BIOMEDICAL ENGINEERING AND BIOINFORMATICS
INDEPENDENTLY COUPLED HMM SWITCHING CLASSIFIER FOR A BIMODEL BRAIN-MACHINE INTERFACE

Shalom Darmanjian\(^1\), Sung-Phil Kim\(^2\), Michael C. Nechyba\(^3\), Jose Principe\(^1\), Johan Wessberg\(^5\), Miguel A. L. Nicolelis\(^4\)

\(^1\)Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6200
\(^2\)Department of Computer Science, Brown University, Providence, RI 02912
\(^3\)Pittsburg Pattern Recognition Inc, Pittsburgh, PA 15222
\(^4\)Duke University, Durham, NC 27706
\(^5\)Goteborg University, Goteborg, Sweden.

ABSTRACT

Our initial attempt to develop a switching classifier used vector quantization to compress the multi-dimensional neural data recorded from multiple cortical areas of an owl monkey, into a discrete symbol for use in a single Hidden Markov Model (HMM) or HMM chain. After classification, different neural data is delegated to local linear predictors when the monkey’s arm is moving and when it is at rest. This multiple-model approach helped to validate the hypothesis that by switching the neuronal firing data, the performance of the final linear prediction is improved.

In this paper, we take the idea of using multiple models a step further and apply the concept to our actual switching classifier. This new structure uses an ensemble of single neural-channel HMM chains to form an Independently Coupled Hidden Markov Model (ICHMM). Consequently, this classifier takes advantage of the neural firing properties and allows for the removal of Vector Quantization while jointly improving the classification performance and the subsequent linear prediction of the trajectory.

1. INTRODUCTION

The novel area of Brain Machine Interfaces (BMI) seeks to link the human brain directly to the external world in the hopes of bringing mobility to those who are paralyzed. To achieve this ultimate goal, focus has been placed on developing linear and non-linear prediction models that can map the neural firings of an animal to a robotic prosthetic [1, 11, 2]. Generally in this type of experiment, neural data is recorded from a primate’s motor cortex as it engages in a movement task (Figure 1). Once a prediction model has been trained with the trajectory/neural data, neural data is solely used to control a robotic arm in real-time [1].

Our group’s work with primates has revealed that in tasks that contain stationary hand locations (e.g. food reaching tasks) the linear and non-linear models are inaccurate, whereas during movement periods they reasonably approximate the desired trajectory [7, 2]. Basically, there is a need to differentiate when the arm is moving or is at rest. Mason and Birch [12] also state that there is a need to instruct the BMI when a movement command is desired since the motor cortex is always active. Additionally, these accuracies could become exacerbated if a single linear prediction model is generalized over an array of movement tasks.

We believe using multiple models for BMIs is one step towards overcoming these issues. Specifically, we focus on using multiple linear prediction models to map discrete portions of the neural data to respective portions of the trajectory (see Figure 2). For this segmentation, we use the two most basic partitions of movement and rest, since they prove difficult for our group’s trajectory reconstruction. Consequently, the individual linear prediction models only learn a segment of the trajectory, outperforming single linear models that must generalize over the full trajectory [7].

In order to switch between these two different “states” of the trajectory, we employ a classifier that delegates neural/trajecory data of the partitions (or classes) to respective linear prediction models. This is different from previous work done in brain computer interfaces (BCI) where the switch simply turns the system on and off [12].

Our initial attempt to develop this switching classifier used vector quantization to compress the multi-dimensional neural data into a discrete symbol for use in a single Hidden Markov Model (HMM) or HMM chain (one per class) [9]. This state-spaced classifier helped to validate the hypothesis that by switching the neuronal firing data, the performance of the final linear prediction is

This paper was supported by DARPA project #N66001-02-C-8022 and NSF project #0540304.
improved.

In this paper, we take the idea of using multiple models a step further and apply the concept to our actual switching classifier. This new structure uses an ensemble of single neural-channel HMM chains to form an Independently Coupled Hidden Markov Model (ICHMM). Consequently, this classifier takes advantage of the neural firing properties and removes the distortion associated with Vector Quantization (VQ) while jointly improving the classification performance and the subsequent linear prediction of the trajectory.

The development and evaluation of the ICHMM, within a bi-model mapping framework, serves as the core of this paper and directs the following organization. First, we discuss our motivation and technique for the ICHMM. Second, we compare new results to our previous results, for both classification and overall mapping performance. Finally, we discuss the results and suggest areas of future work.

