A3D3 Postbac Update

Lauren Peterson

Overview

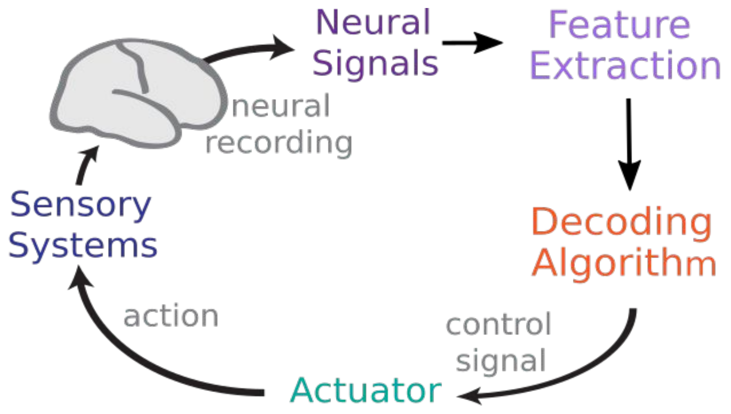
- Convex optimization with neural data
- Time series reconstruction using multiblock RNN autoencoders (ECOG-MRAE)

Convex Optimization

Adaptive Feature Selection

- Can we adapt to a BCI's user learning strategy in trying out different features?
    - Brain engages in **sparsification** of learning
    - Could we free up features for other tasks?

- Goal: develop an online adaptive feature selection scheme for neural interface learning
    - Relevance
    - Sparsity
    - Smoothness



Feature Selection Complications

Non-neural data:

- Assumes data is stationary
- Lots of data
- Trained once

Neural data:

- In general, not stationary
- Less data; requires smoothness
- Decoders need to be calibrated and refitted

Si Jia's innovations

- Incorporate multiobjective perspective
- Add smoothness objective

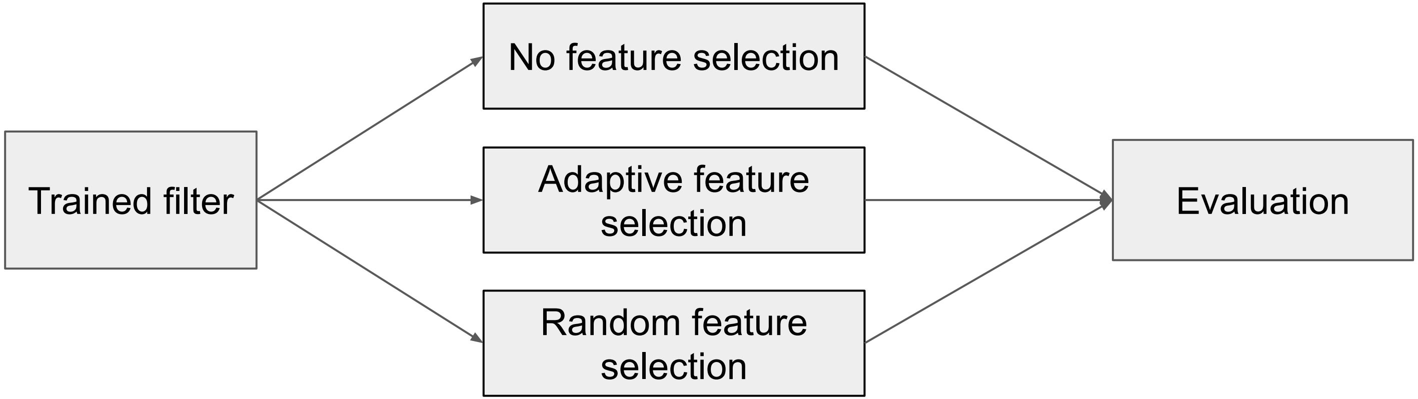
$$\min_{\theta_i} \quad \textbf{-logdet}(C_i^T Q^{-1} \Theta_i C_i) + \lambda \theta_i^T \mathbf{1} - \mu \theta_i^T \begin{bmatrix} \theta_{i-1} & \cdots & \theta_{i-k} \end{bmatrix} \begin{bmatrix} \rho \\ \vdots \\ \rho^k \end{bmatrix}$$

$$\text{s.t.} \quad 0 \leq \theta_i \leq 1$$

My goal: apply Si Jia's convex feature selection algorithm to data from a monkey

General Project Outline

- Train a kalman filter on recorded data
- Run convex optimization algorithm
- Remove “less useful” features
- Evaluate new kalman filter performance



Aside: Kalman Filter

- State-based prediction method: models data based on observed points and assumed hidden states

State-transition:

$$x_t = \boldsymbol{A} x_{t-1} + w_t$$

State transformation (linear)

State covariance (i.e. 'uncertainty')

$$w_t \sim N(0, \mathbf{W})$$

Observation-model:

$$y_t = \boldsymbol{H} x_t + q_t$$

Observation transformation (linear)

