Predicting Behavior From Neural Dynamics In C.elegans

Xinwei Yu
Department of Physics
Princeton University
xinweiy@princeton.edu

Abstract

How behaviors are generated from brain is a central question in neuroscience. Human brain contains more than 100 billion nerve cells and it's hard to measure and analyze the neural activity in a system level. Caenorhabditis elegans is a nematode which has a compact nervous system that contains much less neurons. With the recent technique development, we are able to measure the "whole brain" neural activity while the worm is freely moving[1]. Here I explore the neural dynamics in C.elegans and study its correlation with behaviors using machine learning methods. I use supervised methods to study the correlation between neural activity and behavior. And I use Autoregressive Hidden Markov Model to find latent states in neural dynamics. I predict continuous behavior variable from the neural activity. I find the correlation between neural latent states and behavior states.

1 Introduction

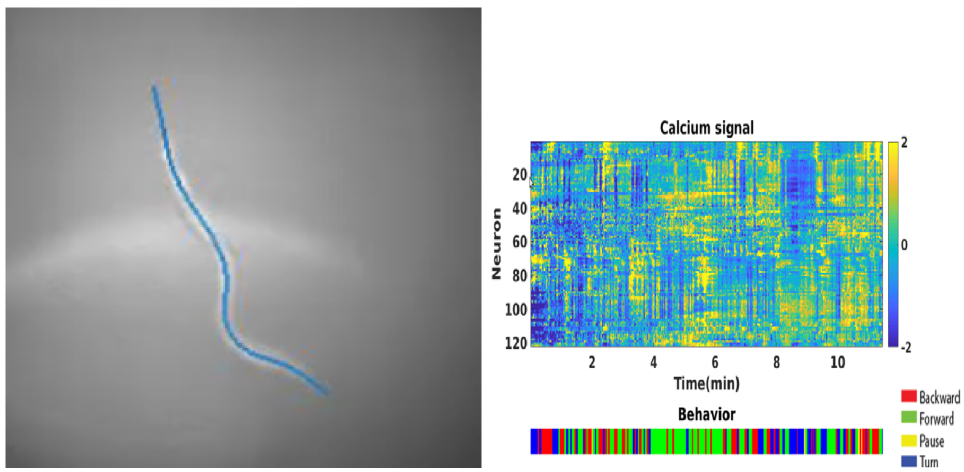
How are behaviors generated from neural activity? Researchers have been working on different animals trying to answer this question. By studying small regions of the brain, people decode the neural coding for head direction[2][3], spatial location[4][5] or arm movements[6]. However, it remains unknown that how the other regions of the brain are involved in controlling the behavior. To fully understand the correlation between neural activity and behavior in system level, a whole-brain scale neural recording is required. Recording the neural signals from all the neurons is unrealistic now in animals like human. Human brains contain more than 100 billion nerve cells, which is too many for recording and analyzing. However, for simple animal like C.elegans, it is possible to measure the majority of its neurons in its compact nervous system.

Caenorhabditis elegans, C.elegans in short, is a transparent nematode. C.elegans has a compact nervous system which contains only 302 neurons (for hermaphrodite) [7][8]. Recently developed technology[1] makes it possible to record neural signals from most of the neurons in the head(up to 150 neurons) of a freely moving worm. It makes it possible to record the behavior and neural activity of multiple neurons simultaneously. Here in this report, I first use supervised methods to find correlation between neural signal and behaviors. I try to predict behavior from the neural signal. Then I explore the dynamics of neural signal using a Autoregressive Hidden Markov Model(AR-HMM). The hidden states found with AR-HMM is compared with the behavior states observed in the data. In this report, I will first describe the methods in Section 2 and then describe the results in Section 4

2 Methodology

2.1 Data Description and Imputation

The data I use comes from our own measurement in Prof.Leifer's lab. The nervous system of C.elegans consists of 302 neurons. We are able to simultaneously measure the behavior and the neural activity of neurons in the head of the animal(around 100 neurons). The behavior(locomotion) of the worm comes from a low mag camera. We use the centerline of worm(See in Fig 1a) to describe its behavior. The neural activity recorded(Fig1b from a high mag camera is the fluorescent signal that reflects the activity of neurons(around 100 in field of view). Both behavior and neural activity are time series(around 3600 frames and 10 minutes in time). The centerline is preprocessed with PCA to extract continuous behavior variable. The first two PC components are correlated to the velocity of the worm and the third PC component is correlated to the "turning" behavior. Here, I focus on predicting velocity from neural activity. The neural signal is preprocessed by smoothing with a Gaussian filter and the NAN value is imputed by interpolation.



