Classification of Behavioral Decision using Pre-decision Neural Activity

Bora Erden
Symbolic Systems Program
Stanford Univeristy
berden@stanford

Blue Sheffer
Symbolic Systems Program
Stanford Univeristy
bsheffer@stanford

1 Introduction

Our project aims to use classification techniques to study decision making processes in macaque monkeys. We are working with the Newsome Lab in the Stanford School of Medicine, whose research agenda is to understand the neuronal processes that mediate visually guided behavior. To this end, they conduct parallel behavioral and physiological experiments in animals that are trained to perform selected perceptual decision tasks (see Experimental Design). By recording the activity of cortical neurons during performance of such tasks, initial insights are gained into the relationship of neuronal activity to the animal behavioral responses. Our combination of behavioral, electrophysiological and computational techniques provides a realistic basis for neurophysiological investigation of cognitive functions such as perception, memory and motor planning. For our project, we use neural data from nuclei implicated in perceptual decision making to predict the decision of an animal during a behavioral experiment.

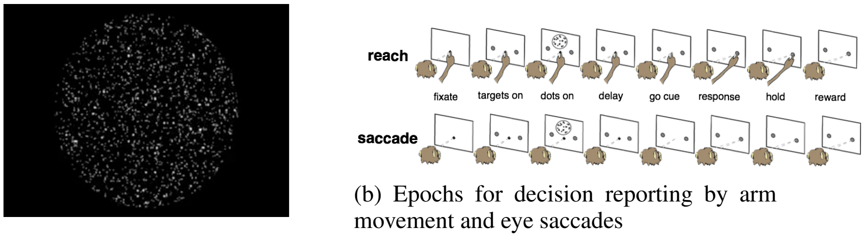
2 Related Work

This project builds upon existing knowledge of the information encoded by the primary motor cortex (M1) and dorsal premotor cortex (PMd). Dr. Newsome, along with other neuroscientists focusing on decision making, have established that areas that are responsible for planning and executing a motor task during a perceptual decision making task show decision signals prior to the decision [1]. What is meant by these signals is that some neurons in those areas will increase their firing rates over time selectively for one decision. Further, recent work in the field has used classification techniques on electrophyshiological data to predict behavior [2][3].

3 Experiment and Dataset

3.1 Experimental Design

We worked with data collected from a series of experiments with macaque monkeys and a random visual stimulus. There are two versions of the task, one where the monkey reports his decision with his arm, and one where he does this with his eyes, but this is of little importance to the classification problem. This stimulus consists of some small dots moving in a specified direction (signal dots) interspersed with others moving randomly (noise dots). By changing the ratio of signal to noise dots we can change the strength of the motion signal they carry. Perceptually, as the ratio of signal to noise dots (coherence) decreases, it becomes harder to tell in which direction the signal dots are moving. The time course of a trial is the following: the monkey starts a trial by putting his hand/eyes on a predetermined fixation point. The trial then starts and the two choice targets where the monkey will eventually move his hand/eyes appear on screen. Then the stimulus comes on for 1 second. Finally, when the monkey sees the go cue, he reports the net motion of the dots by moving



(a) Random dot stimulus

(b) Epochs for decision reporting by arm movement and eye saccades

Figure 1: Visual discrimination task

his hand/eyes to the corresponding target. The monkey's decision is the outcome we are trying to predict.

3.2 Data

While this happens, we record from two 96-channel multielectrode arrays in PMd and M1 from neurons that are known to plan and control arm movement. Using this data, we classify the neural activity to predict which way the monkey will respond. This is reasonable task, using the decision signals mentioned in Related Work.

The data goes through a few stages of processing before it enters the classification algorithms. The electrodes record purely electrical signals, so the data at that point is just time series of voltages for each trial and each electrode.

Then, knowing the typical electrical waveform of a neuron firing or spike, we mark as a spike a certain threshold crossing.

Finally, we spike sort our data. Spike sorting is the process by which the electrical signal from one electrode is clustered into multiple proto-neurons or units. Since the electrode is not directly inside a neuron, but rather somewhere around it, there can be up to 5 neurons around it contributing to its associated voltage. Therefore, to extract the firing rate of a single neuron, we have disentangle the spikes and cluster them into units. To do this, we used ICA and PCA on the features of each spike and used K-means clustering. We used Plexon, a commonly used software for multielectrode spike sorting to do these last two steps (threshold crossing, and spike sorting). One important point that we will come back to later is that this spike sorting is optional. We can use unsorted data, which looks identical to sorted data, except that each electrode is not divided into units.

