

## **Search for informative coding in neural activity correlations**

Omer Hazon, Vasily Kruzhiilin, {hazon, kruzh}@stanford.edu

### **Introduction**

Since the development of brain imaging techniques capable of capturing *in vivo* the firing rates of many hundreds of neurons simultaneously under different experimental and behavior conditions, a question of paramount importance has been how much of our present understanding, both physiological and theoretical, of biological neural networks can be directly observed, confirmed or refuted. Interactions between neurons due to their physiological connectivity are thought to resemble a complex coupled dynamical system which gives rise to cognition, and thus could only be truly understood through the analysis of all interconnected neurons simultaneously. A question remains as to whether it is possible to demonstrate that knowledge of the properties of an entire neural ensemble is greater than knowledge of the properties of each constituent neuron independently.

This project is an application of machine learning techniques to neural decoding. We have analyzed a dataset of calcium imaging of neurons from the mouse hippocampus and prefrontal cortex in a freely moving animal performing a decision of making a left or a right turn according to two possible strategies. We investigated whether the ensemble or population coding of the neurons contribute more information than each neuron considered independently. Place decoding and an intra-class shuffle of the inputs revealed this to be the case in the hippocampus. Additionally we were able to demonstrate the ability of end-arm decoding and strategy decoding.

We used as an input the neural activity from each imaged neuron (collectively known as the neural ensemble), with different training examples coming from different trials, or from different timepoints of brain imaging. We used machine learning methods to predict sensory or cognitive variable signals from the neural ensemble, including the position of the mouse at the current time (place decoding), the position of where the mouse will be at the end of the trial before the decision can be observed externally (predictive decoding), and which reward strategy the mouse was following at the time. As reward strategy is a high level cognitive variable and thus is assumed to be difficult to decode. The methods used were Bernoulli naive Bayes, linear SVMs for binary or multiclass classification, and an intra-class feature shuffling method to control for neural correlational coding.

### **Related work**

There are several successful examples of using ML techniques to decode information from neural imaging data. Commonly used is the Bernoulli Naive Bayes (NB) approach, for instance previous work [5] extensively utilized NB to do hippocampal place decoding from mice on a linear track, and used the performance of the decoder to quantify how stable the hippocampal place code is over time. They trained the naive Bayes algorithm, using a spatial bin scheme not different from the one we used in this work, on data from one day of training and evaluate its performance in generalizing to future days, showing a slow decrease in performance, but relative stability up to 45 days past the day from which the training data was collected. This work builds upon the decoding method, showing that naive Bayes does not capture the full place information content in the neural ensemble.

Glaser et al. [2] have called for the use of more modern machine learning tools for neural decoding, comparing traditional methods with SVMs, neural networks, and RNNs. This work, however, does not deal with decoders possessing memory (such as RNNs), and decoding occurs for each example separately. Remedios et al. [4] uses a linear SVM decoder on mouse neural activity to determine the sex of a mouse that a particular mouse is looking at.

The intra-class shuffling method used in this work was inspired by an in-depth analysis of the proper way to shuffle neural data to test various null hypotheses about the neural ensemble [1], showing that low dimensional dynamics in a neural ensemble do disappear, as expected, when data is shuffled across time, neuron index, and condition in a way that preserves mean and covariance of the data tensor along each axis. Our intra-class shuffle is a simpler special case of this method, with the primary question of whether the neural ensemble represents more than the sum of its parts remaining the same. Recently published Meshulam et al. [5] use a different approach at analyzing neural ensembles. They construct generative models based on experimental data and a maximum entropy framework, preserving correlations between neurons, and are able to predict the firing activity of any one neuron from the activity of all other neurons at a given time, demonstrating a dependence between neural features and evidence for information exchange in the network's connections.

