

HIERARCHICAL CLUSTERING OF NEURAL DATA USING LINKED-MIXTURES OF HIDDEN MARKOV MODELS FOR BRAIN MACHINE INTERFACES

Shalom Darmanjian¹, Jose Principe¹

¹Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611-6200

ABSTRACT

In this paper, we build upon previous Brain Machine Interface (BMI) signal processing models that require a-priori knowledge about the patient's arm kinematics. Specifically, we propose an unsupervised hierarchical clustering model that attempts to discover both the interdependencies between neural channels and the self-organized clusters represented in the spatial-temporal neural data. Given that BMIs must work with disabled patients who lack arm kinematic information, the clustering work describe within this paper is very relevant for future BMIs.

Index Terms—Hidden Markov Model, Brain Machine Interface, Hierarchical Clustering, Neural Spike Data, Neural Data Clustering.

1. INTRODUCTION

Generally, a BMI is a system that directly retrieves neuronal firing patterns from dozens to hundreds of neurons in the brain and then translates this information into desired actions in the external world. In particular, the neural data is recorded from one or more cortices using a single electrode or multiple electrode grid arrays [11]. The amplified analog voltages recorded from one or more neuron is then digitally converted and passed to a spike sorting algorithm. Finally, the discrete binned spike counts are fed into to a signal processing algorithm [11] and subsequent trajectory/lever predictions are sent to a robot arm or display device. All of the processing occurs as an animal engages in a behavioral experiment (lever press, food grasping, finger tracing, or joystick control).

With respect to the BMI signal processing, recent research has demonstrated that neural state structures exist as animal subjects engage in movement [3, 5, 17]. By finding partitions in the neural input space that correspond to specific motor states (hand moving or at rest), trajectory reconstruction is improved when the models are constructed with data from only one partition. [5, 2, 17]. Specifically, a 'switching' generative model partitions the animal's neural firings and corresponding arm movements into different motion primitives [2, 4, 5]. This improvement is primarily due to the ability of the generative models to homogenous the neural data so that the continuous filters specialize in a particular part of the input space rather than generalize over the full space. This work also demonstrated that modeling is improved when dependencies are exploited between neural channels [4, 5].

Unfortunately, moving to the clinical research setting complicates the modeling paradigm since kinematic information is not available from disabled patients. Without the kinematic clues to label the different motion primitives in the final BMI setting, future models must self-organize to exploit this neural input space.

In this paper, we propose an unsupervised hierarchical clustering model that attempts to discover both the interdependencies between

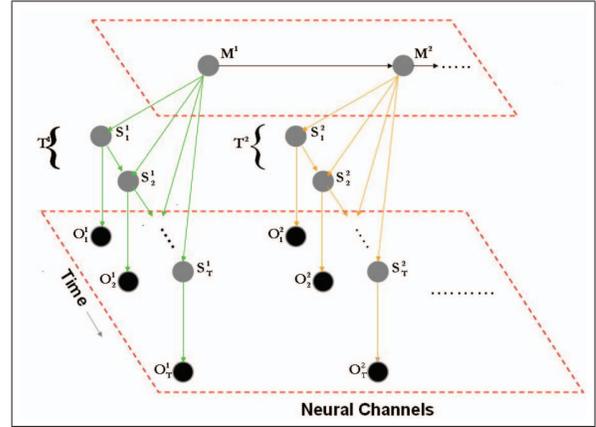


Fig. 1. LM-HMM graphical model

neural channels and the self-organized clusters represented in the spatial-temporal neural data. Section two provides the framework for the proposed method. Section three details the experimental animal data while section four provides the clustering results on the animal data. Finally, section five offers a discussion on the model and results.

2. CLUSTERING FRAMEWORK AND TRAINING

Although there is evidence for multiple neural structures that correspond to motion primitives, we need a way to find them with an automatic procedure or clustering method [2, 3]. Additionally, the clustering model must encapsulate the spatial-temporal dependencies since it has shown to be important [5]. Therefore, we look to the Linked-Mixture of Hidden Markov Models (LM-HMM) as a way to help cluster the class labels while also clustering spatial dependencies between channels.