(a) Behavior: Centerline of Worm

(b) Neural Activity

Figure 1: Example of Data

3 Methods

3.1 Supervised Learning: Regressor

I make the neural activities of neurons as our features and the velocity of worm as the output. Then finding the correlation between neural activity and velocity becomes a regression problem. One limitation here is that we only have recordings of around 10 minutes long(around 3600 time points). The number of samples(time points) is not big enough and model can overfit easily. Additionally, there are many different circuits for different purposes in nervous system. For a specific behavior, we expect only a subset of neurons that are directly related to it. It is reasonable to use models with regularizer or additional feature selection process.

I use different regression models including Lasso, Elastic Net, Random Forest Regressor and Gradient Boosting Regressor. Lasso and Elastic Net are linear models. They use weighted sum of neural activities to predict velocity. Random Forest Regressor and Gradient Boosting Regressor are both tree models. They use an ensemble of simple decision trees to prevent overfitting. There is no good intuition which regressor should be better before the test because we do not know enough about how the neural computation/decision making happens in the biological nervous system. But a good regressor that can predict the behavior very well may provide some insights into how the biological neural network works.

3.2 Supervised Learning: Classification

Besides velocity, I also predict the behavior states of worms from neural activity. Currently, I characterize four classes of behavior states: forward, backward, turning and pause. Discrete behavior states are more robust to the noise compared to the continuous behavior variable such as velocity. Similarly, I use linear classifier such as SVM and logistric regression. And tree models such as Random Forest Classifier.

3.3 A Simple Way to Add Neural Dynamics

In the models I mentioned above, the behavior is predicted by the neural signal from the same time point. The neural signal that is not perfectly at same time as behavior is neglected. However, it is known that some neurons function as integrator and some functions as differentiator. Those models fail to model the dynamics of neural activity. Also the time structure(information flow) is important in the nervous system. To include dynamics, the feature can be extended to include neural signal in a small time window instead of single time points. This extension include some dynamics of neural signal in our regression model. For example, our feature vector can includes the neural signal 1 second before and after when the behavior happens. In this way, instead of fitting a single weight for each neuron, I fit a kernel for each neuron. And the contribution of each neuron is the convolution of its neural signal and kernel. In this way, the dynamics of the neural activity is taken into account.

3.4 Hidden Markov Model

A better way to describe the dynamics of neural signal is using Hidden Markov Model. The time series data is generated from a Markov chain and the dynamics can be modelled by the transition matrix. As shown in Fig2, there are two Markov chains, one is for the hidden state z, and the other for the observed variables. In our case, the observed variables are the observed neural signal. The hidden state can capture long range dependencies, which makes it a good candidate as the biological internal state. Although we can't make direct measurement of what those internal state should be like, it is believed to be exist and may have correlation with the observed behavior.