## Dataset and Features

The dataset comes from an Schnitzer Group<sup>1</sup> experiment in which mice are trained to perform a simple reinforcement learning task on a 4-armed plus-shaped maze. The four arms are denoted by the four cardinal directions (N,S,E,W). Only one of the end arms is rewarded with a drop of water. The mouse tries to learn the strategy by which it can acquire the reward on each trial (Fig. 1). The experiment imposes two different types of strategies: an allocentric strategy and an egocentric strategy. In the allocentric strategy, only one of the end arms is rewarded all the time. In the egocentric strategy, the mouse must have made a specific type of turn (right or left). Allocentric (allo) strategies reward a final position, and egocentric (ego) strategies reward a turn type relative to the body of the mouse (Fig. 1).

The mouse is first trained to perform a particular strategy, such as an ego left strategy. On the following day the mouse performs the strategy for 100 trials, making a small number of mistakes. On the following day the mouse continues with the ego left strategy, but on trial 50, without cue, the strategy switches to the other strategy type, e.g. allo south, and the mouse must relearn by trial and error to perform the new strategy, until trial 150 when the new strategy is just about acquired. On the following day the new strategy is repeated for 100 trials.

During the three days of observation an ensemble of neurons from the hippocampus were imaged using calcium imaging and processed to extract the calcium imaging signal, which originates from the actual firing patterns for each neuron. The signal was then further processed to identify events in time when that neuron experiences a high firing rate, in the form of a set of disjoint time intervals and their associated amplitudes. In summary, each day of the experiment yields a set of neurons, each of which has a set of events with a starting time, ending time, and an amplitude. The x-y position of the mouse is recorded for all times.

For any particular day of the experiment, for each trial, and for each point in time within the trial, a vector of attributes was constructed which contains as many dimensions as recorded neurons, and a ground truth label representing current spatial position or future position. The i-th element of the attributes vector contains a zero if no events in the i-th neuron occurred at that time, otherwise it contains the amplitude of the event that occurred.

The dataset the models were trained on consists of a 3 dimensional array, with dimensions representing trial number, cell number and frame number (time). There were 100, 150, 100 trials for the first, second and third day, correspondingly, with ~150 frames per each trial and approximately 300-500 cell were observed on the course of each day. Each element of the array contains a 0 if no firing event was detected in the particular neuron, at the particular frame of a trial, and the amplitude value of the event if an event occurred. The Bernoulli naive Bayes further binarizes the data, using a 1 instead of the amplitude value. Each time point (time separation 0.1 s) provided an example.

## Methods

For our project we used Bernoulli Naive Bayes (NB) and Support Vector Machines (SVM) to analyze the properties of neural data.

Bernoulli Naive Bayes it is an ideal decoder in the case of independent features, invariant to intra-class shuffles:

$$\hat{k} = \arg \max_k p(k) \prod_{i=1}^n p_{ki}^{x_i} (1 - p_{ki})^{1-x_i}$$

$$p_{ki} = (1 + \sum_{j=1}^m x_j^{(i)} 1\{k^{(i)} = k\}) / (2 + \sum_{j=1}^m 1\{k^{(i)} = k\})$$

By assuming feature independence, Bernoulli naive Bayes counts for each prediction class label how many times each feature was a 1 in that class divided by the number of examples in that class (with laplace smoothing this means assuming each feature had an additional 1 and 0 in each class). I.e. it finds the probability for the feature being 1 given the class ( $p(x_i = 1|k)$  and  $p(x_i = 0|k) = 1 - p(x_i = 1|k)$ ). Then by independence,  $p(\vec{x}|k) = \prod_{i=1}^n p(x_i|k)$ , for a new example  $\vec{x}$ .

Multiplied with the prior over the classes  $p(k)$ , and taking the argmax over  $k$ , the most probable class is inferred for the new example. Since the algorithm simply counts the number of ones for a feature in any class, shuffling where those ones appear within the classes does not change the result of the counting, and therefore naive Bayes is invariant to the intra-class shuffle.