The final form of our data is thus a matrix of trials x time step x units, where each element is the number of spikes in that time step for that unit during that trial.

4 Methods

To classify our data, we used two algorithms: logistic regression (LR) and support vector machines (SVM).

4.1 Logistic Regression

Logistic regression is a popular and widely used model for classification. Given a feature vector $x^{(i)} \in \mathbb{R}^n$ and corresponding labels $y^{(i)}$, logistic regression finds $\theta \in \mathbb{R}^{n+1}$ (with the extra entry as the intercept term) such that the hypothesis $h_\theta(x) = P(y = 1|x;\theta)$. For logistic regression, the hypothesis is the logistic or sigmoid function: $h_\theta(x) = g(\theta^T x) = \frac{1}{1+\exp(-\theta^T x)}$. Fitting θ is achieved by minimizing the cost function $J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1 - h_\theta(x^{(i)}))\right]$. We used both logistic regression and logistic regression with ℓ_1 regularization, where we add the penalty term $\lambda||\theta||_1$ to $J(\theta)$. Regularization is used to incentivize the minimization to choose small values for the entries of θ, thereby preventing overfitting. We used MATLAB's `glmfit`, `lassoglm` and `glmval` functions for our implementation.

4.2 Support Vector Machines

Support vector machines is a powerful classification algorithm that constructs a separating hyperplane between the two classes. The hyperplane is chosen as to maximize the geometric margin γ with respect to w and b where γ = $\min\limits_{i=1,\dots,m} y^{(i)}(\frac{w}{||w||}^T x^{(i)} + \frac{b}{||w||})$. Intuitively, this makes the decision boundary with the greatest distance between the boundary and the data points. By a lengthy derivation which we omit here, we can show by Lagrange duality that this optimization is equivalent to

$$\begin{aligned} \underset{\alpha}{\text{maximize}} \quad & W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \\ \text{subject to} \quad & \alpha_i \geq 0,\ i = 1, \dots, m. \\ & \sum_{i=0}^{m} \alpha_i y^{(i)} = 0 \end{aligned}$$

The equivalence of the optimization problems is dependent on the Karush-Kuhn-Tucker (KKT) conditions. If we reformulate our SVM optimization problem to include ℓ_1 regularization with penalty weight (or "box-constraint") C for the case of non-separable data, we simply modify our first constraint such that $0 \leq \alpha_i \leq C,\ i = 1, \dots, m.$. For our model, we use ℓ_1 regularization in the early experimental epoch where the neural data cannot possibly be indicative of the monkey's decision (and hence the data is far from linearly separable). We used MATLAB's `svmtrain` to train our SVM, and `svmclassify` to classify test data.

5 Experiments and Results

The classification problem we tackled is highly symmetrical. By that we mean that it is a binary classification task, where both outcomes are nearly equally likely (the percentage of trials where the monkey chose left is 49.3%). This entails that confusion matrices, accuracy vs. precision etc. can be expressed without loss of information with the prediction accuracy. Thus, prediction accuracy will remain our main metric. Accuracy is calculated with 3-fold cross validation.

The more quantitative metric that we will use to distinguish between classifier is the prediction accuracy at the end of the dots stimulus. Since this is the point in the trial where most of the sensory information from the visual stimulus has made it to M1 and PMd where the electrodes are recording from, and thus classifiers can differ the most.

5.1 Logistic Regression

For each session, our two multielectrode arrays each yield a matrix of trials × time step × units, where the value of each element is the firing rate during that trial at that time step for that unit. We first tried to classify the monkey's choice using only one multielectrode (e.g. PMd data) at a time. Using LR, we yield a striking 85% accuracy at the end of the "dots on" epoch (Figure 2).

We then explored concatenating the PMd and M1 data matrices in the units dimension to exploit the most predictive features of both nuclei. But as soon as we ran LR on the combined data set, we realized we had an overfitting problem (Figure 3). Our prediction accuracy at the end of dots using just PMd was at 84.9%, whereas with both it was only at 56.9%.