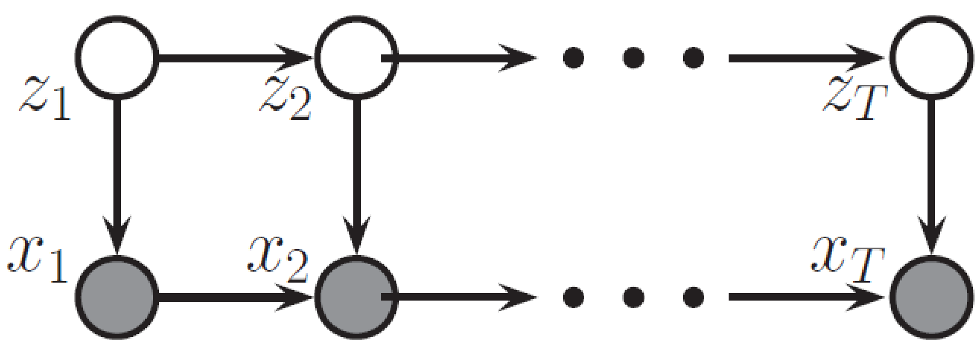


Figure 2: First-order autoregressive HMM

In practice, when fitting the AR-HMM, I first use PCA to reduce the dimension of observed data. We used the first ten PC components and use that as the observed variables x. Those PC components explain 80% of the variance in neural signal.The number of hidden states is a hyperparameter that needs to be set manually. After fitting the model, I compare the hidden state to the observed behavior state. If hidden states and behavior states have some kind of correlation, it provides evidence that AR-HMM is capturing the dynamics in the nervous system and possible way to interpret the hidden states.

4 Results

4.1 Predict Velocity From Neural Signal

We tried different regression model to predict velocity from neural signal. Since we don't have good knowledge about how the velocity is correlated with neural signal, models with good prediction power can provide us with some good insights of how biological neural computation works. We first split data into training set and test set. The first 80% of data is used as training set and the last 20% of the data is used as test set. I don't shuffle the data before split training and test set because, for time series data, sample points are not identical independent distributed. Samples observed later in time can carry the information about what has happened before that. So we choose to use the samples observed earlier as training set and samples observed at last as test set. We used cross validation to search for the optimal hyperparamters. Similarly I used TimeSeriesSplit function to preserve the time structure when split validation set. The result of different models is shown in Table1. The Lasso[9] and Elastic Net have a regularization term to perform the feature selection. For Random Forest and Gradient Boosting, I used SelectKBest function to do the feature selection before feed the data into the model. As it turns out, different models has similar performance and linear models are slightly better.

	Lasso	Elastic Net	Random Forest	Gradient Boosting
R^2(Train)	0.47	0.47	0.51	0.67
R^2(Test)	0.25	0.25	0.23	0.22

Table 1: The R^2 of Predicting Velocity From Neural Signal(Without Dynamics)

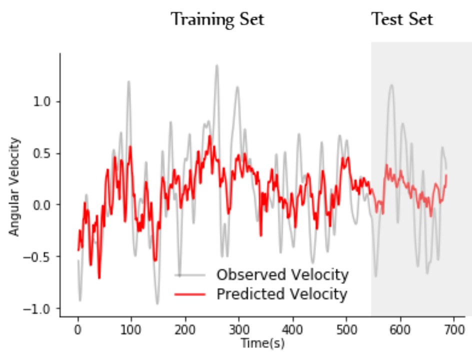
The model we use before does not capture any dynamics in neural signal, to predict behavior y_t, we only use the neural signal x_t observed at the exact same time. All the relation between behavior and neural signal that are not perfectly matched in time are neglected. Here we simply expand the feature to be $\{x_{t-2}, x_{t-1}, x_t, x_{t+1}, x_{t+2}\}$. In this way, we can link the dynamics of neural signal to the observed velocity. Again I use cross validation for the hyperparameters and prevent overfitting. With more dynamics of neural signal, all the models gets a higher R^2 value. It shows that the neural dynamics is useful when predicting the velocity. And Random Forest performs the best.

	Lasso	Elastic Net	Random Forest	Gradient Boosting
R^2(Train)	0.60	0.47	0.42	0.49
R^2(Test)	0.33	0.27	0.34	0.28

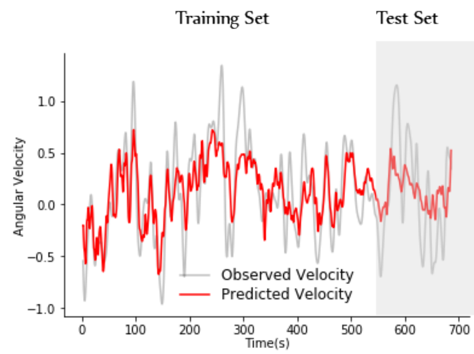
Table 2: The R^2 of Predicting Velocity From Neural Signal(With Dynamics)