SVM formulas (linear SVM with L1 regularization) for end arm decoding:

$$L(X, Y) = \frac{1}{2} \|w\|^2 + \frac{1}{m} \sum_{i=1}^m \max(0, 1 - y_i(w^T x_i + b)) + \lambda \|w\|_1$$

The support vector machine (SVM) attempts to find a linear separation between examples labeled as positive and negative. It does so by finding the separation with the maximal margin between the separator and the nearest examples

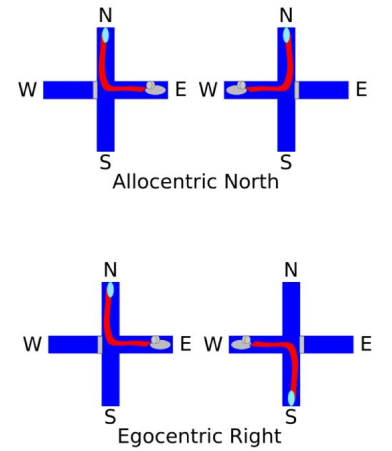


Figure 1. The maze and two possible strategies

<sup>1</sup> pyramidal.stanford.edu

to the separation hyperplane from each class (the support vectors). In this case soft-margin SVMs are used which allow for imperfect separation for some examples for the benefit of having wider margins for the majority of examples. L1 regularization is used as well to improve the generalization of the model as well as to sparsify the parameter vector  $w$ , resulting in automatic feature selection, i.e. finding which neurons have the strongest encodings for the binary classification task. The L1 regularization constant was chosen to be  $\lambda = 0.2$  that showed no sign of significant overfitting, and thus the main purpose of this work was not to create the best possible decoder, but study to the importance of inter-neuron correlations and given a very limited dataset, no further optimization of the parameter was performed

For the multiclass task of place decoding, an ensemble of SVMs (also referred to as error correcting output code SVMs, or ECOC SVMs) was used in a one vs. one scheme whereby each pair of output classes has an associated SVM discriminating between them, resulting in  $K(K-1)/2$  SVMs (190 in this case). The prediction on new data is done by having the SVMs vote for their preferred output class, the winner being the class that is predicted. Due to the sparsity of the input data, the linear SVMs were trained with SpaRSA [6].

One of the goals is to determine to what extent the knowledge about an ensemble of neurons gives more relevant information than the knowledge of the properties of each neuron independently, which can be tested using a machine learning algorithm which does not assume independence of its features, i.e. does not belong to the naive Bayes family. Regardless of the method chosen, the procedure for testing whether ensemble properties matter involves a control whereby each attribute is shuffled across training examples within the same class. While this transformation would not affect the outcome of a naive Bayes method, it can affect the outcome of a method that takes into account the correlations in the feature vector, possibly worsening their performance.

## Experiments/Results/Discussion

### Place decoding

The place decoding was framed as a classification task whereby the track was divided into 20 spatial bins (Fig. 2). The multiclass SVM was implemented using a one-to-one ensemble method, resulting in 190 linear SVMs. It can be consistently seen across three days of neural imaging, that the multiclass SVM performs best, Bernoulli naive Bayes second, and SVM with an intra-class shuffle worst (on test error measuring L1 distance between bins, errors averaged over 16 runs, using holdout cross-validation 70-30), indicating that the neural features are not independent and place information is present in their correlations (Figure 3 and Table 1). Fig. 2 illustrates the most probable errors for this task.