Observation covariance (i.e. 'uncertainty')

$$q_t \sim N(0, \mathbf{Q})$$

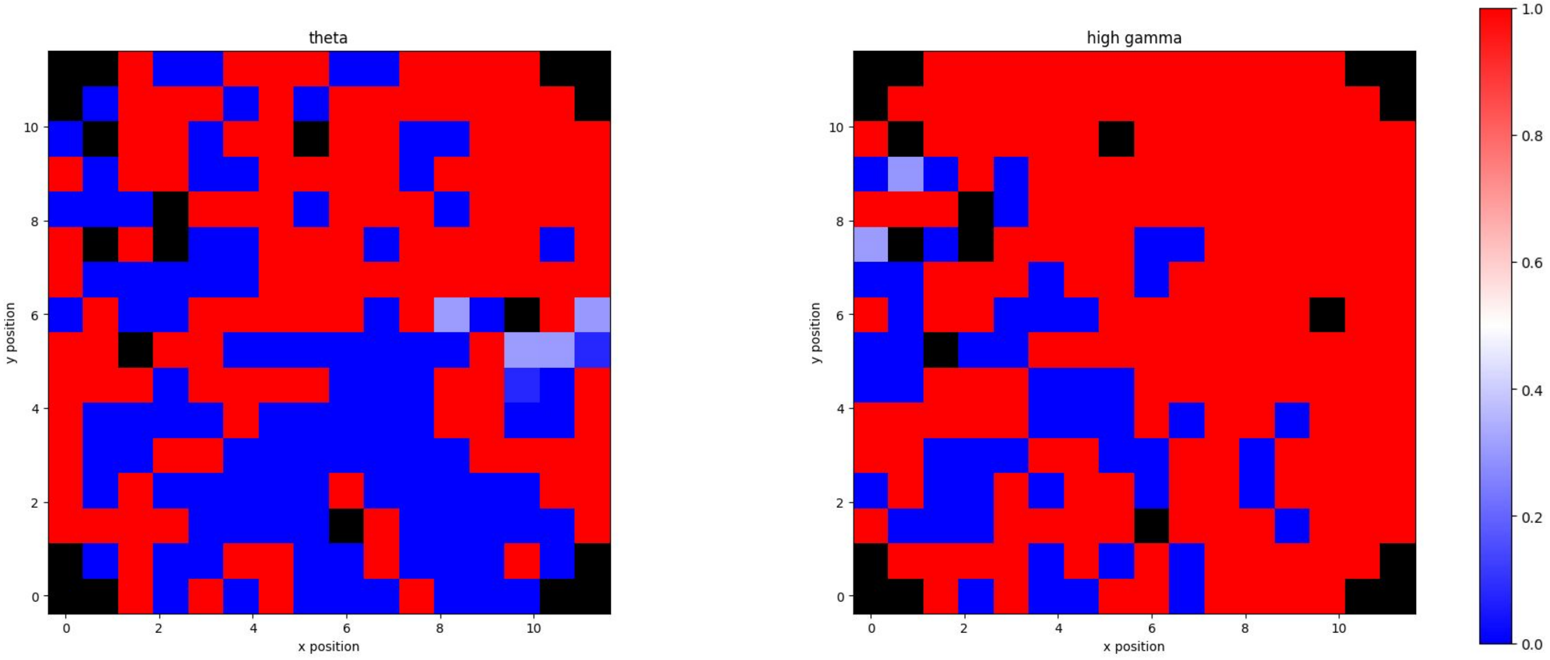
Convex Optimization

Input: neural data recorded from electrodes

H (observation transformation), q□ (observation covariance)

Output: binary value ranking the importance of each feature

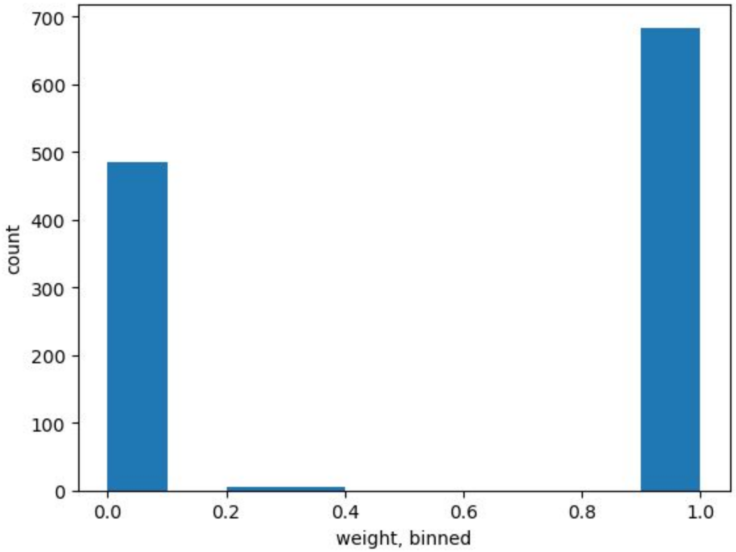
To understand which features the algorithm is selecting, we can compare spatial distributions



