Analysing Discrete-Time Neural Signals

Rouhollah Abolhasani* Mohammad Rabiei †

May 30, 2022

In this assignment, you will be instructed to perform several types of analysis on a simulated extracellular recording. These analyses include spike sorting and basic information estimation which aims at quantifying neuronal "decode-ability".
Keywords: neural data, neuronal data, signal processing, cognitive science, neuroscience, connectivity, electrophysiology, neural recording, single unit, multi unit, neural population, decoding, mgs, encoding, saccade preparation

1 Spike Preprocessing: Sorting

When measuring extracellular activities from a cortical area of interest, we often desire to inspect the electrical activities of single neurons as well as the aggregated electrical field induced due to the activities of a population of neurons that surround the inserted electrode. However, we cannot discriminate the activities of single neurons by just looking at the recorded data. Fortunately, by employing a sequence of mathematical techniques, we can accurately infer the spiking activities of every neuron that is sufficiently close to the electrode. This type of analysis, detecting single-unit activities from extracellular data, is called "spike sorting."

As depicted in Figure 1, there are several steps involved in the process of spike sorting. In a nutshell, those steps are the as follows: [1]

1: For a more detailed explanation of the procedure, refer to [1]

1. Filtering: Applying a bandpass filter filter between 300Hz and 3000Hz. Recorded electrical activities below 300Hz is usually referred to as low-frequency potential or "LFP".
2. Spike detection: Following the previous step, spikes are detected by applying an amplitude threshold on the filtered signal. The threshold in this step should be chosen carefully, for

* roohi.abol@gmail.com
† mohammad.rabiei@ut.ac.ir

picking higher values could lead to detecting no spikes, and choosing a relatively low value could lead to false-positive results due to noise crossing that threshold.

3. Feature extraction: When recording extracellular activities, neurons that are in vicinity of the inserted electrode are often observed to exhibit distinguishable patterns of electrical activity when they perform an action-potential. This gives us a clue to separate each detected spike into clusters each with almost similar waveform features/patterns. In other words, we employ feature extraction methods in order to transform spike waveforms into more informative feature-set of smaller dimensionality.
4. Clustering: The last step in spike sorting is to group the detected spikes into few clusters based on their extracted features. There are many clustering algorithms proposed for spike sorting, some of which you will use in this assignment.

In this question, you are to perform spike-sorting analysis on a single channel recording of a population of simulated neurons. Don't worry! You will be instructed through every steps of this analysis like a cookbook recipe! *But, before proceeding any further, please read this web page:* `http://www.scholarpedia.org/article/Spike_sorting`. [2]

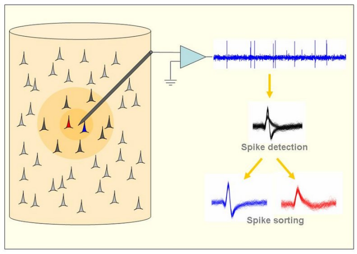


Figure 1: An illustration of the steps involved in spike sorting.[2]

2: **You must thoroughly explain your implementation details and results in your report. Additionally, note that analyzing your results and answering the subsequent questions constitutes a major proportion of your score for this part.**

1.1 Getting Started

(a) Load the dataset stored in `extracellular.mat`. Each data point is associated with the voltage amplitude recorded at a certain time (sampling rate of the recording is 2400Hz). Then, plot a diagram illustrating the amplitude against time.

(b) Plot the histogram of the recorded voltage amplitudes for the entire dataset. What can be inferred about the background noise by looking at this diagram (distribution, etc.)?

1.2 Filtering the Data

(c) Design a highpass Butterworth filter at 300Hz. Set the filter order to 7. Then, apply the filter on the data using MATLAB's `filtfilt` command. If you prefer Python, you can execute this command: `scipy.signal.filtfilt`.

(d) Plot two diagrams depicting: 1) the unfiltered data and 2) the data after applying the filter.

1.3 Detecting the Spike

(e) Using the following equations, calculate the voltage threshold (θ).

$$\theta = 5\sigma_n, \quad \sigma_n = \text{median}\left(\frac{|x|}{0.6745}\right)$$

(f) Extract the peaks in the data. A certain point in the data is considered a peak if the first derivative of former and later data points with respect to that certain point differ in sign.

(g) By comparing the amplitude of each peak point to θ, select each point whose amplitude is greater than or equal to the threshold. Next, extract the waveforms for each detected spike from the filtered signal. The waveform associated with each spike is a short timeseries that includes the data-points from 2ms before the peak to 2ms after the peak.