We next compare the predicted behavior from different models. I choose to compare the results from Lasso and Random Forest. Lasso represents the linear model and Random Forest represents the tree models. I plot the observed velocity and predicted velocity together in Fig3. Fig3a and Fig3bshows the predicted velocity from Lasso. The velocity predicted by Lasso has the similar trend as the observed velocity. However, it fails to predict sharp speeding up and slowing down cases. Adding the dynamics of neural signal helps to improve the performance of Lasso when the velocity changes drastically. In Fig3c and Fig3d, the velocity is predicted by Random Forest. The R^2 of Random Forest and Lasso are similar. But the predicted velocity from the two models varies a lot. The predicted velocity from Random Forest becomes plateau when the observed velocity is a peak. A possible explanation is that we don't have enough data of those velocity peaks. In the training set, there are peaks with different height. We don't have enough data to predict the amplitude of those quick forward or quick backward motion. So the predicted value for those cases is chosen to be the mean of all the values on the peak, which is a rough approximation.

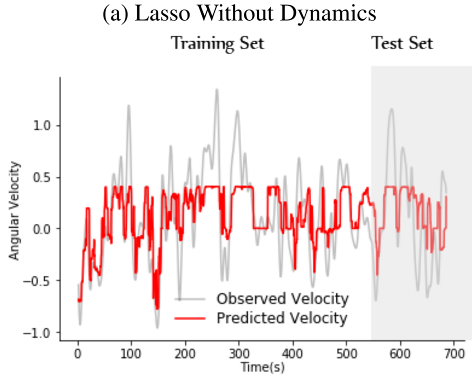
The Lasso predicts the velocity as the weighted sum of some subset of neurons. I think of it as multiple neurons synapse into muscles and control the behavior together. The Random Forest uses decision trees to predict behavior. The corresponding biological process I think of is the neural circuit with multiple layers that perform computation and decide the behaviors. The neurons in different layers are divided into sensory neuron, inter-neuron and motor neurons. The real biological system is much more complicated and highly nonlinear. It seems that our current model is still too easy to predict behavior precisely.



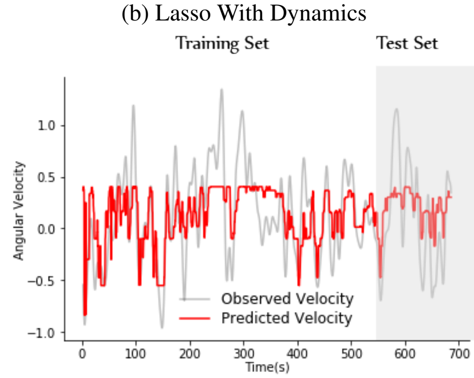
(a) Lasso Without Dynamics



(b) Lasso With Dynamics



(c) Random Forest Without Dynamics



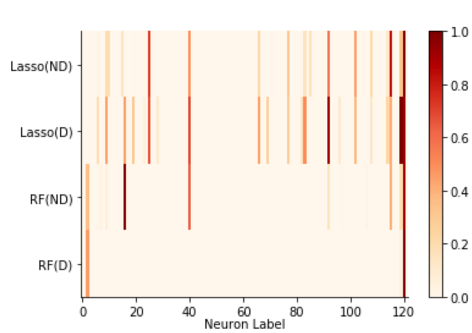
(d) Random Forest With Dynamics

Figure 3: Velocity Predicted From Different Models.

