BIOE/ELEC 548: Final Project

Improving Neural Prosthetic System Performance by Combining Plan and Peri-Movement Activity

Renqin Yang

Report Due Date: 14 December, 2016
Submission Date: 10 December, 2016

On my Honor, I have neither given nor received any unauthorized aid on this report

Introduction:

Neural prosthetic systems aim to assist disabled patients by translating neural activity into control signals for prosthetic devices. Current neural prosthetic decode algorithms are based either on *plan* activity using maximum-likelihood (ML) techniques, or on *peri-movement* activity using linear filters or population vectors. In the paper "Improving Neural Prosthetic System Performance by Combining Plan and Peri-Movement Activity", the author proposed an estimation algorithm that decoded jointly using plan and peri-movement activity, and showed that decoding using **both** types of activity led to more accurate reconstructions of movement trajectories than when decoding based on plan or peri- movement activity alone.

I replicated the key results from the paper, the Fig.4 "Comparison of cross-validated decode performance " , demonstrating the differences and accuracy improvement by using both types of activity. I used the data from Homework 5 of BIOE/ELEC 548 "Neuron Signal Process and Machine Learning" to replicated the key results. Although, the final replicated results have minor discrepancy against the original results, I got the same conclusion with the paper from my replicated results.

Modeling Approaches:

Selecting training and testing dataset:

I just used the data from Homework 5, "ReachData.npz", and applied the same methods how to defined "plan", "move" and "combined" (undifferentiated rate model) windows and how to select training and testing dataset.

I randomly selected 400 trials (50 trials for each target) as training data, and also picked out remaining data as test data. Then, for both training and testing dataset, I counted spikes in plan, move and combined window individually, and its respective time with all the 190 neurons for 400 trials.

Collecting parameters for each direction:

I grouped different rates (combined, plan and move window rates) by eight targets, then calculated the mean and covariance matrix for each target through different rates. For any neurons whose averaged mean rates equals zero, deleted them from dataset (only for training dataset here), and record which neurons are deleted.

Data fitting and decoding:

I fitted the trial-by-trial firing rates and PC scores using a multivariate Gaussian distribution:

$$f(\mathbf{r} \mid d) = \frac{1}{(2\pi)^{n/2}|\Sigma_d|^{1/2}} e^{-\frac{1}{2}(\mathbf{r}-\boldsymbol{\mu_d})^{\mathrm{T}}\Sigma_d^{-1}(\mathbf{r}-\boldsymbol{\mu_d})}$$

And decoded reach direction using maximum likelihood(ignoring items which remain the same for each direction):

$$\begin{aligned} \widehat{d} &= \underset{d}{\operatorname{argmax}} P(d \mid \mathbf{r}) \\ &= \underset{d}{\operatorname{argmax}} \frac{f(\mathbf{r} \mid d)P(d)}{f(\mathbf{r})} \\ &= \underset{d}{\operatorname{argmax}} f(\mathbf{r} \mid d) \end{aligned}$$

Please note, I also deleted the same number and position of neurons, which deleted in the training dataset, for the testing dataset.

PC score:

While the firing rate captures the overall level of spiking activity in a given window, the PC score takes

into account the time-varying structure of activity in the window.

For each neuron of one trial, I used 5 ms bins to convert “SpikeTimes” array to impulse-like array which makes sure every neuron has the same time series. Then used Gaussian kernel (50 ms length) to convolve this impulse-like spike train for each neuron. Finally, performed PCA, and took the first PC score of each trial as the PC score for the trial.

Similarly, grouped different PC score (only plan and move window rates) by eight targets, then calculated the mean and covariance matrix for each target through different PC score. For any neurons whose averaged mean PC score equals zero, deleted them from dataset (only for training dataset here), and record which neurons are deleted.

Evaluating the performance:

I measured the performance both using absolute angular error and decoding accuracy.

$$e = |\theta_d - \theta_{\widehat{d}}|$$

Results:

