33rd Annual International Conference of the IEEE EMBS
Boston, Massachusetts USA, August 30 - September 3, 2011

A Pilot Study on Two Stage Decoding Strategies

Bo Jiang, Rui Wang, Qiaosheng Zhang, Jicai Zhang, Xiaoxiang Zheng, Ting Zhao*

Abstract—**Brain-machine interfaces (BMIs) use neural activity related to motion parameters to enable brain directly control external devices. Some linear and nonlinear decoding techniques have been used successfully to infer arm trajectory from neural data. Unfortunately, these One stage decoding techniques can hardly get high accuracy and low computational demands at the same time. Here we introduce a Two Stage Model (TSM) which consists of two linear models, on the basis that different motion states have different neural firing patterns when rats were doing the lever pressing task. The accuracies of the neural firing patterns classification were higher than 90% for all the three datasets. The Correlation coefficients (CC) between the trajectory predicted by TSM and the measured one were up to 0.89, 0.85 and 0.95 for the three datasets respectively higher than those of Kalman Filter (KF) and Partial Least Squares Regression (PLSR). The time consumption of TSM was about only 10% of that of Generalized Regression Neural Network (GRNN). These results show that TSM can simultaneously get both high accuracy and low computational cost.**

I. Introduction

THE technology of brain-machine interfaces (BMIs), a means of assisting or repairing human cognitive or sensory-motor functions enable a brain to control an external device directly [1, 2]. It has been demonstrated by sufficient previous studies carried on non-human primate [3, 4] that recoding of neural activities in motor cortex had been successfully made use of to control 2D or 3D cursors, even a 4-DOF robotic arm.

One of the most important components of a BMI system is the decoding algorithm which translates neural activity into desired movements. Many linear and nonlinear algorithms have been successfully applied to neuron decoding such as population vectors (PV) [5], Wiener Filter (WF) [6], Kalman Filter (KF) [7], partial least squares regression (PLSR) [8], and Generalized Regression Neural Network (GRNN) [9]. Linear algorithms are simple models which are computationally inexpensive, but often with low decoding accuracy. Nonlinear algorithms are much more complex models which can gain high decoding accuracy, but they are very time consuming. It is meaningful to build models which can meet both high accuracy and low time consumption requirements.

Recent study find that neurons in the hippocampus have different firing patterns when rats are in different states [10].Similar phenomenon has also been found in premotor cortex when monkeys are in different motion states [11]. Byron M. Yu built a Mixture of Trajectory Models which established different trajectory models for different motion states [12].

In this paper, we introduced a Tow Stage Model (TSM) which can gain high decoding accuracy with low computation expense. TSM classifies neural firing rates into different state patterns in the first stage and then decodes the trajectories of these different states using different regression models. For both the classification and regression algorithms are linear, TSM's time consumption is much more less than nonlinear algorithms'.

II. Methods

A. Behavioral Task and Animal Surgery

We recoded the data from three male Spraue-Dawley rate (weighing 275-300 g), and used protocols that the Animal Care Committee of Zhejiang University approved and followed the Care and Use of Laboratory Animals (China Ministry of Health). In our task, three rats were randomly labeled respectively as R1, R2 and R3. They were trained to perform operant conditional task. Each time a rat was put into an operant chamber, and learned to press a lever down so as to get water rewards. All rats took no water before experiments, and the amount of water as reward restrict to 12 mL during the task. When the success rate of animal task reached more than 75%, 16-channel (2×8) mircrowire arrays (California Fine Wire) were implanted into the forelimb region of rats' primary motor cortex. We placed the electrode tips in the layer V, within depth from 1.1mm to 1.8mm beneath the pia. The spike signals did not be acquired until the rats recovered from the surgery, taking a week or more.

B. Data Acquisition

We used a Cerebus multichannel data acquisition system (Blackrock Inc., USA) to filter 16 channels of neural signal analogly and recorded them at the 30 kHz sampling rate. In our experiment, any 1.6 ms long signal segment that had a peak above 5.5 RMS of the voltage at its 0.4 ms offset was recorded as spike. Pressure on the lever was synchronously recorded at a sampling rate of 500 Hz during the experiment. The neurons on each electrode were offline discriminated using Principal Component Analysis (PCA) and K-Means

Manuscript received March 26, 2011. This work was supported by the National Science Foundation of China (30800287, 61031002) and Zhejiang provincial key science and technology program for international cooperation(No. 2011C14005)

BO Jiang is with Qiushi Academy of Advanced Studies and College of Biomedical Engineering and Instrumental Science, Zhejiang University, Hangzhou, 310027, PR China. (e-mail: jiangbodf@gmail.com)

Rui Wang, Qiaosheng Zhang, Xiaoxiang Zheng, Jicai Zhang are with Qiushi Academy of Advanced Studies, Zhejiang University, Hangzhou, 310027, PR China.