The log likelihood of the dynamic neural firings from all of the neurons for this structure (Figure 1) is

$$\begin{aligned} \log P(O|S, M, \Theta) &= \log P(M^1) + \sum_{i=2}^N \log P(M^i | M^{i-1}, \Theta) \\ &+ \sum_{i=1}^N \left(\sum_{t=1}^T \log P(O_t^i | S_t^i, \Theta^i) \right) \\ &+ \sum_{t=1}^T \log P(S_t^i | S_{t-1}^i, M^i, \Theta^i) \end{aligned} \quad (1)$$

Where the dependency between the tree cliques T^j are represented by a hidden variable M in the second layer,

$$P(M^i | M^{i-1}, \Theta^i) \quad (2)$$

and the hidden state sequence S also has a dependency on the hidden variable M in the second layer

$$P(S_t^i | S_{t-1}^i, M^i, \Theta^i) \quad (3)$$

From Figure 1, we see that the third layer of observable variables O^i are conditionally independent from the first layer of hidden variables M^i , as well as the sub-graphs of the other neural channels T^j (where $i \neq j$). The hidden variable M in the second layer of Equation 2 can be interpreted as a mixture variable (when excluding the hierarchic links).

The LM-HMM is a compromise between making an independence assumption and assuming dependencies across all channels. For further understanding and EM implementation please see [5]. Although this hierarchical model can find dependencies between channels in an unsupervised way, we still need to address how to obtain the class labels without user intervention.

The use of hidden Markov models for clustering sequences appears to have first been mentioned in Juang and Rabiner [12] and subsequently used in the context of discovering subfamilies of protein sequences in Krogh et al [1]. Other work uses single HMM chains for understanding the transition matrices [1, 13, 9]. Our work differs since this model finds unsupervised hierarchal dependencies between HMM chains (per neuron) while also clustering the data. Essentially we are refining the model parameters and the structure as we cluster.

Let data set D consist of N sequences for J neural channels, $D = S_1^1, \dots, S_N^J$, where $S_n^j = (O_1^j, \dots, O_T^j)$ is a sequences of observables length T and $\Lambda = (\lambda_1, \dots, \lambda_K)$ a set of Models. Our goal is to locally maximize the log-likelihood function:

$$\log P(D|\Lambda) = \sum_{S_n^j \in D} \log P(S_n^j | \lambda_{y(S_n^j)}) \quad (4)$$

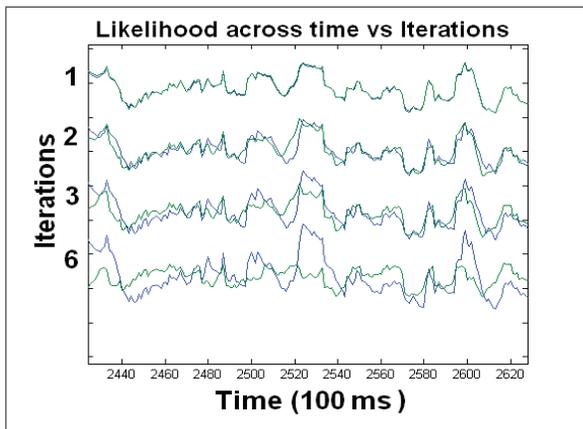


Fig. 2. Likelihoods through time vs. iterations

In effect, we are trying to discover a natural grouping of the sequences S into K clusters. This is similar to K-means except the vector centroids are now probabilistic models representing dynamic temporal data [10]. Since we are clustering the sequences

from multiple channels, the underlying clusters or probabilistic models to which they belong must be inferred. This is difficult since the LM-HMMs can have a varying number of states, the sequences can vary in length and the choice of a distance metric for the models is not straightforward.

Assumptions based on prior results are used to alleviate some of these model initialization problems [2]. Specifically, we make a-priori assumptions that the neural channels are of the same sequence length and same number of hidden states. Below we outline the clustering framework:

1. Randomly assign K LM-HMMs, one for each observation sequence S_n , $1 \leq n \leq N$. The LM-HMM parameters are initialized randomly.
2. Train each assigned model with the respective sequences using the LM-HMM procedure discussed in [5]. During this step the model learns the dependency structure for the current cluster of sequences.
3. For each model evaluate the log-likelihood of each of the N sequences give model λ_i , i.e., calculate $L_{in} = \text{Log } L(S_n | \lambda_i)$, $1 \leq n \leq N$ and $1 \leq i \leq K$. $y(S_n) = \text{argmax}_y \text{Log } L(S_n | \lambda_i)$ is the cluster identity of the sequence. Then re-label all the sequences based on cluster identity which in turn maximizes eq 4.
4. Repeat steps two and three until convergence occurs or until a percentage of labeled data does not change (i.e. set a threshold for changing samples). More advanced metrics for deciding when to stop clustering could be used.