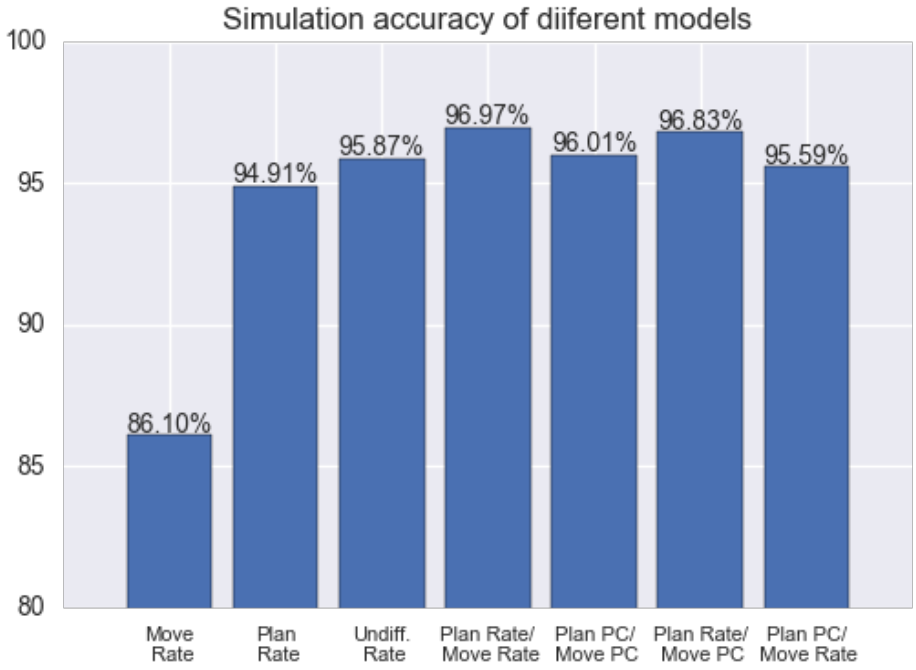

Fig1. Comparison of cross-validated decode performance.(By decoding accuracy)

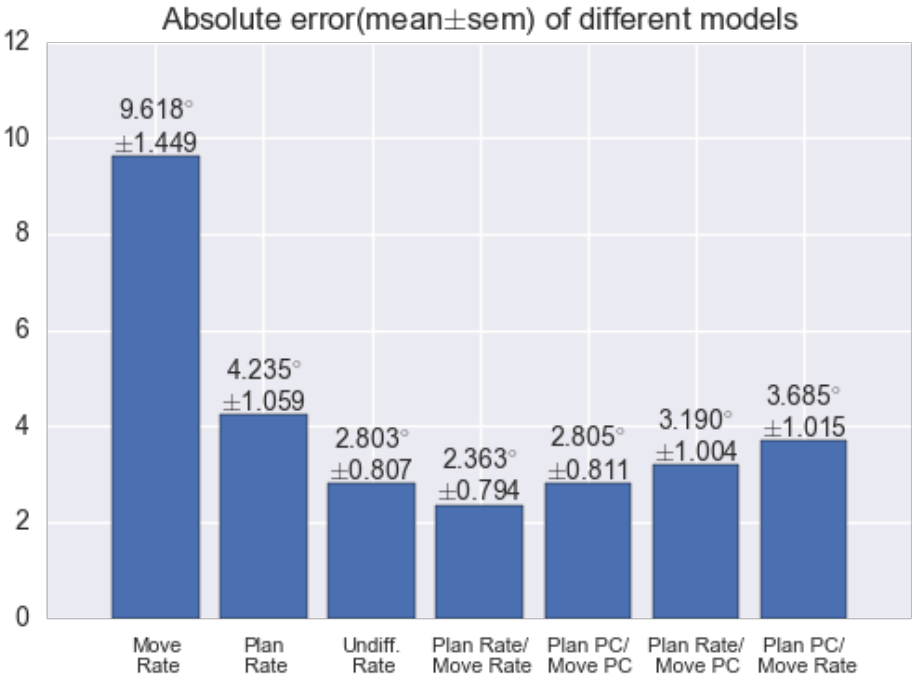

Fig2. Comparison of cross-validated decode performance.(By absolute angular error)

Discussion:

From the two figures based on my replication and modeling, there are two discrepancies against the results from the paper.

1). Not all models follow the rule: The more accurate simulated model got less absolute angular error.

2). According to the two figures, the most ideal model is Plan Rate/Move Rate model, rather than the Plan Rate/Move PC score model as the paper proposed.

However, I still got almost the same pattern and results with the paper. It does improve the accuracy and reduced the absolute angular error when using both plan and peri-movement activity separately (compared to the undifferentiated model). The minor discrepancy between the results from replication against original results from the paper may due to the different data used for modeling and decoding. Additional, the Plan Rate/Move PC score model also does improve a lot the accuracy and reduced, comparing to single Plan, Move or Undifferentiated model. And its accuracy is almost equal the most accurate model Plan Rate/Move Rate model. This Model still has the possibility and potential to make better and more accurate decoding.