Decoding Neural "Stop" State

Werapong (Joe) Goo

Background

Approximately 1.3 million Americans are paralyzed due to some form of spinal cord injury or neurological diseases that cut the connectivity between their brains and bodies. Brain Machine Interface (BMI) is a novel technology aiming to restore lost function to paralyzed patients by translating neural activity from the cortex to control computer cursors, robotic arms or other prosthetic devices. In the past decade BMIs have shown considerable promise in a number of studies including animal experiments[1] and human clinical trials[2]. Preliminary results from these experiments demonstrate that neural spiking activity, both single-unit and multi-unit activities, can be detected by an intracortical electrode microarray, decoded and used for voluntary control of prosthetic devices. Despite early successes, the quality of device control using neural signals is still suboptimal in comparison to that of the native arm's control. Current BMIs are significantly slower, less stable, and fairly inaccurate. The amount of information that the user can obtain per minute is considerably less than that achieved by able-bodied humans. For example, a typical typing speed is about 35 words/min, whereas the disabled patients can generate approximately 5 words/min via BMIs. Hence, a number of improvements are still needed for successful practical applications in the future.

The development of BMIs thus far focuses on effective decoding algorithm, precise cursor positioning and velocity. However, these are not all of the factors that could affect the information transfer rates. Typical applications of computer cursor control assume the ability of the user to click and select the target of interest but this assumption is not true in BMIs user population. The current state-of-the-art BMI can decode continuous (cursor motion) states in real time from a population of motor cortical neurons with certain accuracy but still requires users to hold the cursor still at the target to be considered the selection of a target[2]. The holding time in previous studies ranges from a few hundred milliseconds to a few seconds depending on the quality of the decoder. With a reliable "selection" decoder, we should be able to shave off time and thus gain faster information transfer due to higher number of target selections per time period. In 2007, Kim et al. also saw the importance of the point-and-click feature in BMIs and suggested a method that could simultaneously decode continuous (cursor movement) and discrete (clicking) states in real time. The authors used a linear discriminant analysis (LDA) classifier to decode the "clicking" movement from neural data collected from the patients that were asked to imagine to squeeze their hands[3].

In this work, I have developed a decoder that can detect the neural "stop" state from the neural data that is collected from subjects, i.e. monkeys, who do not need to imagine them doing an unnatural task to them, i.e. squeezing their hands, when making a selection. In addition, I have compared the performance of a few different algorithms in decoding the discrete state of selecting as well as investigate the effect of smoothing/unsmoothing neural data prior to the decoding. I have also investigated the impact of dimensionality reduction, if any, on the performance of the "stop" state decoder.

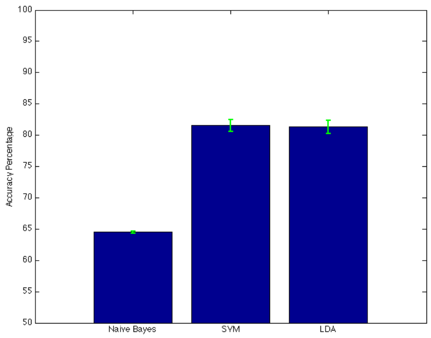
Methods

All procedures and experiments were approved by the Stanford University Institutional Animal Care and Use Committee (IACUC). Experiments were conducted with adult male rhesus macaque (L), implanted with 96 electrode Utah arrays (Blackrock Microsystems Inc, Salt Lake City, UT) using standard neurosurgical techniques. Electrode arrays were implanted in the dorsal aspect of dorsal premotor cortex (PMd) and primary motor cortex (M1) based on the local anatomical landmarks.