Although we start with noisy estimates of the model parameters by fitting the K models with sequences that may be from different models, eventually the parameters should cluster into K groups about their true values as the iterations progress. The Log-likelihoods are a natural way to provide distances between models rather than clustering in the parameter space (which is unknown). Essentially, during each round, each training example is re-labeled by the winning model with the final outcome retaining a set of labels that relate to a particular cluster or neural state structure for which spatial dependencies have also been learned.

3. EXPERIMENTAL ANIMAL DATA

In the clustering experiments we used two animal data sets. For the first data set, thirty-two microwire electrodes were implanted uni-

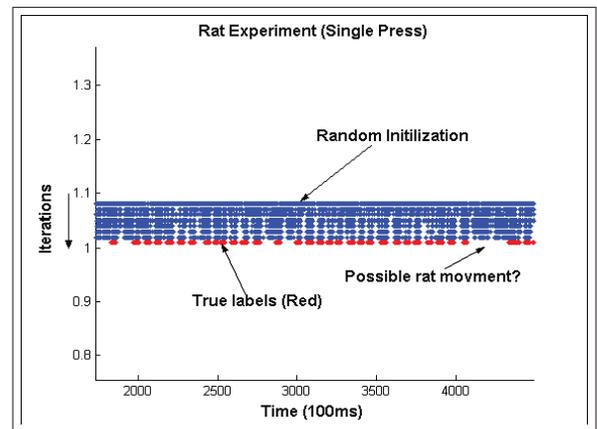


Fig. 3. Rat Clustering Experiment, One lever, Two Classes

Table 1. Classification Results of Lever Presses

Experiment Type	Class 1	Class 2
Unsupervised	65.2	67.4%
Supervised	62.1	63.3%

laterally in the forelimb region of primary motor cortex of a male Sprague-Dawley rat [14]. The task involved an LED visual stimulus to press a single lever for a minimum of 0.5s to achieve a water reward. Essentially this go-no-go experiment includes neural data along with lever presses. This particular data set contains 16 neurons yielding 13000x16 time bins. With each time being the spike sorted count per 100ms of a particular neuron (sixteen in this data set).

For the second rat data set, two 16-microelectrode arrays in the forelimb regions of each hemisphere [6]. The task involved an LED visual stimulus to press one of two levers for a minimum of 0.5s to achieve a water reward. Essentially this go-no-go experiment also provides neural data along with lever presses. This particular data set contains 42 neurons yielding 19000x42 time bins (100ms each).

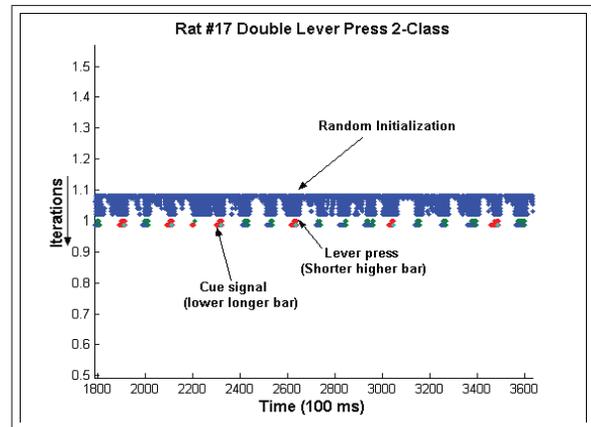
4. RESULTS

For the first experiment, 5000 data points of the single lever press experiment were used for clustering. The initial cluster labels were randomly selected for the two classes (lever press and non-lever-press) across the full time series. We took into consideration two important points. First, we ran multiple Monte Carlo simulations in order to verify our results. Second, we empirically determined the parameters that needed to be initialized. These include:

1. Observation Sequence length
2. Number of states
3. Number of clustering rounds

For the observation length, we varied it from 5 to 15 which correspond to .5 seconds to 1.5 seconds. We varied the number of hidden states from 3 to 5. Finally, we varied the number of rounds from 4 to 10. After exhausting the number of possible combinations we found a length of 10 for observation length along with 3 states and 6 rounds to be acceptable (since less than 5% of the labels changed after small number of iterations). We set these parameters for all neural channels.