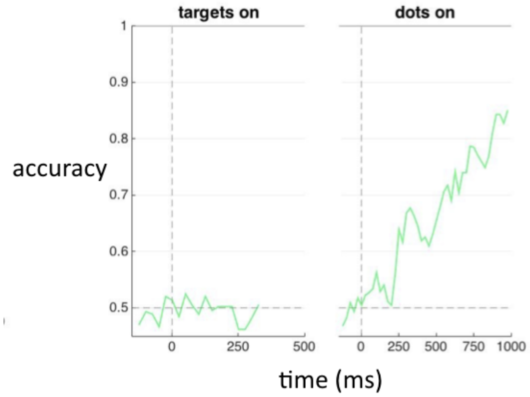


Figure 2: Prediction accuracy using LR with only the PMd multielectrode

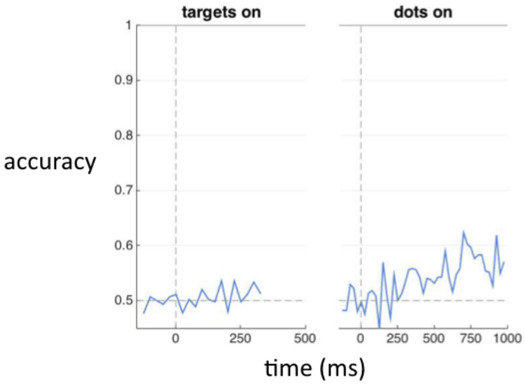


Figure 3: Prediction accuracy using unregularized LR with both PMd and M1

Our next step was therefore to use ℓ_1 regularization. At this point we experimented with different λ values (penalty term) to find a satisfactory one (Figure 4). We settled at a λ value of 0.005, since it had the highest prediction (89.1%) accuracy during dots.

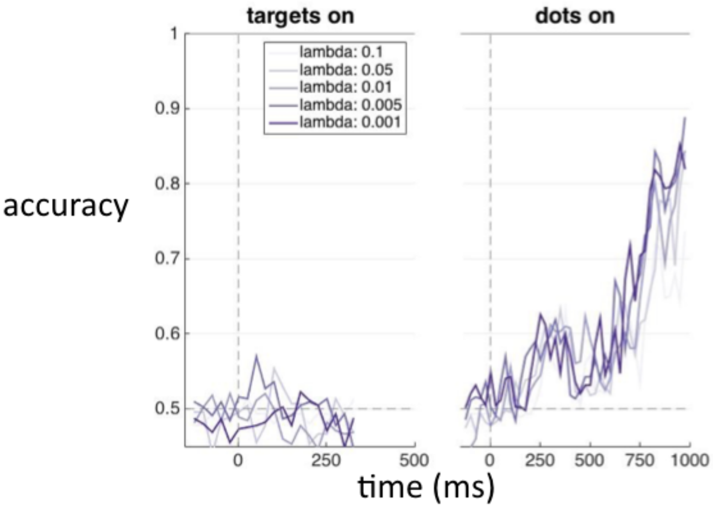


Figure 4: Comparison of accuracy with multiple lambda values

5.2 Support Vector Machines

Our initial pass using LR seemed to be doing well with almost 90% accuracy at the end of dots, but we hypothesized that using SVM, possibly with non-linear kernels, would allow us to extract many more features from our data. We first searched the parameter space for the box constraint (as discussed in 4), and the KKT conditions violation level (the proportion of points allowed to break the inequalities) (Figures 5, 6).

It seemed like these values did not affect the prediction accuracy significantly. We then tried different kernels: linear, quadratic, polynomial, gaussian radial basis, and multlayer perceptron. Surprisingly, the linear kernel performed best. We tested for overfitting but this was not the case (see Discussion and Conclusion).