2. APPROACH

2.1. Experimental Data

Within the neurological community, the question of whether the motor cortex encodes the arm’s velocity, position, or other kinematic encodings (joint angle, muscle activity, muscle synergy, etc), continues to be debated [8, 13, 11]. For our experimental data, the monkey’s arm is motionless in space for a brief amount of time while it reaches for food or places food in its mouth. During this time of “active holding” it is unknown if the brain is encoding information to contract the muscles in the holding position.

Our work as well as other research has shown this active holding encoding is likely [21]. Due to this belief, we include it as part of the movement class for our classifier [9]. An example of this type of data is shown in Figure 2, along with the superimposed gray-scale colors representing the monkey’s arm movement on the three Cartesian coordinate axes (x, y, and z). Note that we label the movement and rest classes by ‘hand’ from the 10Hz (100ms) trajectory data.

The corresponding neural data is a sparse time series of discrete firing counts that were extracted from the dorsal premotor cortex (PMd), primary motor cortex (MI), and posterior parietal cortex (PP) of an owl-monkey [11, 1]. Each firing count represents the number of neural firings in a 100ms span of time, which is consistent with methods used within the neurological community [8, 13, 14]. Our particular data set contains 104 neural channels recorded for 38.33 minutes. This time recording corresponds to a dataset of 23000x104 time bins.

2.2. Motivation for the ICHMM

Our previous classifier converted the multi-dimensional neural data into a discrete symbol for the discrete-output HMM’s [9]. We used the Linde-Buzo-Gray (LBG) VQ algorithm [5] to do this conversion but only achieved maximally an 87% classification accuracy when combined with the HMM [9].

In trying to improve our previous performance, we used different neural subsets to eliminate irrelevant neurons and retain useful ones. To quantify this differentiation, we examined how well an individual neuron can classify movement vs. rest when trained and tested on an individual HMM chain. Since each neural channel is binned into a discrete number of spikes per 100ms, we were able to directly use the neural data as input.

During the evaluation of these particular HMM chains, we compute the conditional probabilities \( P(O^{(i)}|\lambda^{(m)}_m) \) and \( P(O^{(i)}|\lambda^{(r)}_r) \) for the \( i \)-th neural channel where \( O^{(i)} \) is the respective observation sequence of binned firing counts and \( \lambda^{(m)}_m, \lambda^{(r)}_r \) represent the given HMM chain parameters for the class of movement and rest, respectively. To give a qualitative understanding of these weak classifiers, we present in Figure 3 the probabilistic ratios from 14 single-channel HMM chains (shown between the top and bottom movement segmentation) that produced the best classifications individually. Specifically, we present the ratio

\[
\frac{P(O^{(i)}|\lambda^{(m)}_m)}{P(O^{(i)}|\lambda^{(r)}_r)} \quad (1)
\]

for each neural channel in a grayscale gradient format. The darker bands represent ratios larger than one and correspond to a higher probability for the movement class. Lighter bands represent ratios smaller than one and correspond to a higher probability for the rest class. The conditional probabilities nearly equal to one another show up as grey bands, indicating that classification for the movement or rest classes is inconclusive.

Overall, Figure 3 illustrates that the single-channel HMMs can roughly predict movement and rest segments from the neural data. In Figure 4, they even seem tuned to certain parts of the trajectory like rest-food, food-mouth, and mouth-rest. Specifically, we

<table>
<thead>
<tr>
<th>Neuron #</th>
<th>Rest (%)</th>
<th>Moving (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>83.4%</td>
<td>75.0%</td>
</tr>
<tr>
<td>62</td>
<td>80.0%</td>
<td>75.3%</td>
</tr>
<tr>
<td>8</td>
<td>72.0%</td>
<td>64.7%</td>
</tr>
<tr>
<td>29</td>
<td>63.9%</td>
<td>82.0%</td>
</tr>
<tr>
<td>72</td>
<td>62.6%</td>
<td>82.6%</td>
</tr>
</tbody>
</table>
where the rest segments and darker bands chain HMM the posterior of the state sequence is:

$$P(S^{(1)}|O) = \prod_{d=1}^{D} P(s_{d,1}^{(d)}|o_{d,1}^{(d)}) \prod_{t=2}^{T} P(s_{d,t}^{(d)}|o_{d,t}^{(d)}) \prod_{i=1}^{D} P(r_{i}^{(d)}|s_{i-1}^{(i)}),$$

(2)

where the $d$th chain is represented as a superscript variable, such that $p_{s_{d,t}^{(d)}|o_{d,t}^{(d)}}$ is the probability output given chain $d$’s state and $P(r_{i}^{(d)}|s_{i-1}^{(i)}$ is the probability of a state in chain $d$ given a previous state in chain $e$ [10].

