Spring | 14

Rotation Report

- Isaiah S. Morales

The University of Texas at Austin

Table of Contents

Introduction

Neurons are arguably one of the most interesting cells in the body due to their ability to propagate signals over large distances rapidly. They perform this function by generating and sending action potentials down their axon. These action potentials are then sent to other neurons in the body's nervous system.

Interestingly, even though an action potential is thought of as an all or none event, information about something can be hidden within the signal itself. It has been known for sometime that neurons which innervate muscles will flex them based on the average number of spike per unit time, a so called 'rate code'. Conversely, neurons that change the precise timing of their spikes are said to be based on a 'temporal code'. These are by no means the only mechanisms by which neurons modulate there responses, but exploring and understanding these modulations, as well as answering how information is encoded at the population level for a given stimulus, underlies computational and theoretical neuroscience.

Of note, Graf[1] investigated how visual stimuli could be inferred from the responses present in V1 neuronal populations. He found that a parametric decoder that assumes neuronal independence, and an empirical decoder, which uses an assumption free, pooling rule, can both learn the structure of underlying neuronal response distributions. However, the empirical decoder performed at a higher level when compared to the parametric decoder, suggesting that a data driven decoder can better extract the structure of neuronal response distributions, which are in essence carrying sensory information.

Herein, we will try to reproduce these findings, as well as try to extend them by testing whether a support vector machine (SVM) decoder based on a one vs. all classification scheme can generate better results then previously seen. This scheme requires that N binary classifiers be trained to distinguish between examples from a single class vs. the examples of all remaining classes. When classification is necessary, all classifiers are run, and the classifier, which output's the most positive value is chosen as the datum's class.

Methods

A summary of Graf's methods is given below in the next few sections[1].

Visual Stimulation and Recording:

Sinusoidal gratings were presented through a large circular aperture surrounded by grey luminance. The orientation of the grating was rotated in steps of 5 degrees in a clockwise fashion, yielding 72 different orientations (0-355 degrees). 50 trials at each orientation were recorded from V1 with an electrode array. Stimuli where presented for 1280ms, followed by a blank screen of the same duration. Three hemispheres from two pig-tailed (*M. nemestrina*) and two hemispheres of one cynomolgus (*M. fascicularis*) where recorded, and comprised the five different datasets analyzed for this paper.

Spike Count Procedure:

From Graf's data, only spikes that appear from 100ms to 1280ms from stimulus onset will be counted as spikes.

Neuron Selection

In Graf's original paper[1] stimulus orientation decoding accuracy for the Empirical linear decoder (ELD), and Poisson independent decoder (PID) was done across 5 datasets. From these datasets N neurons where chosen for decoding, as seen in *table 1*. Here, the neurons with the highest mean firing rate across the different datasets will be chosen accordingly.

Poisson independent decoder:

The PID assumes the distribution of spike counts are from a Poisson distribution and pools the likelihood function across neurons assuming statistical independence. This process allows the log likelihood function for a given stimulus orientation θ to linearly combine the population response r_i of N neurons.

$$\log L(\theta) = \log\left(\prod_{i=1}^{N} p(r_i \mid \theta)\right) = \sum_{i=1}^{N} \log\left(\frac{f_i(\theta)^{r_i}}{r_i!}\exp(-f_i(\theta))\right)$$

$$= \sum_{i=1}^{N} \log\left(f_i(\theta)\right)r_i - \sum_{i=1}^{N} f_i(\theta) - \sum_{i=1}^{N} \log\left(r_i!\right) = \sum_{i=1}^{N} W_i(\theta)r_i + B(\theta)$$

Here the pooling weight (W) is the logarithm of the individual tuning curves fi(θ). Due to the instability of the log-likelihood function close to 0, a floor (ε) was put on the fi(θ), which was the inverse of the number of trials. The offset term B represents the sum across tuning curves.