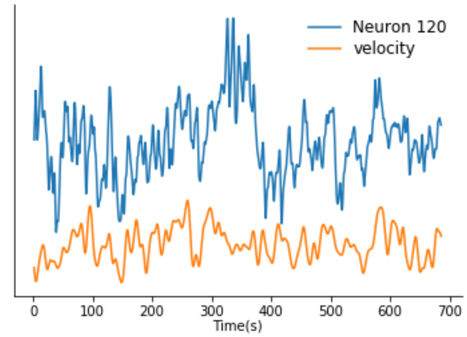
Lasso and Random Forest model choose different features(neurons) to predict velocity. One interesting observation is that Random Forest can predict velocity with a much less number of neurons. For the result above, Random Forest uses only 8 neurons when no neural dynamics is considered and only 2 neurons when dynamics is considered. For Lasso, it uses around 20 neurons to achieve similar predicting power with or without the neural dynamics considered. I plot the heatmap of importance of each neurons in predicting velocity in Fig4a. Neuron 120 is selected by all the models to predict velocity. And Random Forest(with dynamics) mostly rely on this neuron to predict velocity. I plot the neural signal of neuron 120 and velocity in the same figure in Fig4b. There is a good correlation between worm's velocity and neural signal from neuron 120. There are other neurons are selected by multiple models to predict velocity. This makes them a good candidate set of neurons directly control worm's velocity.

4.2 Predict Discrete Behavior State From Neural Signal

In the field, the behavior of worms is more commonly described as discrete behavioral states: forward, backward, pause and omega turn. Here I use supervised classifiers to predict the behavior states from neural signal. The results from different models is shown in Table3. Also I calculated the result after adding the dynamics of neural signal in Table4. Adding the dynamics doesn't help to improve the classification accuracy for discrete behavior state. Unlike the continuous variable, the behavior state is less sensitive to time(state usually last for seconds).



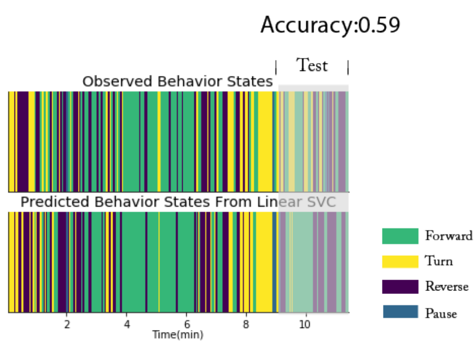
(a) Neuron Importance In Different Models



(b) Most Predictive Neuron

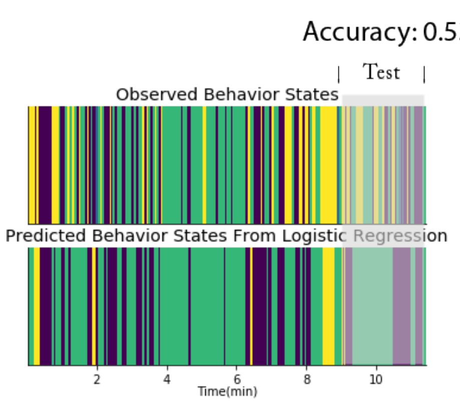
Figure 4: Candidate Neurons For Controlling Velocity

I plot the behavior states predicted by different models in Fig5. Different models gives similar predicting accuracy for the behavior states and support vector machine classifier performs slightly better than others.