Unfortunately, the complexity ($O(TN^{2D})$ or $O(TD^{2}N^{2})$) and number of parameters necessary for the CHMM (and variants) grows exponentially with the number of $D$ chains (i.e. neural channels), $N$ states and $T$ observation length. For our particular dataset, we would not have enough data to adequately train this type of classifier. Therefore, we must devise a switching classifier that supports the underlying biological system as well as make sure that it can successfully train and classify the neural data.

We look to the neural science community for a solution that can overcome these shortcomings and still support the underlying biological system. The literature in this area explains that different neurons in the brain modulate independently from other neurons [17] during the control of movement. Specifically, during movement, different muscles may activate for synchronized directions and velocities, yet, are controlled by independent neural masses or clusters in the motor cortex [17, 14, 16].

Conversely, within the neural clusters themselves, temporal dependencies (and co-activations) have been shown to exist [17]. Therefore, for our classifier, we make the assumption that enough neurons are sampled from different neural clusters to avoid overlap or dependencies. We can further justify this assumption by looking at the correlation coefficients (CC) between all the neural channels in our data set.

The best CCs (0.59, 0.44, 0.42, 0.36) occurred between only four of the possible 10,700 neural pairs while the rest of the neural pairs were a magnitude smaller (abs(CC)<0.01). We believe this indicates weak dependencies between the neurons in our particular dataset. Additionally, despite these possible weak underlying dependencies, there is a long history of making such independence assumptions in order to create models that are tractable or computationally efficient. The Factorial Hidden Markov Model is one example amongst many [15, 10].

By making an independence assumption between neurons, we can treat each neural channel HMM independently. Therefore the joint probability

$$P(O_{1}^{(1)}, O_{2}^{(2)}, \ldots O_{T}^{(D)}, \lambda_{full})$$

becomes the product of the marginals

$$\prod_{i=1}^{D} P(O_{T}^{(i)}|\lambda^{(i)})$$

(4)

of the observation sequences (each length T) for each $d$th HMM chain $\lambda$. Since the marginal probabilities are independently coupled, yet try to model multiple hidden processes, we name this classifier the Independently Coupled Hidden Markov Model (ICHMM) Figure 5.

By using an ICHMM instead of a CHMM (Figure 5), the overall complexity reduces from ($O(TN^{2D})$ or $O(TD^{2}N^{2})$) to $O(DTN^{2})$ given that each HMM chain has a complexity of $O(TN^{2})$. Additionally, since we are using a single HMM chain to train on a single neural channel, the number of parameters is greatly reduced and can support the amount of training data. Specifically, the individual HMM chains in the ICHMM contain around 70 parameters for a training set of 10,000 samples as opposed to almost 18,000 parameters necessary for a comparable CHMM (due to the dependent states).

The detailed ICHMM structure is as follows:

1. Using a single neural channel $d$, we evaluate the conditional probabilities $P(O_{T}^{(d)}|\lambda_{M}^{(d)}$ and $P(O_{T}^{(d)}|\lambda_{M}^{(d)}$ , where,

$$O_{T}^{(d)} = \{O_{t-2}^{(d)}, O_{t-1}^{(d)}, O_{t}^{(d)}\}, \quad T > I$$

(5)
and \( \lambda^{(d)}_i \) and \( \lambda^{(g)}_i \) denote HMM chains that represent the two possible states of the monkey’s arm (moving vs. rest). We previously trained all of the HMM chains with their respective neural channels using the Baum-Welch algorithm [6, 4]. Specifically, we use three hidden states and an observation sequence length \( T \) of 10, which corresponds to a second of data (given the 100ms bins).