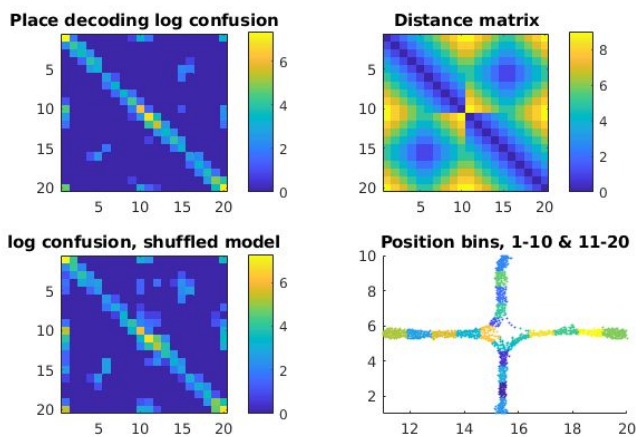


Figure 2: (Top left) Confusion matrix (in  $\log_e$  scale) for place classification for multiclass linear SVM. Most confusions occur with neighboring bins, with some with end arm positions. (Bottom left) Same, but for the shuffled model. More confusions appear beyond just neighbors. Some end arms are confused with non end arm bins. (Top right) Heatmap of the distance between bins, for easier interpretation of confusion matrices. (Bottom right) Visualized spatial binning scheme, with 5 bins per arm, meeting at the center. Numbered 1-10 vertically, and 11-20 horizontally. Each bin is  $\sim 7$ cm.

Switch day place decoding Model:	Train error (distance) 70% 13,727 samples	Test error (distance) 30% 5,883 samples
Bernoulli Naive Bayes	$1.0517 \pm 0.0025$	$1.0788 \pm 0.0068$
Bernoulli Naive Bayes Shuffled	$1.0523 \pm 0.0024$	$1.0722 \pm 0.0075$
Multiclass Linear SVM	$0.3772 \pm 0.0023$	$0.6023 \pm 0.0062$
Multiclass Linear SVM Shuffled	$1.3294 \pm 0.0075$	$1.3700 \pm 0.0107$

Table 1: The comparison of performance of Naive Bayes and SVM position decoding for shuffled and unshuffled hippocampal data

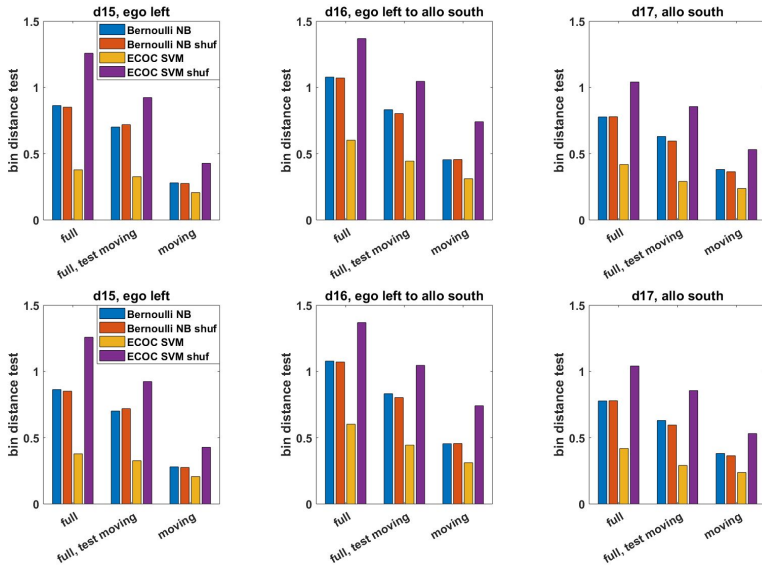


Figure 3: Training (top) and testing (bottom) average distance between true and predicted position bins for 4 decoders, Bernoulli naive Bayes and multiclass linear SVM, both plain and intra-class shuffled. Evaluated on (left groups) entire data set (middle groups) trained on entire set, tested on subset of mouse motion (right groups) trained and tested on subset of mouse motion. Blue and red columns correspond to shuffled and unshuffled Naive Bayes, orange and violet – to shuffled and unshuffled SVM, correspondingly.