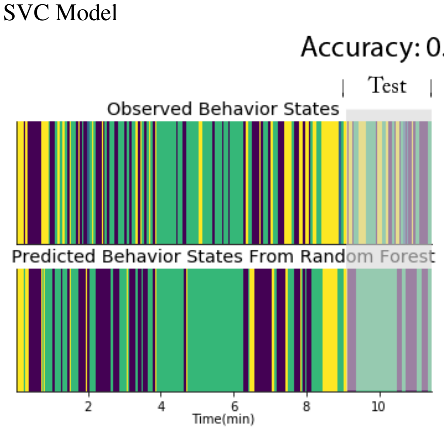


(a) Behavior States Predicted By Linear SVC Model

(b) Behavior States Predicted By RBF Kernel SVC Model



(c) Behavior States Predicted By Logistic Regression

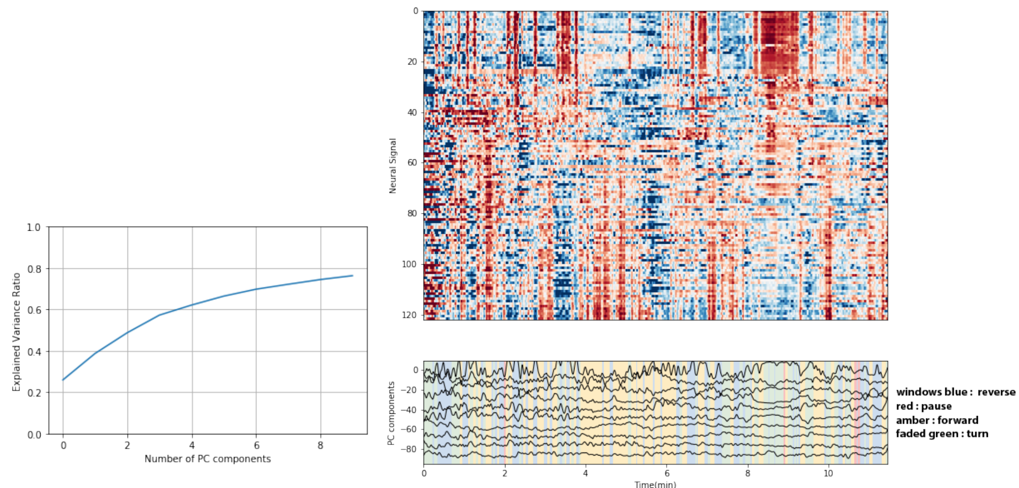


(d) Behavior States Predicted By Random Forest Classifier

Figure 5: Behavior States Predicted By Different Models

4.3 Hidden Markov Model For Neural Dynamics

In Section4.1, I find that the dynamics of neural activity provides more useful information when predicting the velocity of worm. I use Hidden Markov Model(HMM) to learn the dynamics of neural activity from our neural data. In Autoregressive Hidden Markov Model(ARHMM), the observed variable x(Fig2) is the observed neural signal. The hidden state z is interpreted as the internal state. To make it easy to fit the model, I first use PCA to reduce the dimension of observed data. I choose to keep the first 10 PC components as our observed neural data. The first 10 PC components can explain about 80% of the variance in neural activity(Fig6a). The observed neural activity and the first 10 PC components is shown in Fig6b. In this way, we significantly reduce the dimension without losing too much information.



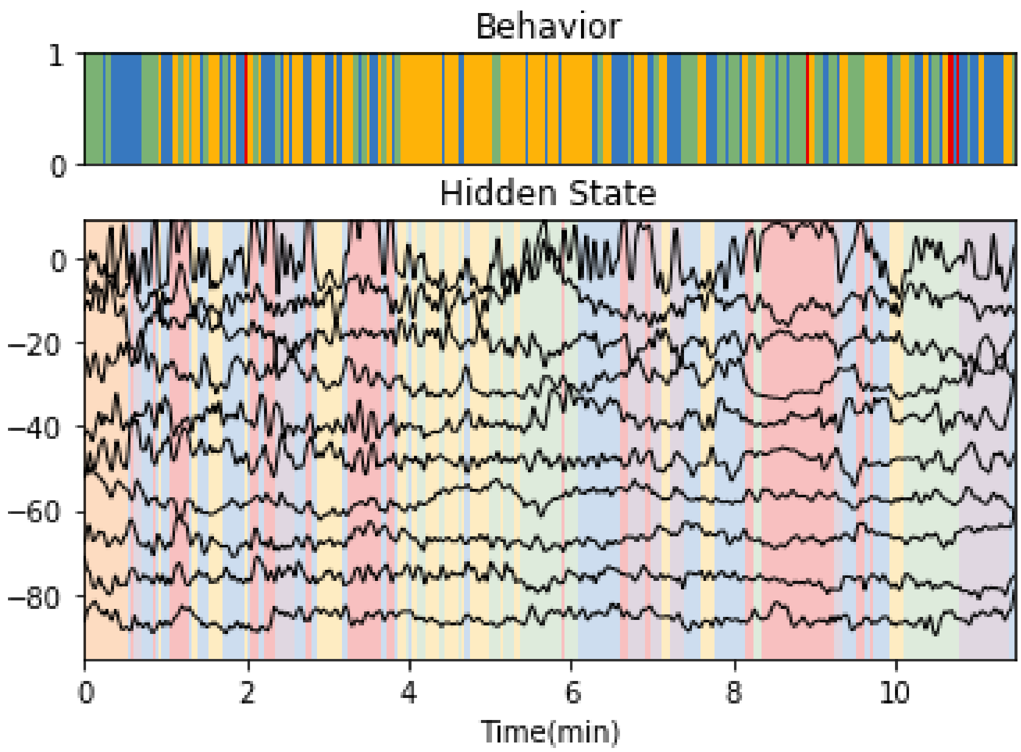
(a) Explained Variance Ratio Of PC components