Support vector machine:

An SVM finds a hyperplane that maximizes the normalized margin between 2 classes, while also minimizing classification error, as well as responses in the stripe. Here N neurons where recorded on p trails, and response vector is denoted by $\boldsymbol{r_p} \in R^N$. The algorithm discovers the normal vector $\boldsymbol{w} \in R^N$ and offset $\boldsymbol{b} \in R$, as seen below:

$$\min_{\boldsymbol{w}, b, \xi}\left(\|\boldsymbol{w}\|^2 + C \sum_{p=1}^{P} \xi_p \right)$$

for each P=1…p ->

$$t_p\left(\boldsymbol{w} \cdot \boldsymbol{r}_p + b\right) \geq 1 - \xi_p \text{ and } \xi_p \geq 0$$

Where epsilon is a slack variable allowing for overlap and miss matched classes. C is set by cross-validation.

The decision function can be expressed as

$$sign(w * r - b)$$

Empirical linear decoder:

Taking the sign of the SVM decision function will discriminate between two orientations in our case.

$$y(\theta_1,\theta_2) = \sum_{i=1}^{N} w_i(\theta_1,\theta_2)\, r_i \; + b(\theta_1,\theta_2) \equiv \log LR(\theta_1,\theta_2)$$

For the ELD, the SVM decision function is equated to the log likelihood ratio between adjacent orientations, and the likelihood function is given by the cumulative sum of these ratios.

$$\log L(\theta_i) = \sum_{k=2}^{i} \log LR(\theta_k,\theta_{k-1}) \; = \sum_{j=1}^{N} W_j(\theta_i) r_j + B(\theta_i)$$

One vs. All SVM

In the one vs. all classification scheme, one SVM is constructed per class, which is trained to distinguish one class from all remaining classes. At test time, an input's class is chosen according to the maximum output among all SVMs.

$$F(x) = \underset{n=1\ldots N}{Argmax} f_n(x)$$

Here $f_n(x)$ is the unsigned SVM decision function for class n, among all classes (N).

Training and Validation

All decoding models will be trained by randomly selecting 90% of the responses available at each orientation, and validated on the remaining 10%. This process will be repeated a total of 10 times for each model on every dataset. Thereafter, statistics will be performed on the aggregated errors for each model across datasets. Errors are calculated as follows:

$$errors = decoded_{ori} - true_{ori}$$

Errors that are greater than 220 degrees will have 180 degrees subtracted from them; conversely, errors less than -220 degrees will have 180 degrees added to them.

Results

Average and STD of the Error Across Datasets and Methodologies

As can be seen in *figure 1,* the model that made the lowest absolute error on average was the One vs All SVM, only on Dataset 5 did the PID have a lower absolute error then any other model. The ELD was by far the worse at decoding, indicative by always having an average error above 60 degrees. See figures 2, 3, and 4 to see the distribution of errors across decoding models.

ANOVA & Multiple Comparison

A one-way ANOVA was performed on the decoding errors across the different datasets by the three different decoding methodologies. Decoding performance differed significantly across the three models, F (2, 64797) = 15.7, p = 1.52e-07. Tukey post-hoc comparisons of the three models indicate that the PID (M =-1.39, 95% CI [-2.18, -0.60]), the ELD (M =-3.36, 95% CI [-4.99, -1.73]), and the One vs. All SVM (M =1.90 95% CI [0.46, 1.90]) were all significantly different from one another, p < .05. See *table 2 and 3* for more information.

Discussion

Here, we examined if visual stimuli could be inferred from V1 population responses. Similar to Graf we found that a parametric decoder (PID) and an empirical decoder (ELD), which uses an assumption free pooling rule, can both learn the structure of the underlying neuronal response distributions. In contrast to Graf, however, the ELD wasn't superior to the PID. This is most likely explained by one of two reasons: (1) an error in the implementation ELD, (2) neuronal selection. In the ELD, the SVM decision function was used as an approximation of the difference between the log-likelihood of adjacent stimulus orientations. Thereafter, the log-likelihood function was computed by taking the cumulative sum of the log-likelihood ratio between adjacent neurons. It could be argued this methodology has no theoretical guarantee, and could be heavily influenced by neuronal selection.

Lastly, we examined whether an SVM decoder based on a one vs. all classification scheme can outperform the PID and ELD. Indeed, the one vs. all SVM outperformed the ELD and PID. While the performance gain between the one vs. all and the PID was small, it turned out to be significant. This performance gain could be explained by the fact, the SVM doesn't assume an underlying response distribution or even independence, unlike the PID. As a practical matter, however, the one vs. all SVM was many orders of magnitude slower decoding then the PID.