### Strategy decoding

The three datasets are comprised of consecutive days, the first and last of which have the mouse following different reward strategies, and on the middle day the reward strategy changes abruptly. The training data for this task is comprised of trials from the first and last days, labeled by the day, and tests on the middle day to determine which trials were from before the switch and which were after. The best strategy decoding performance was found near the turning point on the track (Fig. 4 left), with the AUC value of 0.7.. No difference was observed upon intra-class shuffling. Figure 4 (right) demonstrates the exact positions of errors along the track. We have to note that surprisingly strategy decoding errors at ~0.5 arm position (Fig. 4 left) begin before the strategy switch (trial 24).

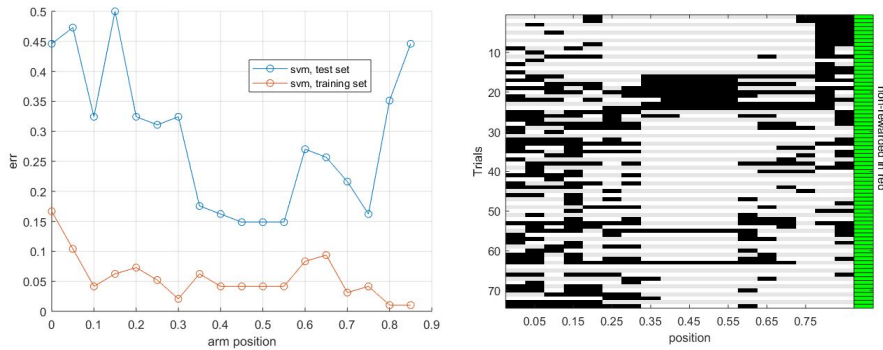


Figure 4: (Left) Strategy-decoding SVM error (fraction of incorrect outputs as a result of a leave-one out cross-validation) as a function of the arm position for the test (blue) and train (orange) data sets. (Right) Map showing at which trials and arm positions errors occurred (black blocks).

### End arm decoding

We also analyzed the possibility to decode where the mouse will end up in each trial before the actual turn takes place from cortex data. Only trials starting from the west arm on the switching day were analyzed (others lack examples from both classes). The Naive Bayes and SVM demonstrated similar performance and no difference was observed upon intra-class shuffling (Fig. 5). The L1 SVM regularization allowed us to identify which neurons encode the trial output at different arm positions, and Fig. 6 shows the firing rate of one neuron that is important for end arm decoding at the very beginning of a trial.

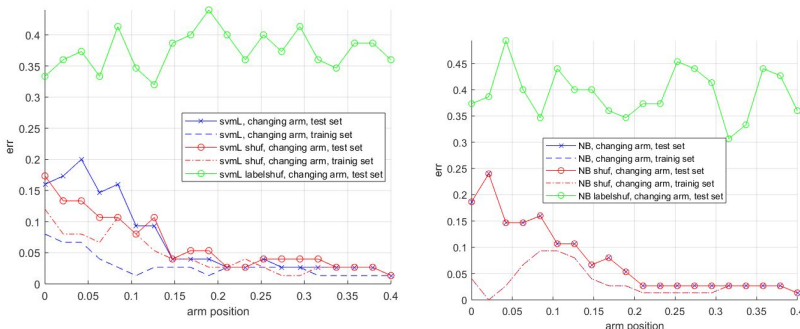


Figure 5: Left: Linear SVM, Right: NB, training error (dashed) and test error (solid) (fraction of incorrect outputs as a result of a leave-one out cross-validation). (Blue) Original data (Red) Intra-class shuffled data (Green) ground truth labels shuffled. Decoding ending position (North/South) for west-starting trials (75 trials) on the switching day.