(h) Plot a diagram depicting the waveform for every detected spike in one single diagram. Are there any noticeable difference among the overall shapes of these waveforms? Store these waveforms in a 2D matrix for the next step.

1.4 Extracting Features

(i) Apply PCA on the waveforms matrix. In MATLAB, you can simply call `pca` command with proper arguments. Equally, you can invoke `sklearn.decomposition.PCA` function if you are programming in Python.

(j) By comparing the score for each principal component, choose the three most informative components (PC_1, PC_2, PC_3) for the next part.

1.5 Clustering the Spikes

(k) Using K-Means algorithm, cluster the waveforms based on PC_1, PC_2, and PC_3 features. You are not required to implement this algorithm yourself.

(l) Visualize the results by depicting three scatter plots for every possible pair of PC_1, PC_2, and PC_3 features. Each dot in these plots represents one waveform whose color indicates the cluster its associated waveform belongs to.

(m) Repeat part (k) and (l) with different values of K for the clustering algorithm. Which value of K gives the best results? Explain why.

Now, answer the following questions. Your must justify your answers by thoroughly analyzing the results and presenting proper diagrams in your report.

(n) Load `spikes.mat` file. This file contains a vector of time points in which true action potentials have occurred. Using this data and the spikes you detected previously, evaluate the performance of this spike-sorting pipeline. It is up to you to come up with a proper evaluation metric. Justify your answer by plotting diagrams and analyzing your observations.

(o) Use the following equation to determine a new threshold for detecting spikes (θ_{new}). X_t is the amplitude of data at time t.

$$\theta_{new} = 0.9 \times \max_{0 \le t \le T} (X_t)$$

Next, repeat the subsequent steps to sort the detected spikes. Do you think choosing the new threshold (θ_{new}) improved the results of spike-sorting? Justify your answer.

(p) Instead of using PCA algorithm to extract features, use tSNE algorithm (For MATLAB use `tsne` command and for Python, you may use `sklearn.manifold.TSNE`) and repeat the subsequent steps of spike-sorting pipeline. Do you observe any improvement in the results? Justify you answer.

2 Data Exploring: Discrete-Time Signals

As a new data modality, spike data requires modified visualizations. We will cover some of such visualizations used on discrete-time signals. From now on, we'll use Abolqasemi et al. [3] as our reference paper and refer to one session of its recordings as data. Have a look at the journal article for more information about data collection. In the zip file attached to this description, there exists a file named `cm`, containing the order of 16 locations presented to the subject monkey. Each of the `Spike*.mat` files contain the spike data of a specific electrode channel.

Assume that spikes are detected and sorted successfully. The output will be a binary spike train, with values equal to one whenever an action potential happens and zero elsewhere. The research questions that associate single-unit data are usually about how some external or internal event changes the firing dynamics of this spike train.

Raster Plot

The most basic visualization of spike data is called the raster plot. It represents every instance of a spike along the time axis as a small vertical line, while different trials of neurons overlayed on top of each other. So, a raster plot can be plotted for every neuron in the dataset, giving insight into how the neuron responded with time. One may rearrange the trials to place similar stimuli's representatives next to one another. Refer to 2 for an example of raster plot. example of raster plot.

Raster plots are very similar to event plots in other contexts.

Raster plots get hard to interpret as the difference between stimuli gets small. Moreover, they tend to become useless as the number of spikes increases, like when we're examining multi-unit clusters.

- For neurons given in the dataset, generate the raster plots. Choose one or two of the best figures and describe the response of the corresponding neuron(s).

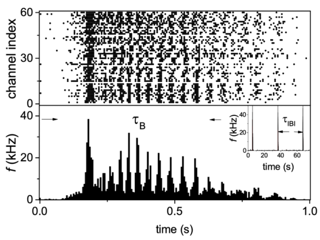


Figure 2: An example of raster plot (top) and PSTH (bottom) for a neuron during burstic activity. From [4]

Peri-Stimulus Time Histogram (PSTH)

Peri-Stimulus Time Histograms or Pre-Event Time Histograms (PETH) try to capture the temporal dynamics of neurons in a more interpretable discipline. Fundamentally, PSTH is simply the histogram of spike times of either a single neuron or a bundle of neurons. PSTHs can be grouped by stimulus type, providing an insightful visualization of neural selectivity.

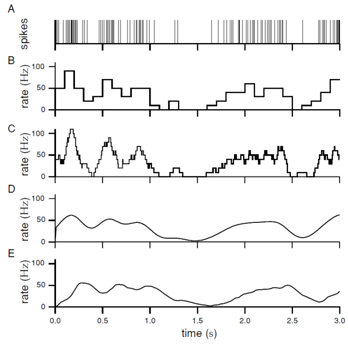
- For different types of stimulus in the dataset, draw the PSTH plots. Submit some of your best plots within your report.

