening filter is employed.

The benefit of employing all of the available information is exemplified in the MAP classifier's performance. This fact suggests that the use of higher order statistics with a parametric model other than Gaussian may result in a Bayes classifier capable of 100% correct classification. While all of the random variables that form X are unimodal, a number of them are only approximately Gaussian. This suggests that an implementation of a Bayes classifier, employing nonparametric PDF estimates, may also improve classification results.

References


**Appendix 1: Sources of error**

An activated neuron communicates its level of activity by sending a pulse frequency modulated (PFM) signal from the axon hillock to its synaptic terminal. The pulse frequency (i.e., pulses/second) is directly related to the level of activity; however, the frequency is also a random process. Thus, the interspike is a nonlinear (random) process making it impossible to align the phase of the peak pulse amplitude of AP. In general, the peak amplitude will occur between two sample points, and the digitized APs will invariably have different amplitudes. This form of error is termed phase error and it is suggested that phase error accounts for the largest portion of the noise within an AP's PDF (7). In addition to phase error, the amplitude may fluctuate from one AP to the next due to the random nature of channel openings; this may result in a different number of channels being open during any AP.

A second source of error that introduces noise into the AP signal occurs when the continuous signal is digitized. Conversion of continuous amplitude to finite length digital amplitude requires amplitude quantization. The continuous valued amplitude is rounded to the nearest discrete level. This quantization error is a source of uniform random noise.

The recording equipment introduces a third source of noise. The electronic components within the equipment are the source of this thermal noise.