Ting Zhao is with Qiushi Academy for Advanced Studies and College of Biomedical Engineering and Instrumental Science, Zhejiang University, Hangzhou, 310027, PR China. (Phone: +86 571 87953860; fax: +86 571 87951676; e-mail: qaas@zju.edu.cn).

method (spike sorting). Then we binned the neuronal data with a certain bin size (100 ms), and got the synchronized pressure value of each bin by averaging the pressure value in the bin (spike binning).

C. Data Analysis

1) Naive Bayes Classifier:

Naive Bayes Classifier [13] was applied to classify neural signals into two classes which were Press State (PS) and Free State (FS). Suppose $C \epsilon (PS, FS)$ be the random variable denoting the class label, $\boldsymbol{X}(t) = [F_1(t), F_1(t-1) \dots F_1(t-L+1), F_2(t) \dots F_M(t-L+1)]^T$ be a vector of random variables denoting the observed neural firing rates at time bin t. Where $F_k(t)$ is the firing rate of neuron k at time bin t, L is the number of time lags and M is the number of neurons. For convenience of expression we denoted $\boldsymbol{X}$ as $\boldsymbol{X} = [X_1, X_2 \dots X_{M\times L}]^T$. Following the Bayes' rule the probability of each class given the vector of observed neural firing rates can be expressed as:

$$p(C|\boldsymbol{X}) = \frac{p(C)p(\boldsymbol{X}|C)}{p(\boldsymbol{X})} \quad (1)$$

Where $p(C)$ is the *class prior*, $p(\boldsymbol{X}|C)$ is the *likelihood* and $p(\boldsymbol{X})$ is the *evidence*. Since the *evidence* does not depend on C and the values of X_i are given, the *evidence* is constant. The *class prior* can be calculated by *p(C)=(number of training samples in the class)/(total number of training samples)*. Naive Bayes Classifier assumes that each feature X_i is conditionally independent of every other feature X_j for $j \neq i$. Therefore the conditional distribution over the class variable C can be expressed like this:

$$p(C|\boldsymbol{X}) \propto p(C) \prod_{i=1}^{M\times L} p(X_i|C) \quad (2)$$

$p(X_i|C)$ can be represented as a normal distribution with its mean μ_{ic} and standard deviation σ_{ic} calculated from the training data set:

$$p(X_i = v|C) = \frac{1}{\sqrt{2\pi\sigma_{ic}^2}} \mathrm{e}^{-\frac{(v-\mu_{ic})^2}{2\sigma_{ic}^2}} \quad (3)$$

The discrimination function of Naive Bayes Classifier then can be defined as:

$$Class(x_i, \dots, x_{M\times L}) = argmax_c\, p(C=c) \prod_{i=1}^{M\times L} p(X_i = x_i|C=c) \quad (4).$$

2) Decoding Algorithms:

In this paper we decoded the lever pressure from neural ensemble firing rate using Kalman Filter, PLSR, GRNN and Tow Stage Model respectively.

Two Stage Model. We designed a Two Stage Model (figure 1) to decode the lever pressure. In the first stage, neural ensemble firing rates were put into the *Naive Bayes Classifier* and were classified into *Press State* or *Free State*. The second stage includes two models which were PLSR Model and Baseline Model. If the output of the Naive Bayes Classifier is *Press State* then the neural ensemble firing rates are set as the inputs of the PLSR Model and are used to predict the pressure value. Otherwise if the classification result is *Free State* then the Baseline Model is activated and the predicted pressure is set to the baseline value.

One Stage Models for Comparison. Kalman Filter, Partial Least Squares Regression and Generalized Regression Neural Network are also tested for comparison. The inputs and outputs of these one stage models are also neural ensemble firing rates and lever pressures. However, these models had no classification stage and used a single regression model on the entier dataset.

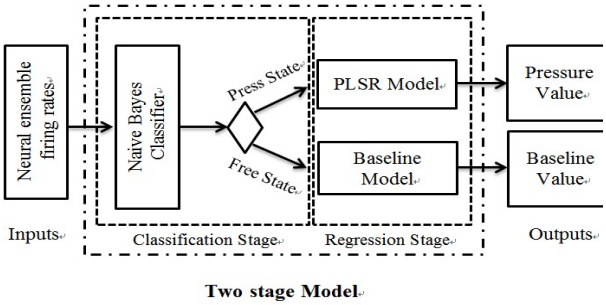

Figure 1. An illustration of Two-stage decoding Model.