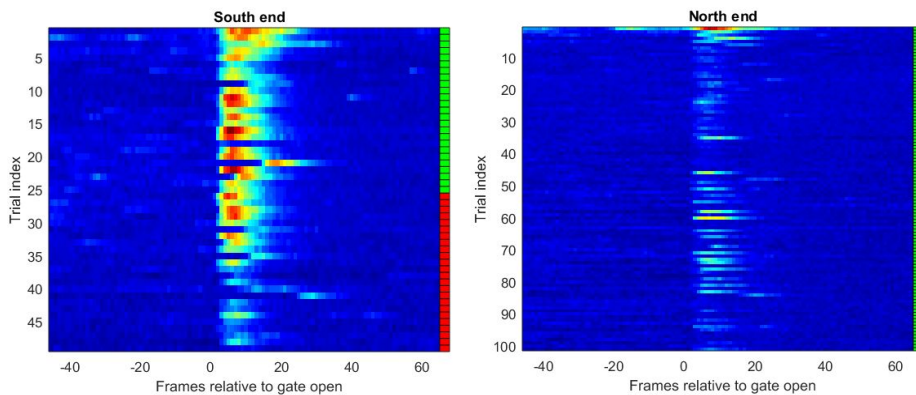


Figure 6: Calcium imaging traces for neuron found by L1 regularization of a linear SVM decoding end arm. Showing activity for trials ending at (left) south and (right) north arms.

### Conclusion/Future Work

The primary goal of the project was to determine whether the neural encoding utilizes dependencies between neurons. For the problem of place decoding, it was found that intra-class shuffling degraded SVM performance, which serves as evidence that the feature independence assumption is incorrect and interneuron dependencies are present in the neural encoding of place in the hippocampus and indicates that for certain problems, such as hippocampal place decoding, commonly used naive Bayes methods do not capture the full information from the neurons. The end-arm predictive decoding and strategy decoding produced no such performance difference from hippocampus or prefrontal cortex data sets.

We were able to demonstrate that one can predict trial output based on hippocampus imaging and identify a small portion of cells that encode for this output in different points along the track. Our data suggests that even the encoding of a complicated strategy concept from the imaging data is possible.

In our project we were mostly limited by the amount of training data collected. When more data will be collected, the natural continuation of this project will involve the application of nonlinear SVM kernels and identification of particular sets of correlated neurons by selective shuffling.

### Contributions

O.H. contributed work on same-time place decoding. V.K. contributed work on predictive end arm decoding and strategy decoding. Both contributed to the data preprocessing, model evaluation code, poster and final report. O.H. and V.K. were advised by Prof. Mark J. Schnitzer (Applied Physics, Biology) regarding this project.

### References/Bibliography

- [1] G. F. Elsayed and J. P. Cunningham, "Structure in neural population recordings: an expected byproduct of simpler phenomena?," *Nat Neurosci*, vol. 20, no. 9, pp. 1310–1318, Sep. 2017.
- [2] J. I. Glaser, R. H. Chowdhury, M. G. Perich, L. E. Miller, and K. P. Kording, "Machine learning for neural decoding," *arXiv:1708.00909 [cs, q-bio, stat]*, Aug. 2017.
- [3] L. Meshulam, J. L. Gauthier, C. D. Brody, D. W. Tank, and W. Bialek, "Collective Behavior of Place and Non-place Neurons in the Hippocampal Network," *Neuron*, vol. 96, no. 5, p. 1178–1191.e4, Dec. 2017.
- [4] R. Remedios, A. Kennedy, M. Zelikowsky, B. F. Grewe, M. J. Schnitzer, and D. J. Anderson, "Social behaviour shapes hypothalamic neural ensemble representations of conspecific sex," *Nature*, vol. 550, no. 7676, p. 388, Oct. 2017.
- [5]

Y. Ziv *et al.*, “Long-term dynamics of CA1 hippocampal place codes,” *Nat Neurosci*, vol. 16, no. 3, pp. 264–266, Mar. 2013.

[6]

S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, “Sparse Reconstruction by Separable Approximation,” *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, Jul. 2009.

[7]

MATLAB and Statistics Toolbox Release 2017b, The MathWorks, Inc., Natick, Massachusetts, United States.