Figure 2 shows the likelihoods computed on the single-lever press at each time step during a particular clustering iteration (iterations 1,2,3 and 6). Notice how initially the class likelihoods are very similar but after the next round more discrimination between classes is possible. Figure 3, demonstrates that the hierarchical clustering applied to these likelihoods gives a somewhat reasonable clustering for the two classes in the single lever press. In terms of quantification, Table 1 shows that the hierarchical clustering is able to correctly classify each class around 66% of the time. Of course the models do not know which classes the samples belong. We use the ground truth of the lever presses to determine the accuracy of the clustering. It is remarkable that the results are similar to the supervised classification (and slightly better for this particular data-set). For the supervised classification we use Adaline/RNN/HMM results for this data set (with similar results obtained between the different classifiers). We see that the clustering fails in certain locations. We believe this is due to the clinical experiment since the rat moves around the cage without being recorded. This will be addressed further in the discussion section.

**Fig. 4.** Rat Clustering Experiment, Two levers, Two Classes

For the second experiment, 10000 data points of the two-lever rat experiment were used for clustering. All of the data points were randomly labeled for the two classes. Parameters were also empirically selected. For the following double-lever experiments there are cue-times included and rewards. Figure 4 shows where the cue signals are and the rewards as well as the lever presses (red is left, green is right). For example on the fifth cue-signal the rat was supposed to press left but instead pressed right (as indicated by colors on the plot).

From figure 4, we see that there are repeating patterns between classes. Unfortunately, they are not comparable to the experiment with a single lever press.

Figure 5 shows the results when clustering the data for the two-lever press into three classes. It is interesting that we see that there are some repeating patterns. The consistencies seem to be that the clusters transition from class to another (like red to green or green to blue) and appears to correspond to a kinematic/experimental event(i.e. lever pressed, cue signal, reward, etc).

As the number of clusters is increased, similar results are obtained. Specifically, there are repeating patterns when looking at the transitions from one class to another before and after the lever presses. Across all of the double-lever press experiments quantification (in terms of classification) is poor. The next section provides a possible reason for this poor performance.

5. DISCUSSION

Despite these encouraging results, we believe that substantial improvements in the hierarchical clustering are possible. First, we are missing most of the kinematic information from the experiment. Sometimes the rat grooms during the experiments or walks away from the lever or moves towards the water reward. Even how the animal presses the lever is inconsistent since it uses different paws and presses different locations on the levers (without this information being recorded). This also helps to explain why the supervised classification had such poor results (since labels aren't true ground truths).

Second, the LM-HMM in the hierarchical clustering framework may not be taking full advantage of the dynamic spatial relationships that may be evolving through time between the neurons. Although the hierarchical training methodology does create dependencies between the HMM experts, perhaps there are better ways to exploit the

dependencies or aggregate the local information. This is most certainly true when considering that different neural processes that are interacting with other neural processes in an asynchronous fashion.

One way to see if the clustering results are providing any meaningful information is to see how the final reconstruction of a kinematic trajectory is improved. Obviously that will have to be done with other data where full kinematic information about the clinical experiment is provided.

As a final point, there was an interesting effect from the rat experiments. Looking closely at some of the results, repeating patterns of transitions from one class to another exists. For example there may be a consistent transition from class 1 to class 3 and class 2 to class 1. It would be interesting to investigate this further and see if perhaps there is a switching behavior between stationary points in the input space. Perhaps we can ascertain when a stationary switching point has occurred and exploit that information for multiple modeling.

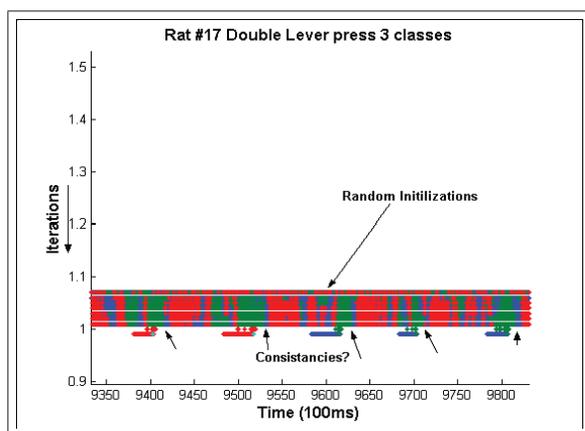


Fig. 5. Rat Clustering Experiment, Two levers, Three Classes

6. ACKNOWLEDGEMENTS

We thank Justin Sanchez and Jack DiGiovanna for sharing their rat data and experience. This paper was supported by DARPA project #N66001-02-C-8022 and NSF project #0540304.