3) Performance Measurements:

The classification performances were measured by Accuracy Rate (AR), False Negative Rate (FNR) and False Positive Rate (FPR) which are defined as follows: *AR=(number of correctly classified testing samples)/(total number of testing samples)*, *FNR=(number of misclassified samples in press state testing samples)/(total number of press state testing samples)*, *FPR=(number of misclassified samples in free state testing samples)/(total number of free state testing samples)*.

The decoding performance was measured by Correlation coefficient (CC) between the predicted and measured pressure values. To eliminate the dependence of data set selection, 10-fold cross-validation was performed. To compare the time consumptions of the decoding algorithms, we measured the time consumptions of testing a signal sample.

III. Results

A. Neural Ensemble Firing Pattern Analysis and Classification

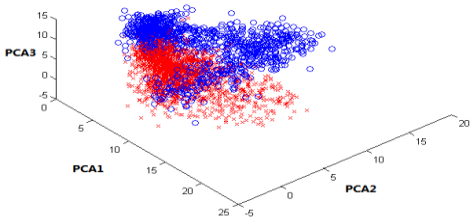

Figure 2. Visualization of neural ensemble firing rates by projecting neural ensemble firing rates to the first three PCA components space. The blue circles represented the neural data when the rat was pressing the lever (Press State). The red crosses represented the neural data when the rat was not pressing the lever (Free State).

Figure 2 shows the distributions of Press State and Free State neural ensemble data. As we can see the press state data and the Free State data distributed differently even in the three dimensional PCA space which can not represent all the information of the data. Therefore, we could infer that the neurons have different firing patterns when the rat was in different motion states, so it makes sense to classify neural

firing data into different states which in our work were Press State and Free State.

Figure 3 shows the classification results of each of the three rats' data. The classification accuracies were greater than 90 percent and rise with the increase of time delay and then reach a plateau for all the rats. It is because the recent motions are related to a short period time of previous neural activities, but when the time delay is too long there is no additional information in the previous neural activities. In our work we choose 500 ms (5 bins) as the length of time lag for both classification and regression.

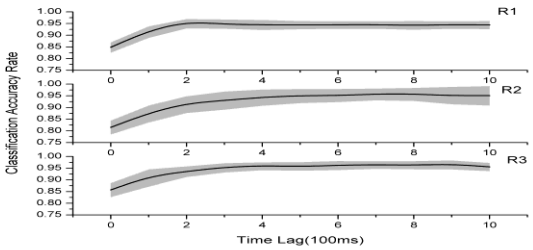

Figure 3. Classification accuracy rates with different time lags. The black lines in each subplot indicate the average accuracy rates and the gray areas indicate the standard division.

Table 1 shows the classification results when taking 500 ms as the time lag. As we can see from the table the overall accuracy rates were very high for all the three rats. The False Positive Rates which represent the degree of misclassifying Free State to Press State were no more than 4 percent. Although the False Negative Rates were higher than FPRs, the classification results of Press State were good enough for further regression, because most of the misclassified Press State samples were on the edges of the pressure peaks as illustrated in Figure 4. Figure 4 shows a data segment with corresponding classification results. The number of red dots which represented the misclassified samples was very small and no one appeared on the pressure peaks.

TABLE I
NEURAL ENSEMBLE PATTERN CLASSIFICATION RESULTS

Criteria	R1	R2	R3
AR	0.944±0.01824	0.94967±0.03089	0.95833±0.01881
FNR	0.1690±0.1240	0.1147±0.0280	0.0718±0.0277
FPR	0.0346±0.0135	0.0378±0.0369	0.0269±0.0236

AR: Accuracy Rate, FNR: False Negative Rate, FPR: False Positive Rate of classification when setting the time lag to 500 ms.

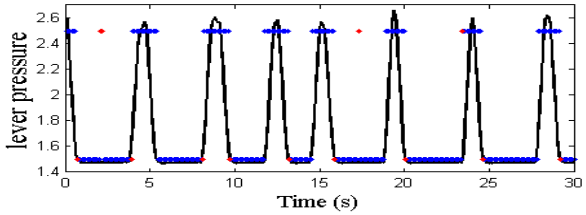

Figure 4. A data segment illustrating the classification result. Black solid line represents the real pressure. Samples with the lever pressure bigger than 1.5 were in the press stage. The blue markers denote the correctly classified samples and the red markers represent the misclassified ones.