(b) Neural Signal and PC Components

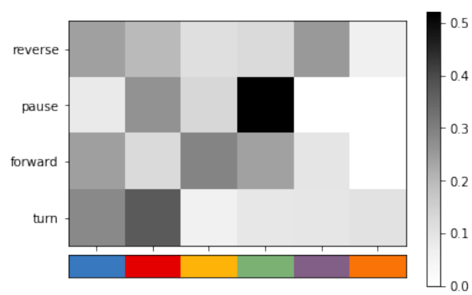
Figure 6: PCA Dimension Reduction

The code for HMM is adopted from https://github.com/slinderman/ssm. I use the Autoregressive Hidden Markov model. The number of hidden states is set manually to be 6. Too many hidden states makes it hard to interpret our results. And we also need to have enough different hidden states to describe the latent dynamics. After fitting the model with our neural data, I compare the hidden states with the observed behavior states. Since behavior data is held out when training the HMM, a good coincidence between hidden state and behavior state is a good evidence that we are finding the right latent state. In Fig7a, I plot the behavior states and hiddens state across time.The hidden states are plotted together with the ten PC components selected as input. In the behavior plot, windows blue:backward, red:pause, amber:forward and faded green:turn. The hidden states learned from neural data does not perfectly match the behavior state. However, we can find hidden state that overlap more with specific behavior state(Fig7b). For example, hidden state 4 overlap mostly with the pause state. Hidden state 3 correlates with forward motion and hidden state 5 correlates with backward motion. Furthermore, I do the statistical test on the overlapping between hidden state and behavior state. I use run-length encoding to encode the time series of hidden state and then shuffle the order of hidden state. The entropy of overlapping between shuffled hidden states and behavior states is tested (Fig7c. The entropy for hidden states without shuffling is located at the red line. The p-value for the test is 0.014, which means that our hidden states is significantly overlapping with the behavior states.

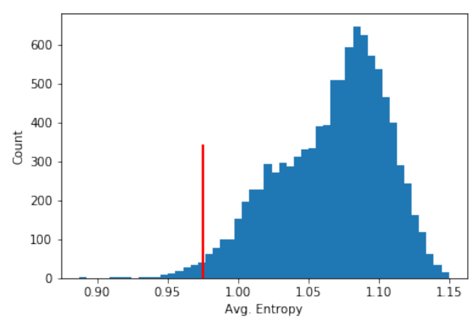
I plot the trajectory of neural dynamics in PC space and color them with the inferred hidden state from HMM in Fig8. Different hidden spaces are separated in the PC space. And our HMM captures the dynamics of transition between the states. We can use those plot to visualize the describe neural dynamics.