Bibliography

[1] Graf, A. B., Kohn, A., Jazayeri, M. & Movshon, J. A. Decoding the activity of neuronal populations in macaque primary visual cortex. *Nature Neurosci.* 14, 239–245 (2011)

Dataset	N
1	40
2	57
3	60
4	70
5	74

Table 1: In Graf's paper N neurons where chosen from each dataset.

ANOVA Table

Source	SS	df	MS	F	Prob>F
Columns	223492.8	2	111746.4	15.7	1.51737e-07
Error	461054836.5	64797	7115.4		
Total	461278329.3	64799			

Table 2: One-way ANOVA performed on the decoding errors across the different datasets by the three different decoding methodologies

Tukey post-hoc comparison				
Group		Mean Diff	CI	
1	2	1.9656	0.0634	3.8681
1	3	-2.5699	-4.4723	-0.6676
2	3	-4.5356	-6.4380	-2.6333

Table 3: Tukey post-hoc comparison of the three different decoding methodologies. Here group # 1 is PID, #2 is ELD, and #3 is One vs. All SVM.

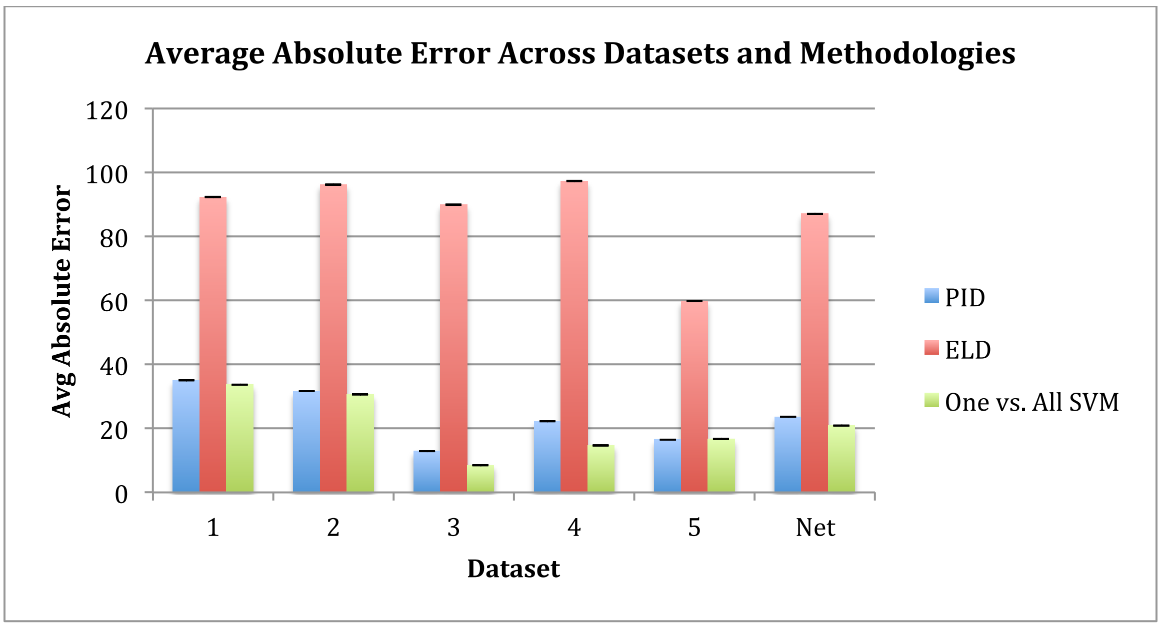

Figure 1: Average absolute error across subjects and methodologies. Here 'Net' refers to avg absolute error across all datasets. Each model was cross-validated across datasets by randomly splitting the dataset into a training (90%) and validation (10%) set. Cross-validation occurred 10 times across each dataset and each model.