Firing Rate

To make the PSTHs easier on the eyes, one can impose the idea of event-related potential plots on the neural data. Assuming that the firing rate of neural data is a stochastic process during the time, it may be estimated given various observations. By modeling the response of neuron n_i to stimulus s_j as a stochastic process like $r(t)+\sigma(t)$, in which σ is white noise, one can reduce the effect noise by averaging over repetitions of the same stimulus. Moving a kernel along the time axis can further reduce the noise. Such averaging over trials and small-time windows produce a time series (firing rate) corresponding to each stimulus type.

Select a responsive neuron from the data and:

- Plot firing rates over time for different stimulus angles.
- Plot firing rate over time for the stimuli with most and least responsiveness. Add a shadow corresponding to the confidence interval for each of the curves. Perform a test for evaluating the significance of difference for the selected pair over time. Explain why you have chosen that specific test.



Firing rates approximated by different procedures. (A) A spike train from a neuron in the inferotemporal cortex of a monkey recorded while that animal watched a video on a monitor under free viewing conditions. (B) Discrete-time firing rate obtained by binning time and counting spikes with Δt = 100 ms. (C) Approximate firing rate determined by sliding a rectangular window function along the spike train with Δt = 100 ms. (D) Approximate firing rate computed using a Gaussian window function with σ_t = 100 ms. (E) Approximate firing rate using the exponential window function with $1/\alpha$ = 100 ms. (Data from Baddeley et al., 1997. Figure from [5])

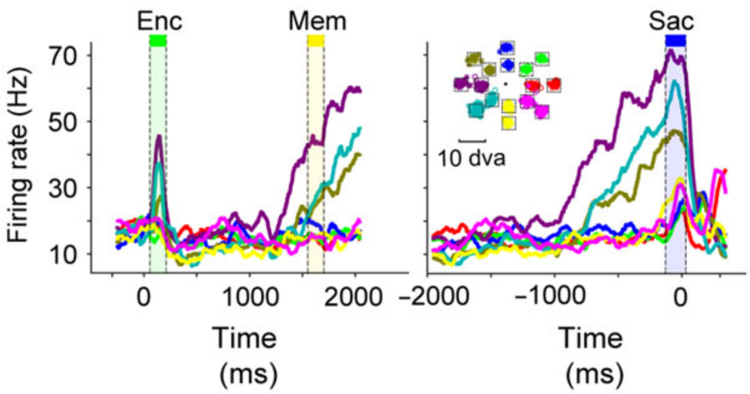


Figure 3: An example of estimated fire-rate for a sample neuron from [3]

Scatter Plots

In the context of spike data processing, scatter plots are widely used for displaying an effect in the neural population, e.g., comparing the information in neural decoding in different states (Fig. 4). We'll get to this in the following sections.

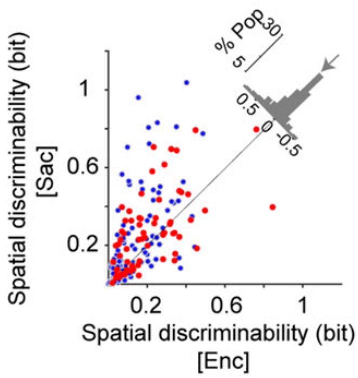


Figure 4: An example of scatter plot for comparison of spatial discriminability in an MGS task. From [3]

3 Single Units: Measuring the Information

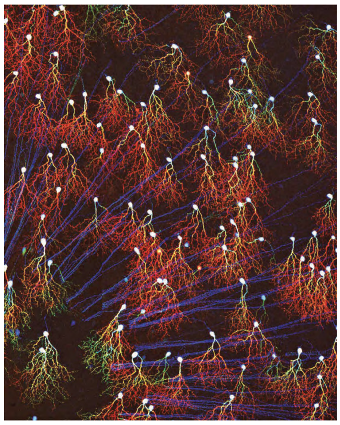
Here, we're trying to decode the information within spikes of each individual single units, disregarding the correlations they have as a population or network. One of the most common ways to quantify this metric is the mutual information, which is a metric derived from information theory.

Claude Elwood Shannon (April 30, 1916 – February 24, 2001) was an American mathematician, electrical engineer, and cryptographer known as a "father of information theory". Claude Shannon defined and analyzed Mutual Information (MI) in his landmark paper "A Mathematical Theory of Communication", although he did not call it "mutual information". This term was coined later by Robert Fano. Mutual Information is also known as information gain.