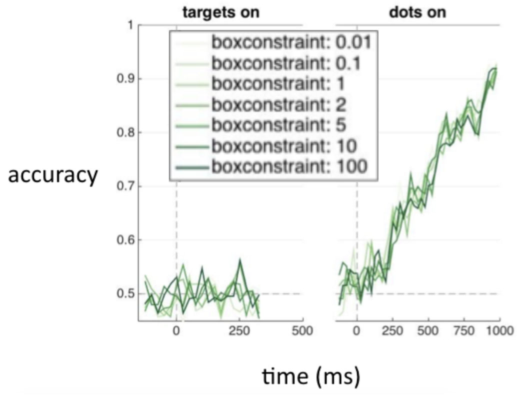


Figure 5: Comparison of accuracy with multiple box constraint values

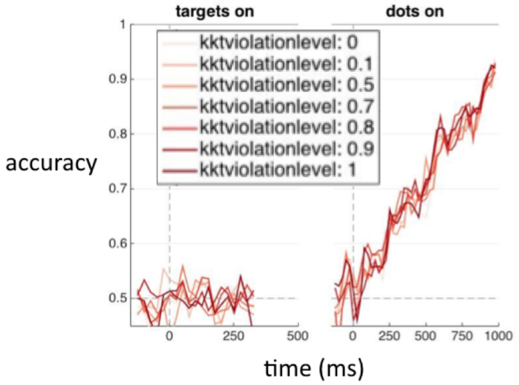


Figure 6: Comparison of accuracy with multiple KKT violation values

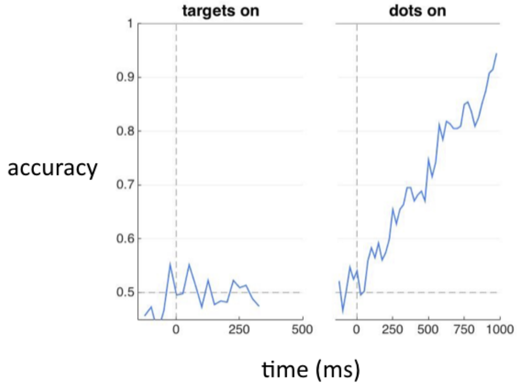


Figure 7: SVM trained on sorted data

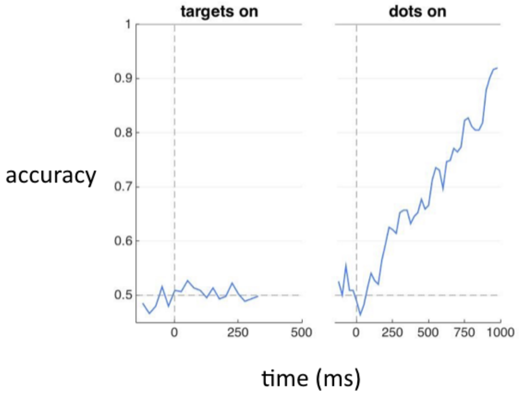


Figure 8: SVM trained on unsorted data

Finally, we come back to the sorted/unsorted difference. Our last figure the difference between using sorted and unsorted data. With a prediction accuracy of 91.9%, unsorted accuracy is just a few points shy of sorted accuracy with 94.4% (Figure 7 and 8). This leads us to believe that using unsorted data is satisfactory for closed loop experiments (see Conclusion).

6 Discussion and Conclusion

Our analyses have shown that SVM performs both better and faster (up to 3x) than LR. Additionally a linear kernel achieves highest accuracy, though this will undoubtedly be our next focus of analysis.

Our efforts leave us in a good position to continue this scientific question. The next step is to train a classifier on one day's data, and then apply it online while the monkey is performing the task, and to use the prediction to modify the experimental conditions (e.g. interrupt the trial when the prediction reaches threshold certainty, and ask the monkey to report his decision at that moment). To do this, we would not be able to sort the data clearly, and the computation would need to be done in a reasonable time scale, both of which we have shown to be possible.

Acknowledgments

We thank our faculty mentor, Dr. William Newsome, our research supervisor, Diogo Peixoto, and all of the members of the Newsome Lab.

References

[1] Shadlen, Michael and William T. Newsome. *Neural basis of a perceptual decision in the parietal cortex (area LIP) of the rhesus monkey.* Journal of Neurophysiology 86.4, (2001): 1916-1936.

[2] Kiani, Roozbeh, Christopher J. Cueva, John B. Reppas, and William T. Newsome. *Dynamics of Neural Population Responses in Prefrontal Cortex Indicate Changes of Mind on Single Trials*. Current Biology 24.13 (2014): 1542-547.

[3] Kim, JN and Michael Shadlen. *Neural correlates of a decision in the dorsolateral prefrontal cortex of the macaque*. Nat Neuroscience 2.2 (1999): 176-85.