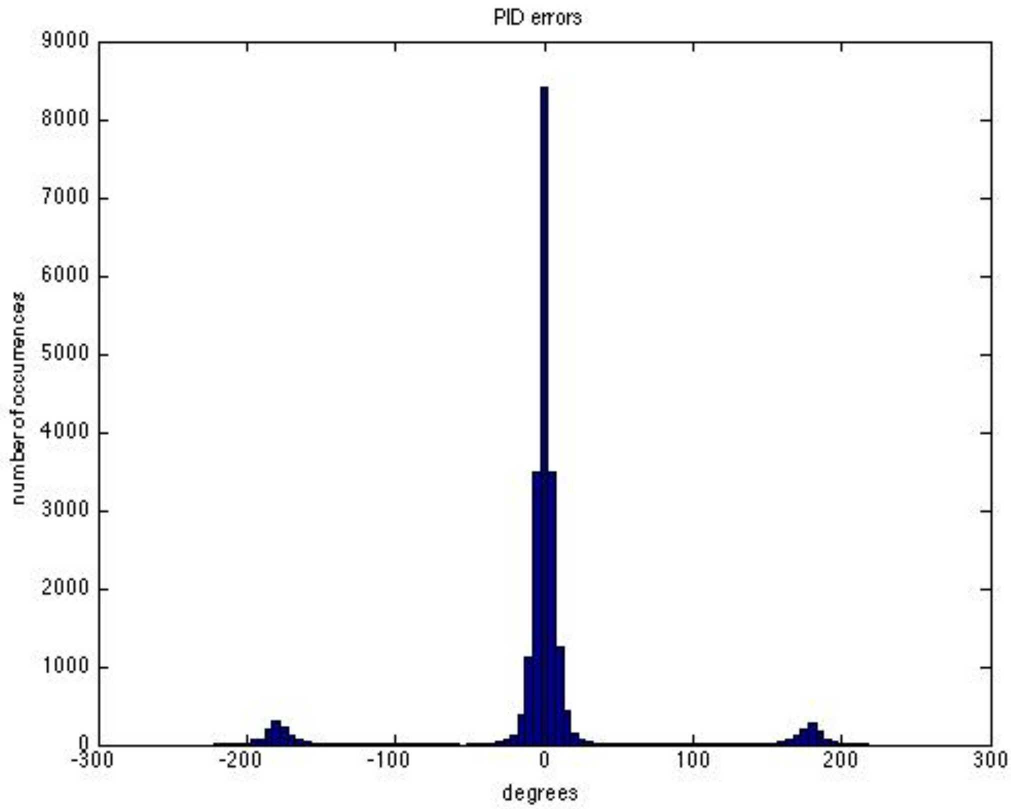

Figure 2: Decoding errors produced by the PID across the 5 datasets. The model was tested on each dataset by randomly training the model on 90% of the data, and tested on the remaining 10%. This process was repeated on each dataset 10 times, and the errors across the 5 different datasets are seen above.