- Read the fourth chapter of Theoretical Neuroscience [5]. Explain how and why mutual information is a good measure for neural decoding.
- What are the disadvantages of using the mutual information?
- (bonus) Refering to [6], explain how the *finite sample size* bias corrupts the information estimation.
- Using the code provided, calculate the mutual information over time for given neural signals. Provide figures of your most informative neurons in the report.
- Compare the average of *encoding* and *saccade* stages of task (as defined in the reference paper) in a scatter plot. Test the significancy of difference and explain why you you chose that test.

4 Population of Units: Quantifying the Information

As you might be aware now, mutual information is not an easy metric to work with for population response. In place of that, some might choose other statistics to measure the neural signal information, like classifier accuracy, fisher information, separability index and etc. In this part, we will focus on the classifier accuracy, in particular support vector machines with linear kernel. After forming your design matrix in the form of $n_{\text{trial}} \times n_{\text{neuron}} \times n_{\text{sample}}$,



Depiction of neural population, from [7]

(Angles) Randomly, split the trials in 4 to 1 ratio. Train an SVM classifier on the larger batch and test it over the smaller one. Repeat this procedure for 200 times, each time with different random seed. Save the confusion matrix for all 8 angular locations at each time step. Plot the recall of the classifier over time for each category, as displayed in figure 5. (Data must be aligned to the stimulus onset)

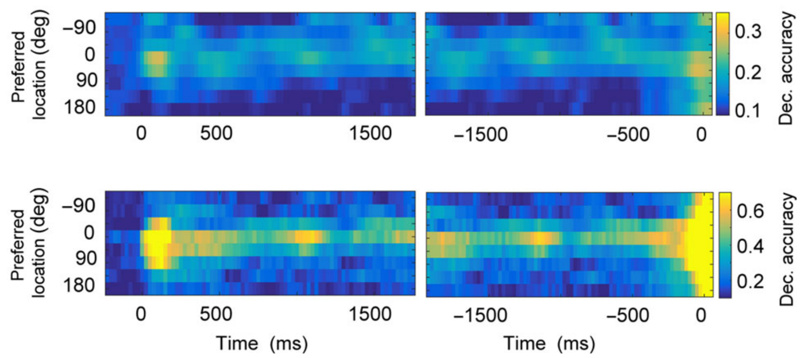


Figure 5: Recall of SVM classifier for onset-aligned and saccade-aligned data. From [3]

(Radius) Repeat the above procedure, but to decode the radius of stimuli location.

5 Submission

This assignment requires pdf report. Each student shall submit a typed report about the analysis procedure and results. For each question, at least one MATLAB or python script is required. Refer to the script name in your pdf report when you try to explain your code, but avoid putting your code snippets inside the report.

Each provided script must be runnable on the grader's system, and failing so causes loss in the question's mark. Provide the functions and nonstandard libraries you used alongside your scripts.

Assignment deadline is announced via elearn module. If you had problem understanding the questions, feel free to interact with the assignment's TAs via the e-mail address provided on the first page of assignment.

References

[1] Ramin Toosi, Mohammad Ali Akhaee, and Mohammad-Reza A Dehaqani. 'An automatic spike sorting algorithm based on adaptive spike detection and a mixture of skew-t distributions'. In: *Scientific Reports* 11.1 (2021), pp. 1–18 (cited on page 1).

[2] Rodrigo Quiroga. *Spike sorting*. 2007. DOI: 10.4249/scholarpedia.3583 (cited on page 2).

[3] Mohammad-Reza A Dehaqani et al. 'Selective changes in noise correlations contribute to an enhanced representation of saccadic targets in prefrontal neuronal ensembles'. In: *Cerebral Cortex* 28.8 (2018), pp. 3046–3063 (cited on pages 5, 6, 8).

[4] Yu-Ting Huang et al. 'Positive feedback and synchronized bursts in neuronal cultures'. In: *PloS one* 12.11 (2017), e0187276 (cited on page 5).

[5] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems.* MIT press, 2005 (cited on pages 6, 7).

[6] Stefano Panzeri et al. 'Correcting for the sampling bias problem in spike train information measures'. In: *Journal of neurophysiology* 98.3 (2007), pp. 1064–1072 (cited on page 7).

[7] Eric R Kandel et al. *Principles of neural science.* Vol. 4. McGraw-hill New York, 2000 (cited on page 8).

Some images in this report has been designed using resources from Freepik.com.