The monkeys were trained to make point-to-point reaches in a 2D plane with a virtual cursor controlled by the contralateral arm. The virtual cursor and targets were presented in a 3D environment (MSMS, MDDF, USC, Los Angeles, CA). Hand position data were measured with an infrared reflective bead tracking system (Polaris, Northern Digital, Ontario, Canada). Spike counts were collected by applying a negative threshold, ~ 4.5 x root mean square of the spike band of each neural channel. Neural data were processed by the Cerebus recording system (Blackrock Microsystems Inc., Salt Lake City, UT) and were available to the behavioral control system within 5 +/- 1ms. Visual presentation was provided by using two LCD monitors with refresh rates at 120 Hz. In the brain-controlled mode, cursor kinematics was predicted from spiking activity through a modified Kalman Filter. In the offline analysis, hold/stop state indicator is identified based upon the cursor kinematics data. The indicator is assigned 1 if the velocity is zero and 0 otherwise.

A few data processing procedures, based on the state-of-the-art decoding algorithm, have been done to optimize the performance. First the neural and kinematics data are analyzed in bins of 5-, 20- and 50-ms width, i.e. windowed spike count, based on the suggestion that much longer bin widths can yield higher decode performance offline[4]. To reduce the number of channels used in the training and decoding, mutual information is used to determine the top 10 channels containing highest information regarding the kinematics of the cursor. In some analyses, dimensionality reduction techniques such as PCA or Gaussian-process factor analysis (GPFA)[5] are used instead to reduce the number of dimensions for the model training. Processed neural and kinematics data are then used as training and testing data in the cross-validation process. The classifier is selected from one of the three algorithms: Naïve Bayes, Support Vector Machines (SVM) and Linear Discriminant Analysis (LDA). Model parameters are then fit to the training data (20% of total data). Then, based on data not used for model fitting, the prediction accuracy, precision and recall are evaluated for further comparison.

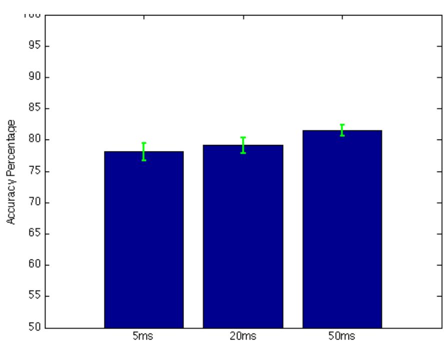
Results



Precision (%)		Recall (%)	
SVM	LDA	SVM	LDA
86.6	81.4	83.0	90.6

Classification Algorithm Comparison

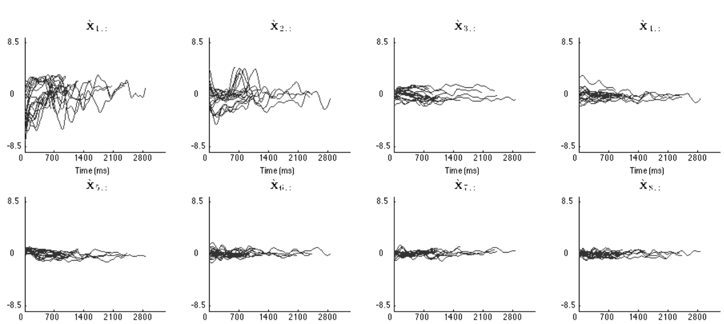
Figure 1 & Table 1: In this experiment, the neural and kinematics data were extracted from the time 150ms after the target onset – the time when the target first appears on screen – to the time when the subject holds onto the target. Twenty percent of trials were randomly selected to be used as training data and the rest was used for testing. Before the data was used for model parameter fitting, the neural and kinematics data were first integrated in 50-ms bin width to increase the amount of information per time step and to reduce any noises in the signal. After preprocessing, we fit the model parameters of Naïve Bayes, SVM and LDA. We found that SVM predicted with comparable accuracy to LDA (81.5±0.9% vs 81.3±1.0%) but both of these algorithms performed significantly better than Naïve Bayes (64.5±0.2%). To further compare SVM's and LDA's predicting performance, we checked their precision and recall values and observed slight differences between the two algorithms. SVM resulted in 86.6% precision and 83.0% recall percentage whereas LDA resulted in 81.4% precision and 90.6% recall percentage.