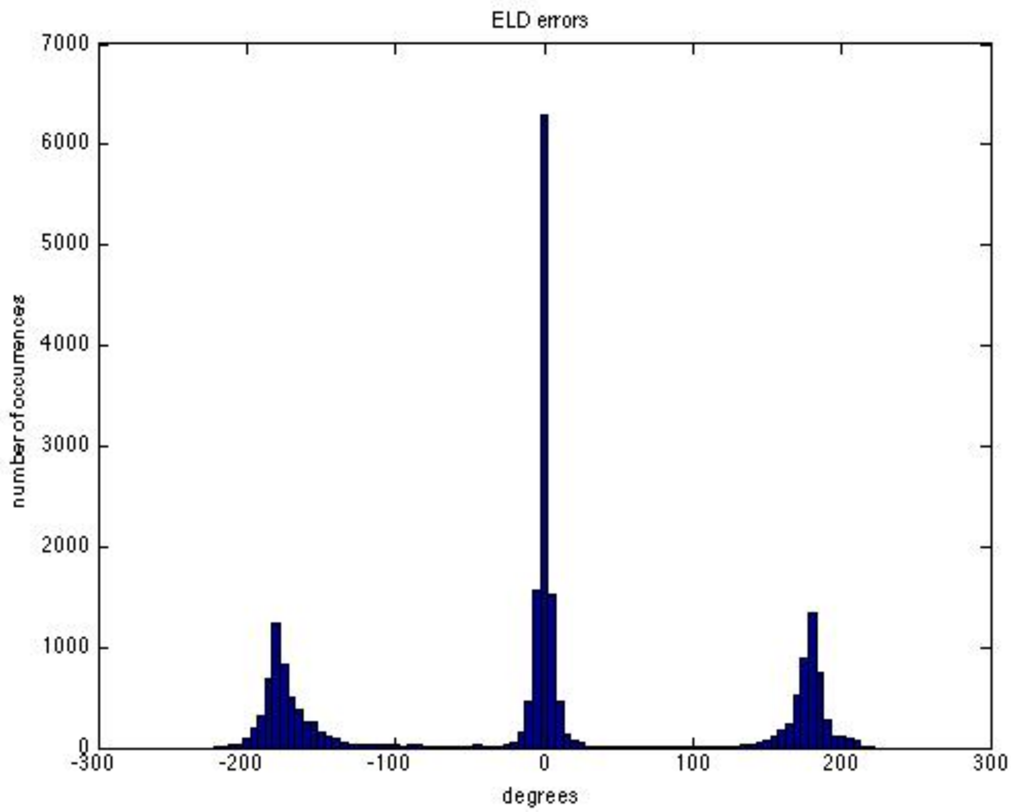

Figure 3: Decoding errors produced by the ELD across the 5 datasets. The model was tested on each dataset by randomly training the model on 90% of the data, and tested on the remaining 10%. This process was repeated on each dataset 10 times, and the errors across the 5 different datasets are seen above.

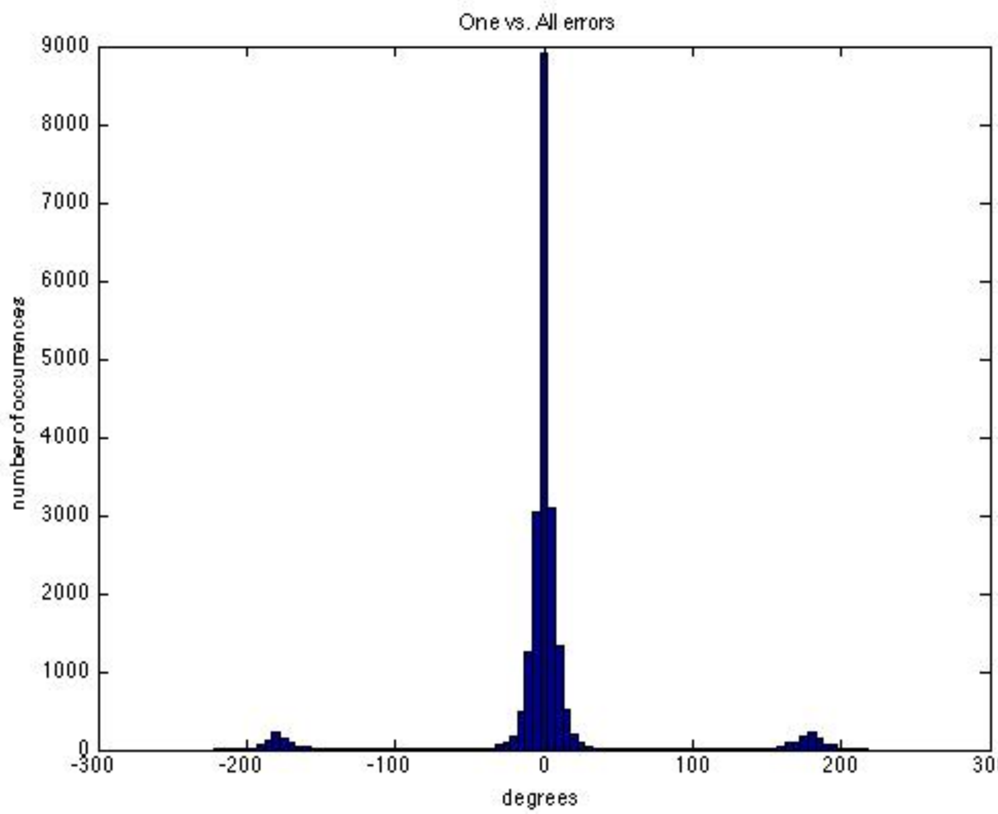

Figure 4: Decoding errors produced by the one vs. all model across the 5 datasets. The model was tested on each dataset by randomly training the model on 90% of the data, and tested on the remaining 10%. This process was repeated on each dataset 10 times, and the errors across the 5 different datasets are seen above.