7. REFERENCES

- [1] I. Cadez and P. Smyth, "Probabilistic Clustering using Hierarchical Models," Technical Report No. 99-16 Department of Information and Computer Science University of California, Irvine
- [2] S. Darmanjian, S. P. Kim, M. C. Nechyba, S. Morrison, J. Principe, J. Wessberg, and M. A. L. Nicolelis, "Bimodel Brain-Machine Interface for Motor Control of Robotic Prosthetic," *IEEE Int. Conf. on Intelligent Robots and Systems*, pp. 112-116, 2003.
- [3] S. Darmanjian, S. P. Kim, M. C. Nechyba, J. Principe, J. Wessberg, and M. A. L. Nicolelis, "Bimodel Brain-Machine Interface for Motor Control of Robotic Prosthetic," *IEEE Machine Learning For Signal Processing*, pp. 379-384, 2006.
- [4] S. Darmanjian, A. R. C. Paiva, J. C. Principe, M. C. Nechyba, J. Wessberg, M. A. L. Nicolelis, and J. C. Sanchez, "Hierarchical decomposition of neural data using boosted mixtures of independently coupled hidden markov chains," *International Joint Conference on Neural Networks*, pp. 89-93, 2007.
- [5] S. Darmanjian and J. Principe, "Boosted and Linked Mixtures of HMMs for Brain-Machine Interfaces," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, Article ID 216453, 12 pages doi:10.1155/2008/216453
- [6] J. DiGiovanna, J. C. Sanchez, and J. C. Principe, "Improved Linear BMI Systems via Population Averaging," presented at IEEE International Conference of the Engineering in Medicine and Biology Society, New York, pp. 1608-1611, 2006.
- [7] Donoghue, J. P. and S. P. Wise (1982). "The motor cortex of the rat: cytoarchitecture and microstimulation mapping." *J. Comp. Neurol.* 212(12):
- [8] Luis Goncalves, Enrico Di Bernardo and Pietro Perona, *Movemes for Modeling Biological Motion Perception Book Series Theory and Decision Library Volume Volume 38 Book Seeing, Thinking and Knowing Publisher Springer Netherlands*
- [9] Kemere C, Santhanam G, Yu BM, Afshar A, Ryu SI, Meng TH, Shenoy KV (2008) Detecting neural state transitions using hidden Markov models for motor cortical prostheses. *Journal of Neurophysiology.* 100:2441-2452
- [10] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849-856. MIT Press, 2002.
- [11] M. A. L. Nicolelis, D.F. Dimitrov, J.M. Carmena, R.E. Crist, G. LeHew, J. D. Kralik, and S.P. Wise, "Chronic, multisite, multi-electrode recordings in macaque monkeys," *PNAS*, Vol. 100, No. 19, pp. 11041 - 11046, 2003.
- [12] B.H Juang, and L. R. Rabiner, "Issues in using Hidden Markov models for speech recognition," *Advances in speech signal processing*, edited by S. Furui and M.M. Sondhi, Marcel Dekker, inc., pp. 509-553, 1992
- [13] Radons G, Becker JD, Dulfer B, Kruger J. Analysis, classification, and coding of multi-electrode spike trains with hidden Markov models. *Biol Cybern* 71: 359-373, 1994.
- [14] J. C. Sanchez, J. C. Principe, and P. R. Carney, "Is Neuron Discrimination Preprocessing Necessary for Linear and Nonlinear Brain Machine Interface Models?," accepted to 11th International Conference on Human-Computer Interaction, vol. 5, pp. 1-5, 2005.
- [15] A. B. Schwartz, D. M. Taylor, and S. I. H. Tillery, "Extraction algorithms for cortical control of arm prosthetics," *Current Opinion in Neurobiology*, Vol. 11, pp. 701-708, 2001.
- [16] E. Todorov, *On the role of primary motor cortex in arm movement control*, To appear in *Progress in Motor Control III*, chap 6, pp 125-166, 2003. *Journal of Neurophysiology.* 100:2441-2452
- [17] B.M. Yu, C. Kemere, G. Santhanam, A. Afshar, S.I. Ryu, T.H. Meng, M. Sahani M*, K.V. Shenoy, (2007) Mixture of trajectory models for neural decoding of goal-directed movements. *Journal of Neurophysiology.* 97:3763-3780.