2. Normally, we would decide that the monkey’s arm is at rest if, \( P(O_T|\lambda_r) > P(O_T|\lambda_M) \) and is moving if, \( P(O_T|\lambda_M) > P(O_T|\lambda_r) \), but since we want to combine the predictive powers of all the neural channels, we use Equation (4) to produce the decision boundary:

\[
\prod_{i=1}^{D} P(O^{(i)}_T|\lambda^{(i)}_M) > \prod_{i=1}^{D} P(O^{(i)}_T|\lambda^{(i)}_r)
\]

or more aptly,

\[
l(O) = \frac{\prod_{i=1}^{D} P(O^{(i)}_T|\lambda^{(i)}_M)}{\prod_{i=1}^{D} P(O^{(i)}_T|\lambda^{(i)}_r)} > \zeta
\]

where \( l(O) \) is the likelihood ratio, a basic quantity in hypothesis testing [20, 18]. Essentially, ratios greater than the threshold \( \zeta \) are classified as movement and those less than \( \zeta \) as the rest class. The use of thresholds for the likelihood ratio has been used in neural science and other areas of research [20, 18]. Often, it is more common to use the log-likelihood ratio instead of the likelihood ratio for the decision rule so that a relative scaling between the ratios can be found (as well as stymieing unruly ratios) [18]:

\[
\log(l(O)) = \log\left(\frac{\prod_{i=1}^{D} P(O^{(i)}_T|\lambda^{(i)}_M)}{\prod_{i=1}^{D} P(O^{(i)}_T|\lambda^{(i)}_r)}\right) > \zeta
\]

We see that by applying the log to the product of the likelihood ratios, we are essentially finding the sum of the log likelihood ratios to see if it is larger or smaller than a threshold (log \( \zeta \)). In simple terms, this decision rules poses the question of how large is the probability for one class compared to the other and is it occurring over a majority of the single classifiers. Note that by varying the threshold \( \log(\zeta) \), we can tune classification performance to fit our particular requirements for choosing one class over another. For this experiment we require equal classifications for each class (no bias for one or another). Moreover, optimization of the classifier is now no longer a function of the individual HMM evaluation probabilities, but rather a function of overall classification performance.

2.4. Bimodel Structure and Training

We now want to generate an overall mapping between the neural data and 3-D arm position by combining the ICHMM’s output with the multiple linear predictors. To complete this bimodel framework, we first train an ICHMM on the neural data to generate the decision boundaries for the two classes: movement and rest. We then assign a single linear prediction model to each class. Finally, based on this class assignment, we use the ICHMM to select different neural/trajectory data for each respective linear model (as shown in Figure 6).

During training, we first define a linear model with 10 time delays (1 sec), and 3 outputs so that its weight vector has 3,120 elements for the 104 neural channels (one per class). We then train each linear model for 100 cycles on a set of 10,000 bins (1,000 sec.) of data. Additionally, each model adapts its weights using normalized least mean square (NLMS) with a learning rate of 0.03 [19].

After training, we fix all model parameters and test on 2,988 bins of neural data with the model to predict hand positions. During this testing phase, neural inputs are first fed to a trained ICHMM classifier, which again acts as a switching function for the linear models. Based on the decision rule in Equation (7), one of the two selected linear models generates the predicted continuous 3-D arm position from the selected neural data.

3. RESULTS

3.1. Performance of the ICHMM

Since the threshold \( \zeta \) (Equation 7) determines the best classification, we need to first select a \( \zeta \) that will generate the best results. Although the standard receiver operator characteristic (ROC) curve can determine the \( \zeta \) thresholds, they fail to convey an explicit relationship between the classes and the threshold when optimizing performance. The “optimal” operating point on the ROC curve is the point closest to the ideal upper left-hand corner. This ROC operating point only provides the true positives and false negatives of the overall classifier. We use the true positives and true negatives of the respective classes to provide an explicit inter-class relationship. For nomenclature simplicity, we label these two curves the likelihood ratio operating characteristic (LROC) curves since they represent the quantities in the likelihood ratio. The “optimal” operating point on the LROC occurs when the two curves of the classes
Fig. 7. LROC Curves for classifier using random data

Fig. 8. LROC Curves for ICHMM on testing data and training data

Fig. 9. Z-coordinate of predicted trajectory of a monkey’s arm

intersect since this intersection represents equal classification for the two classes.

Figure 7 shows an example LROC plot of an ICHMM trained and tested on random neural data. This data is created by uniformly randomizing the spikes of a neural channel so that we retain the firing rate statistical properties but remove temporal structure. From the plot, we notice that as the $\zeta$ threshold (X-axis) is manipulated, classification performance, or the true positives, for the movement and rest classes shifts proportionately (Y-axis). We also see that the joint maximum, or equilibrium point, occurs near the area where the likelihood ratio threshold equals one (signifying equal classifications for the two classes). Specifically, the equilibrium point in Figure 7 illustrates that we can only correctly classify the movement and rest classes around 50% (joint maximum). Having both a $\zeta$ threshold equal to one and 50% classification is the expected when using randomized data for two classes.