Precision (%)			Recall (%)		
5ms	20ms	50ms	5ms	20ms	50ms
82.6	81.7	86.6	79.7	83.8	83.0

Effects of Integration Bin Width

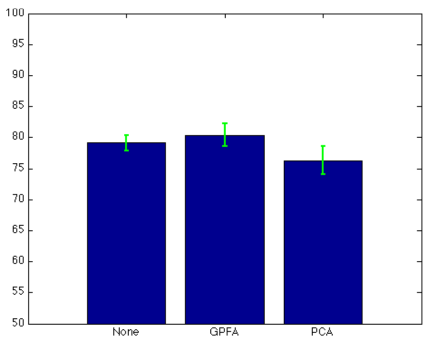
Figure 2 & Table 2: Similar preprocessing steps to the previous session were performed with the data in this experiment with an exception of the integration bin width being varied from 5 to 50 ms to compare the effect of data integration bin width. To investigate the causal effect of integration bin width on accuracy, all other parameters are held constant and the algorithm used, SVM with a linear kernel, was the same across different bin width values. The prediction accuracy percentages were 78.0±1.3%, 79.1±1.2%, and 81.5±0.9% for 5-, 20- and 50-ms bin width, respectively. The increase in integration time appeared to mildly improve the prediction performance. However, the precision and recall percentages are relatively comparable across the three bin width values (see Table 2 above).



Impacts of Dimensionality Reduction

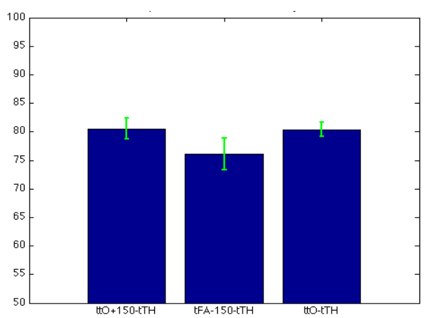
Figure 3: To improve the prediction accuracy, we implemented two different dimensionality reduction techniques, PCA and GPFA[5], to increase the information contained in each feature dimension. We hypothesized that

without dimensionality reduction there might be too much redundant information across multiple channels, especially those channels selected by mutual information method. As an example of dimensionally-reduced data, the top 8 latent dimensions derived from GPFA are plotted in Figure 3. It can be seen that most of the variance are contained in dimensions 1 to 4, whereas the signals in dimensions 5 to 8 are relatively dormant.



Precision (%)			Recall (%)		
None	GPFA	PCA	None	GPFA	PCA
81.7	85.5	80.1	83.8	81.7	81.1

Figure 4 & Table 3: Neither PCA nor GPFA appeared to have much impact on the prediction accuracy. In fact the performance became worse when we applied PCA to reduce the dimensionality of the raw data before fitting the model parameters of SVM. The accuracy percentage without any dimensionality reduction is 79.1±1.3% whereas that when PCA was applied is 76.3±2.3%. As for the decoder with GPFA, we observed that the accuracy percentage in predicting the "stop" state is 80.4±1.8%, which is not significantly different from the baseline value.



Precision (%)		
ttO+150 to End	tFA-150 to End	ttO to End
83.7	85.7	79.3

Recall (%)		
ttO+150 to End	tFA-150 to End	ttO to End
84.4	83.4	84.0

Effects of Time Period Used in Training

Figure 5 & Table 4: In this experiment we evaluated the importance of period of time used in the classification of neural state. The simplest choice for time period used is to include time from the target onset (ttO) to the end of the trial (End). Or we can discard the first 150ms after the target onset (ttO+150) with the assumption that the neural activity during this time period is not highly correlated with either movement or stop state. Another option is to include the data from 150ms before the target is first acquired (i.e. cursor entering the target: tFA-150) to the end of the trial. We found that when we incorporated the information from the target onset or 150-ms after the target onset the decoder could predict the neural "stop" state better than when we only considered 150-ms before the target was first acquired till the end. The accuracy percentages of ttO+150 to end, tFA-150 to end and ttO to end are 80.6±2.1%, 76.0±2.8%, and 80.4±1.2%, respectively.