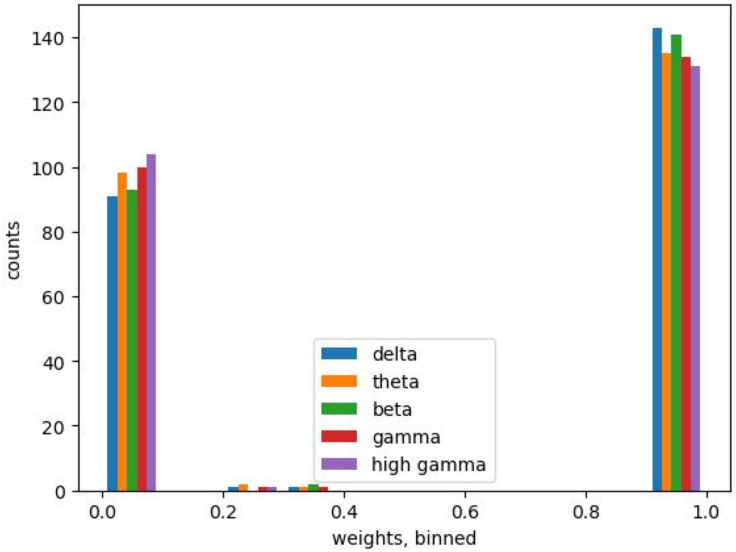
4-8 Hz

60-150 Hz

and we can compare intra- and inter-band distributions



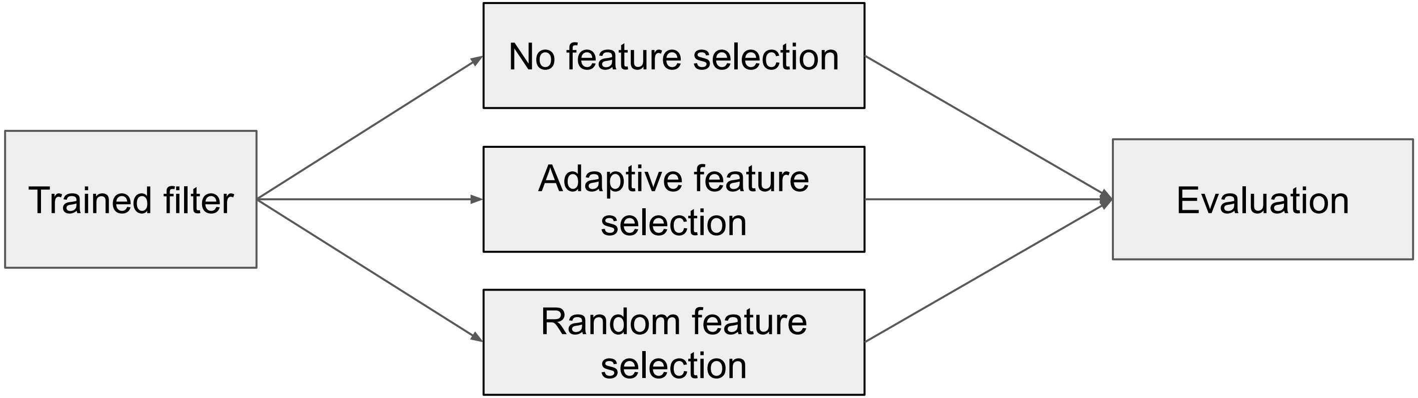
(all frequency bands together)



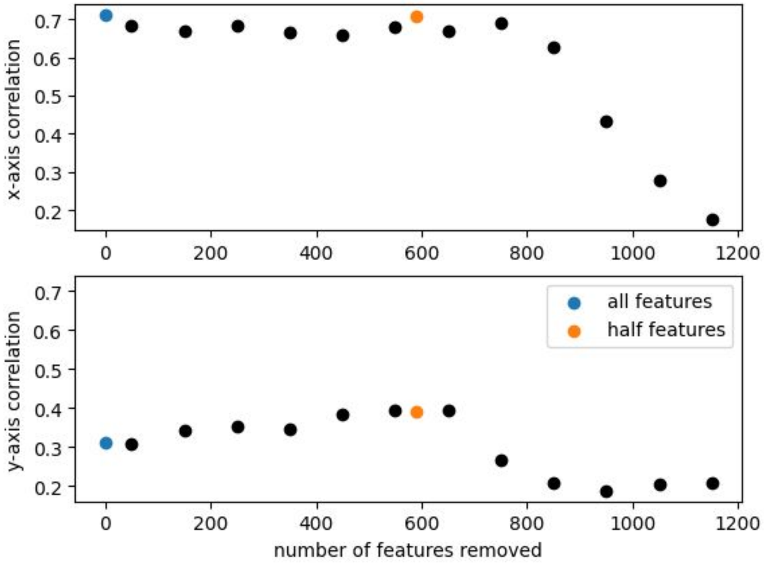
(separated by frequency bands)