(a) Compare Behavior State And Hidden State



(b) Overlapping Between Hidden States And Behavior States



(c) Statistic Test On Overlapping, P value:0.014

Figure 7: Hidden States And Behavior States

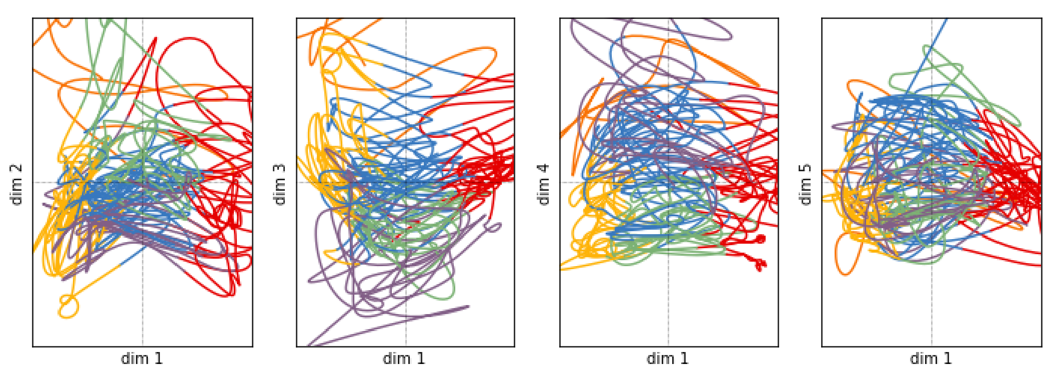


Figure 8: Trajectory of Neural Activity In PC space

5 Conclusion

In this project, I predict behavior from the neural activity. I use different regression models to predict velocity from neural signal. Random Forest has a better prediction power for velocity than other methods. I also predict discrete behavior states with classification models. Among the classifier I use, support vector machine predicts the discrete behavior states best. In order to study the neural dynamics, I use Autoregressive Hidden Markov Model to learn the latent states in the neural activity. I compare the hidden state with the behavior state and find a good correlation between the two. I visualize the neural trajectory in PC space and find that different hidden states are located in different place in the PC space. This project explores the neural dynamics and the correlation between neural activity and behavior. However, we may still need more data to prevent overfitting and further validate the results. We need to be careful whether we actually learn the structure in data or we are just overfitting the models.

References

[1] J. P. Nguyen, F. B. Shipley, A. N. Linder, G. S. Plummer, M. Liu, S. U. Setru, J. W. Shaevitz, and A. M. Leifer., "Whole-brain calcium imaging with cellular resolution in freely behaving caenorhabditis elegans," *Proceedings of the National Academy of Sciences of the United States of America*, no. 10.1073/pnas.1507110112, 2015.

[2] J. S. Taube, R. U. Muller, and J. B. Ranck, "Head-direction cells recorded from the postsubiculum in freely moving rats.," *J. Neurosci.*, no. doi:10.1212/01.wnl.0000299117.48935.2e., 1990.

[3] S. S. Kim, H. Rouault, S. Druckmann, and V. Jayaraman, "Ring attractor dynamics in the drosophila central brain.," *Science*, no. doi:10.1126/science.aal4835, 2017.

[4] J. OKeefe and J. Dostrovsky, "The hippocampus as a spatial map. preliminary evidence from unit activity in the freely-moving rat.," *Brain Res.*, no. doi:10.1016/0006- 8993(71)90358-1, 1971.

[5] T. Hafting, M. Fyhn, S. Molden, M. B. Moser, and E. I. Moser, "Microstructure of a spatial map in the entorhinal cortex.," *Nature*, no. doi:10.1038/nature03721, 2005.

[6] E. V. Evarts, "Relation of pyramidal tract activity to force exerted during voluntary movement.," *J. Neurophysiol*, no. doi:10.1152/jn.1968.31.1.14, 1968.

[7] W. Wood, "The nematode caenorhabditis elegans.," *Cold Spring Harbor Laboratory Press*, no. ISBN 0-87969-433-5., 1988.

[8] E. L. Ardiel and C. H. Rankin, "An elegant mind: learning and memory in caenorhabditis elegans.," *Learn. Mem*, no. 17: 191-201., 2010.

[9] M. Scholz, A. N. Linder, F. Randi, A. K. Sharma, X. Yu, J. W. Shaevitz, and A. M. Leifer., "Predicting natural behavior from whole-brain neural dynamics.," *bioRxiv*, 2018.

6 Appendix

	Linear SVC	Kernel SVC	Logistic Regression	Random Forest
$Accuracy$(Train)	0.74	0.96	0.69	0.98
$Accuracy$(Test)	0.59	0.56	0.53	0.46

Table 3: The Accuracy of Predicting Behavior State From Neural Signal(Without Dynamics)

	Linear SVC	Kernel SVC	Logistic Regression	Random Forest
$Accuracy$(Train)	0.85	0.99	0.63	0.70
$Accuracy$(Test)	0.59	0.56	0.53	0.51

Table 4: The Accuracy of Predicting Behavior State From Neural Signal(With Dynamics)