B. Comparison of the decoding accuracy of four algorithms

The decoding performances of GRNN and TSM measured by CC values were superior to that of KF and PLSR. The decoding performance of TSM was comparable to that of GRNN on all of the datasets. GRNN is a non-linear decoding algorithm while KF and PLSR are both linear algorithms. The results that GRNN outperforms KF and PLSR suggested that there exit a non-linear relationship between neural activities and the lever pressure. However, the performance of TSM which consists of two linear models was no worse than that of GRNN. This indicated that the neural activities are well-modeled by linear processes when rats were pressing the lever, even though they may still be, most likely, inherently non-linear. We analyzed the decoding performances separately when the rats were in Press State, and the results were shown in Figure 6. It can be seen from Figure 6 there were no longer significant differences between linear and non-linear algorithms, and the performances of KF and PLSR were even better than that of GRNN on Rat 2.

Figure 5. Correlation coefficient between the real pressure and the predicted pressure decoded by four types of algorithms of each rat. Its mean, median, and (25%, 75%) percentiles are displayed as a sphere, a horizontal line, and a box, respectively. Top and bottom triangles give the extremum values.

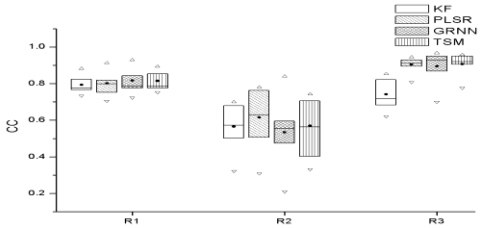

Figure 6. Correlation coefficient between the real pressure and the predicted pressure during the Press State. Its mean, median, and (25%, 75%) percentiles are displayed as a sphere, a horizontal line, and a box, respectively. Top and bottom triangles give the extremum values.

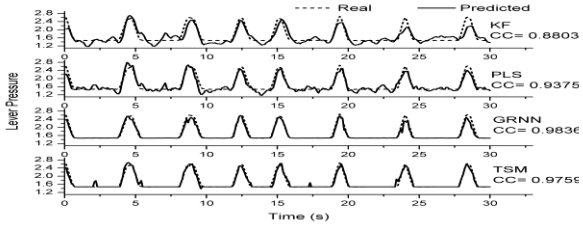

Figure 7. Pressure values were predicted, respectively, using four types of decoding algorithms. Dash line represents the real pressure and solid the predicted one. CC values between the real and the predicted pressures were shown.

Figure 7 shows an example segment data to display the decoding performances of the four algorithms. The CC values of GRNN and TSM were much higher than that of KF and PLSR. From the decoded waveforms, we can see GRNN and

TSM performed much better than KF and PLSR when rats were not pressing the lever. When rats were pressing the lever, nevertheless, the decoded waveforms were not much better than KF even no better than that of PLSR. GRNN's CC value was a little higher than TSM's, but the predicted pressure peak waveforms of TSM were closer to the real ones than that of GRNN. TSM's performance was a little worse than GRNN's during the free period for the reason that some misclassified samples were contained in the classification stage.

C. Comparison of time consumption of four algorithms

Table 2 shows the time consumptions (in seconds) of the four decoding algorithms when testing a signal sample with 2000 training samples. PLSR was the fastest algorithm, because its input dimension depends on the number of latent factors which is much less than the original input data dimension. KF was the slowest one in our work, but its time consumption only depends on the input data dimension and grows linearly with dimension. Although GRNN ran faster than KF, it may became the slowest algorithm if the training samples grows larger, because its time consumption both depends on input data dimension and the length of training samples. The point was that although TSM had two processing stages, its time consumption was an order of magnitude smaller than that of GRNN, because both processing stages of TSM were linear algorithms.

TABLE II
COMPARISON OF TIME CONSUMPTION

Algorithm	R1	R2	R3
KF	0.0316s	0.0213s	0.0641s
PLSR	3.2122e-5s	2.8630e-5s	3.4566e-5s
GRNN	0.0250s	0.0123s	0.0289s
TSM	0.0020s	0.0020s	0.0025s

IV. Discussion and Conclusion

In this studies, we analyzed the neural signals from primary motor cortex which were recorded when rats were using their forelimb to press a lever to get reward. The Naive Bayes Classifier provided a good discrimination of motion state of rats, with accuracy greater than 90%. These classification results confirmed that neural ensemble firing patterns had different patterns when the rats were in different motion states. It also confirmed that the firing patterns of neural ensembles in M1 were more similar during the same forelimb motion state. The results suggested the likelihood of decoding the neural ensemble signals in two motion states with different decoding algorithms respectively. Therefore, we designed a Two Stage Model which classify neural activities to different states firstly and then establish model for each state respectively.