Filter Evaluation

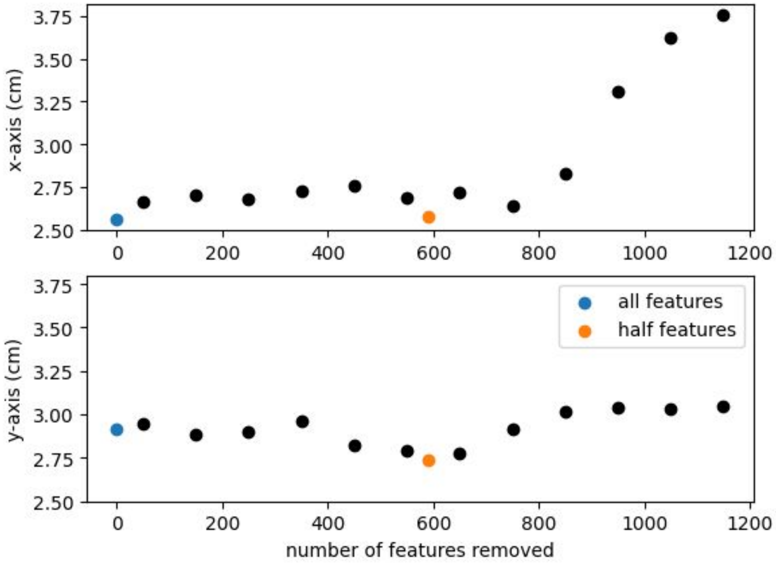
- Comparing predicted and actual cursor positions (separated by axis) with
  - Correlation
  - Root-mean-squared error
- How does filter performance change when we remove different numbers of features?



Correlation vs number of features removed



RMS vs number of features removed



In both metrics, we see a plateau of performance until around 600-700 features removed

Future Directions

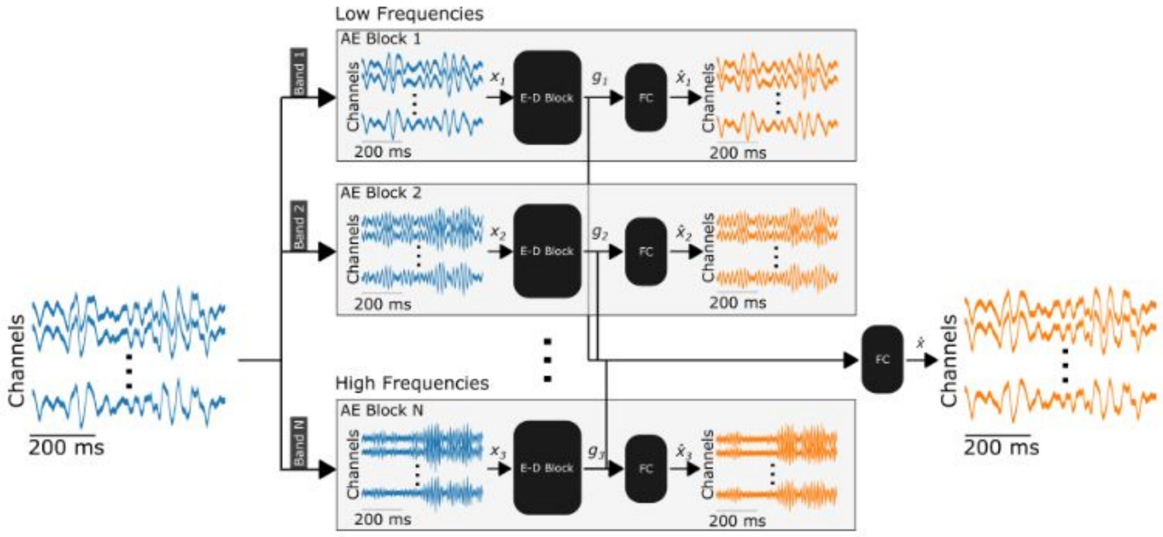
- Implement the random filter
    - Remove a random selection of features and compare performance
- Evaluate performance when changing sparsity and smoothness parameters
- Implement closed-loop system

ECOG-MRAE

Time Series Reconstruction

Aim: reconstruct non-linear time series data recorded from BCI

LFADS “model neural recordings as an observed projection from a latent dynamical system”



Current Work & Future Directions

- Working to replicate results and figures from the paper
- Later, evaluate model with other tests & other data

Extracurriculars