In Figure 8A, the LROC curves show that the ICHMM is a significant improvement in classification over our previous VQ-HMM classifier. This is evident from the equilibrium point showing that the movement and rest classifications occur around 93% as opposed to 87% in our previous work. Note that without the $\zeta$ threshold, the results do not show any important significance (except better than random).

In Figure 8B we see that a similar $\zeta$ threshold ($\zeta = 1.0048$) from the testing set Figure 8A is retrievable from training set ($\zeta = 1.0044$). Assuming that the data is somewhat stationary, these results indicate that we can retrieve a reliable threshold solely from training data.

3.2. Bimodel Prediction Results

In this section, we report results for the neural mapping of a single linear predictor, the bimodel system using the VQ-HMM switching classifier and using the ICHMM switching classifier. Figure 9 shows the predicted hand trajectories of each modeling approach, superimposed over the desired (actual) arm trajectories for the test data; for simplicity, we only plot the trajectory along the Z-coordinate. Qualitatively, we can see that the ICHMM performs better than the others in terms of reaching targets. Overall, prediction performance of the ICHMM classifier is slightly better than the VQ-HMM classifier, and superior to the single linear predictor, as evidenced by the average correlation coefficients (across the three coordinates) of 0.64, 0.80, and 0.86 for the single linear predictor, the bimodel system with VQ-HMM and the bimodel system with ICHMM. Additionally, using 4-sec time windows (because movements take approximately 4 secs) the mean of the Signal to Error Ratio (SER) averaged over all dimensions for the single linear predictor, the bimodel system with VQ-HMM, and the bimodel system with ICHMM are -20.3 dB, -15.0 dB, and -12.2 dB respectively, while the maximum SERs achieved by each model are 10.4 dB, 24.8 dB, and 30.2 dB respectively.

4. DISCUSSION AND FUTURE WORK

Our results show that the final prediction performance of the bimodel system using the ICHMM is much better than using the VQ-HMM, and superior to that of a single linear predictor. Moreover, it lends support to our independence assumption between the neurons and approximates the joint probability (eq 4) with the product of the marginals (Equation 4). Overall, the ICHMM produces good results with few parameters.

The one caveat to the ICHMM is the reliance on the $\zeta$ threshold. Fortunately, this threshold is retrievable from the training set. Interestingly, the $\zeta$ threshold can be viewed as global weighting for the two classes in this system. If we frame the ICHMM as mixture of experts (ME) perhaps boosting or bagging could be used to locally weight these simplistic classifiers in future work. The ME generates complex and powerful models by combining simpler models that often map different regions of the input space [3]. With boosting, models are weighted to create a strong ensemble decisions so that a weighted majority “votes” for the appropriate class labeling [3]. This is analogous to what the ICHMM currently does with a global $\zeta$ weighting or biasing.

To lend further support for localized weighting, we used neural ranking information to see if we could observe the importance of the single-neural HMM chains on the overall classifier (Figure 10). We start with the top-ranked neural HMM chains, and then continually add chains from best to worst (in individual class-
Fig. 10. Neural HMM chain adding experiment from best to worst
classification performance) into an ICHMM and plot the overall classification performance of that ICHMM (Y-axis). Specifically, the ICHMM grows from a one HMM chain to finally including all of the HMM chains (X-axis). In this experiment we use the ζ threshold that jointly maximizes classification between both classes for each ICHMM generated. We see that despite adding "weaker" individual neural HMM chains, there is a continued improvement in ICHMM classification results. Also of interest, the ζ threshold (in light gray) continues to converge as we add more and more single-neural HMM chains. This might be explained by looking back at Figure 4, where different neurons tune to different portions of the trajectory. Perhaps the overlap across the individual chains provides increased classification on the full trajectory (for the ICHMM).

Finally, another area of improvement may come from the training data itself. Since we hand segmented the classes, we believe this may be a source of error. In future work, we want to investigate unsupervised methods of segmenting the classes. We hope this segmentation would also increase the repertoire of "motion" classes to allow the linear or non-linear predictors to fine tune to a specific class for improved 3-D mapping.

5. REFERENCES


[17] Walter J. Freeman, Mass Action in the Nervous System University of California, Berkeley, USA

[18] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, Raj Reddy, Spoken Language Processing: A Guide to Theory, Algorithm and System Development