All of the four decoding algorithms showed similar decoding performances when they were used to decode the pressure in Press State. However, TSM outperformed KF and PLSR and was comparable to GRNN when they were used to decode the entire lever pressure. The results in our studies suggest that Two Stage Model had advantage in complicated decoding task. In paradigams with higher dgree of freedom (such as the Center-out task), the advantage of TSM would be more obvious.

In addition, TSM showed similar accuracy with GRNN in both comparisons. The main idea of GRNN is to restore the neural firing patterns of training data and compare these patterns with newly coming testing data. This is equivalent to establishing a set of memory models with the same number of training samples, so that GRNN is very time-consuming. However, using the prior knowledge of the paradigm, TSM only established a small number of models which in our work were Press State model and Free State model.

In conclusion TSM can simultaneously get both high accuracy and low computational cost.

Acknowledgment

The authors are indebted to Dr. Junming Zhu and Chaonan Yu for exceptional technical assistance.

References

[1] J.P. Donoghue, "Bridging the brain to the world: a perspective on neural interface systems," *Neuron*, vol. 60, (no. 3), pp. 511-521, 2008.

[2] M.A. Lebedev and M.A.L. Nicolelis, "Brain-machine interfaces: past, present and future," *Trends in Neurosciences*, vol. 29, (no. 9), pp. 536-546, 2006.

[3] J. Wessberg, C.R. Stambaugh, J.D. Kralik, P.D. Beck, M. Laubach, J.K. Chapin, J. Kim, S.J. Biggs, M.A. Srinivasan, and M.A.L. Nicolelis, "Real-time prediction of hand trajectory by ensembles of cortical neurons in primates," *Nature*, vol. 408, (no. 6810), pp. 361-365, 2000.

[4] M. Velliste, S. Perel, M.C. Spalding, A.S. Whitford, and A.B. Schwartz, "Cortical control of a prosthetic arm for self-feeding," *Nature*, vol. 453, (no. 7198), pp. 1098-1101, 2008.

[5] A.P. Georgopoulos, A.B. Schwartz, and R.E. Kettner, "Neuronal population coding of movement direction," *Science*, vol. 233, (no. 4771), pp. 1416 -1419, 1986.

[6] L. Paninski, M.R. Fellows, N.G. Hatsopoulos, and J.P. Donoghue, "Spatiotemporal tuning of motor cortical neurons for hand position and velocity," Journal of neurophysiology, vol. 91, (no. 1), pp. 515-532, 2004.

[7] W. Wu, A. Shaikhouni, J. Donoghue, and M. Black, "Closed-loop neural control of cursor motion using a Kalman filter," in *IEEE Eng. Med. Biol. Soci*., 2004, pp. 4126-4129.

[8] S. Kim, J. Sanchez, Y. Rao, D. Erdogmus, J. Carmena, M. Lebedev, M. Nicolelis, and J. Principe, "A comparison of optimal MIMO linear and nonlinear models for brain¨Cmachine interfaces," *Journal of Neural Engineering*, vol. 3, pp. 145-161, 2006.

[9] J. Dai, X. Liu, S. Zhang, H. Zhang, Q. Xu, W. Chen, and X. Zheng, "Continuous Neural Decoding Method Based on General Regression Neural Network," *International Journal of Digital Content Technology and its Applications*, vol. 4, (no. 8), pp. 216-221, 2010.

[10] L. Lin, R. Osan, S. Shoham, W. Jin, W. Zuo, and J.Z. Tsien, "Identification of network-level coding units for real-time representation of episodic experiences in the hippocampus," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, (no. 17), pp. 6125-6130, 2005.

[11] C. Kemere, G. Santhanam, B.M. Yu, A. Afshar, S.I. Ryu, T.H. Meng, and K.V. Shenoy, "Detecting neural-state transitions using hidden Markov models for motor cortical prostheses," *Journal of neurophysiology*, vol. 100, (no. 4), pp. 2441-2452, 2008.

[12] B.M. Yu, C. Kemere, G. Santhanam, A. Afshar, S.I. Ryu, T.H. Meng, M. Sahani, and K.V. Shenoy, "Mixture of trajectory models for neural decoding of goal-directed movements," *Journal of neurophysiology*, vol. 97, (no. 5), pp. 3763-3780, 2007.

[13] G.H. John and P. Langley, "Estimating continuous distributions in Bayesian classifiers," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, 1995, pp. 338-345.