Discussion

In this study we attempted to create a decoder for discrete neural "stop" state for the application in "point-and-click" task. Below are the findings that we have observed:

Naïve Bayes vs SVM vs LDA: First we investigated the decoding performance of three different classification algorithms, Naïve Bayes, SVM and LDA. SVM and LDA are chosen because we need a binary classifier for the discrimination of neural "movement" and "stop" state based on neural activity recorded from two 96-channel microelectrode arrays. From our experiment, we found that SVM and LDA performed better than Naïve Bayes but between the two the prediction accuracy percentages are comparable. Both SVM and LDA compute hyperplanes for classification that are optimal with respective to their individual objectives; hence, they can perform differently in different applications[6]. However, for our data, the two algorithms yield similar results. They differ only in their precision and recall percentages. SVM has a higher precision but a lower recall rate. In practice we want to have as few false positive incidents as possible because a false positive classification could potentially lead to an incorrect selection of choice. Thus, SVM may be a more promising algorithm for neural "stop" state decoding.

Integration Bin Width: Previous literature has demonstrated that offline and online analyses suggest different parameter choices. Online decoding algorithm that incorporates feedback control performs best with shorter bin widths (25-50ms)[4], whereas offline analysis requires longer bin widths (100-300 ms)[7]. In this experiment we evaluated the prediction accuracy with three integration bin widths; 5, 20 and 50 ms. As expected, we found that 50-ms bin width yields highest decode performance (81.5%). The performance may increase if we integrate the neural activity with a wider bin (>200 ms) but there will then be a tradeoff between performance and temporal resolution.

Dimensionality Reduction: It is neither practical nor efficient to use all 192 recording channels in model training. To improve the efficiency, we first computed mutual information of each channel and select only the top ranked to be used in our decoder. However, we hypothesized that we could improve the performance by using dimensionality reduction techniques such as PCA or GPFA to condense the information into fewer dimensions (see Figure 3). However, our current results do not suggest that that is the case. In fact PCA seems to perform worse than the unprocessed data. It should be noted that we did not separate different types of trials (left or right reach, upward or downward reach) when we performed PCA or GPFA. The signal in latent dimensions (i.e. neural trajectory) can be quite different between these different types of reaches. The decode performance may improve if this process was done prior to the binary classification by SVM or LDA.

Time Period Used: Last but not least, we decoded our neural and kinematics data based on a critical assumption that the cognitive process could be distinguished into two distinctive groups. But this may not be true. It is probable that during the time when the cursor kinematics is zero there could be multiple neural states. To improve the performance, we could use multi-class classifier or we can apply other algorithms that take into account the influence of the presence or absence of a previous state on a current state such as Hidden Markov Model (HMM).

References

1. Taylor, D.M., Tillery, S.I.H. & Schwartz, A.B. Direct cortical control of 3D neuroprosthetic devices. *Science (New York, N.Y.)* **296**, 1829-32 (2002).
2. Kim, S.-phil, Simeral, J.D., Hochberg, L.R., Donoghue, J.P. & Black, M.J. Neural control of computer cursor velocity by decoding motor cortical spiking activity in humans with tetraplegia *. *Review Literature And Arts Of The Americas* **455**, (2008).
3. Kim, S.-phil et al. Multi-state decoding of point-and-click control signals from motor cortical activity in a human with tetraplegia. *Engineering* 1-4 (2007).
4. Cunningham, J.P. et al. A closed-loop human simulator for investigating the role of feedback control in brain-machine interfaces. *October* 1932-1949 (2011).doi:10.1152/jn.00503.2010.
5. Tanji, J. & Evarts, E.V. Anticipatory activity of motor cortex neurons in relation to direction of an intended movement. *Journal of neurophysiology* **39**, 1062-8 (1976).
6. Gokcen, I. & Peng, J. Comparing Linear Discriminant Analysis and. *Analysis* 104-113 (2002).
7. Wu, W., Gao, Y., Bienenstock, E., Donoghue, J.P. & Black, M.J. Bayesian population decoding of motor cortical activity using a Kalman filter. *Neural computation* **18**, 80-118 (2006).