Neuronal Data Analysis

Getting into preprocessing and processing pipeline of neuronal data

Elahe Barat Vakili [*], Samin Mahdizadehsani [†], Mohammad Rabiei [‡]

April 18, 2023

Electrophysiology is a branch of biology that studies the electrical properties of biological cells and tissues. In neuroscience, electrophysiology is used to record the electrical activity of neurons in the brain. Neuronal data obtained through electrophysiology is critical for understanding how the brain processes information and how it generates behavior. The ability to record electrical signals from individual neurons has revolutionized our understanding of the brain and has led to the development of many powerful tools and techniques for studying brain function. In recent years, advances in electrophysiological techniques have allowed researchers to record from larger populations of neurons simultaneously, providing unprecedented insights into the collective activity of neurons in the brain. Electrophysiological data is an essential component of many neuroscience studies and has led to many important discoveries about the workings of the brain.

Keywords: electrophysiology, ephys, neuronal data, neural data, mutual information, d-prime, single unit activity, multi unit activity, population-level decoding, spike sorting, response dynamics, neural discriminability

1 Spike Sorting from Scratch

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."

[*] elahe.bvakili97@ut.ac.ir
[†] saminsani162@gmail.com
[‡] m.rabiei.gh@gmail.com

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. 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 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.

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

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]

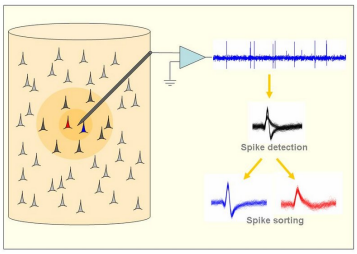


Figure 1: An illustration of the steps involved in spike sorting.[2]

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.**

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.)?

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.

Detecting the Spike

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

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

(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.

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.

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 Spike Sorting with ROSS

As we learned in the previous question, a crucial step in analyzing extracellular data is to differentiate between different neuron activities. However, performing spike sorting manually can be extremely time-consuming and laborious, especially for studies that involve multiple recording sessions. To address this issue, various toolboxes and software have been developed to facilitate the spike sorting process. One such software is ROSS! ROSS[1] is a MATLAB-based offline spike sorting software that enables researchers to perform automatic and manual spike sorting tasks efficiently. It provides various functions to modify the automatic sorting results, such as merging and denoising, and offers useful visualizations to help achieve better results.

1: Please feel free to star its repository on GitHub; it would make the developers very happy!

Before getting started, it is recommended to read the toolbox's documentation [2]. The data for this question is provided in file `ross-data.mat`. In this question, you will work with ROSS as follows:

2: Learn more about this toolbox by reading its corresponding paper [1]

Detection

The first step is activity detection. In this step, spikes are extracted from the bandpass filtered signal. A common way to detect spikes

is to determine activities that cross a threshold. The threshold is calculated based on the estimated noise power. You can adjust the default options, such as filter type and thresholding method, to ensure an accurate detection phase. The detected spikes and their corresponding times of occurrence can be saved as a MAT file.

- Load the raw extracellular data and adjust the provided settings for filtering and thresholding. Then, by clicking the "Start Detection" button, the detection results will appear in a PCA plot.

Auto Sorting

Auto sorting is a clustering method used to assign each spike to the corresponding neuron automatically. Here, a mixture of t-distributions, GMM, k-means, and template matching for clustering purposes is used. Additionally, a statistical filtering method for noise (outlier) removal and a rich alignment method are provided. The output of this section is cluster indices for each spike waveform, which can be saved as a MAT file.

- Use the auto-sorting feature and observe the results in PCA and waveform plots. How many clusters exist in the data?

Manual Sorting

Manual sorting provides several tools to modify the auto sorting output to improve it. These tools include Merge, Delete, Resort, Denoise, and Manual grouping or deleting in PCA domain of clusters. Additionally, you can tag the neurons with arbitrary comments for further analysis. The cluster indices, alongside the neuron tags, can be saved as a MAT file.

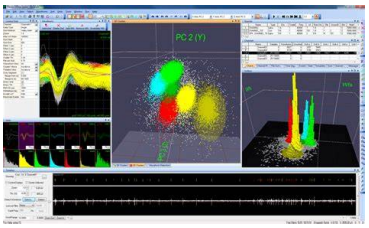


Figure 2: An example of spike sorting session in plexon software, one of the most commonly used spike sorting mediums. From `https://plexon.com/products/omniplex-software/`

- To make manual changes to the automatic results, select the 'Manual Sorting' tab. Then, select the 'Denoise' option and set the data plot percentage and denoising threshold to 85
- Select one of the clusters and apply the automatic sorting process to it. Does the selected cluster divide into new clusters? If so, how many?

Visualization

The software provides a rich set of visualization tools to help you better understand the analyzed extracellular data. These tools include, but are not limited to, Inter-spike interval, Neuron lifetime, Waveforms, 3D plot, and PCA domain plots. You can also track detected spikes on the raw data. In the 3D plot, you can choose arbitrary features for each axis, such as principal components and

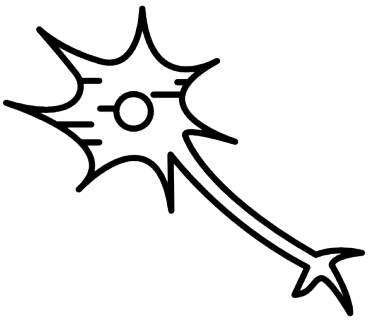
time of occurrence. Each neuron is assigned a unique color across all visualizations in the software, which is shown in the legend section of the main window.

Here are some suggested visualization tasks to perform:

- Plot the waveforms of the different clusters in separate plots and compare them to each other.
- Use a 3D plot to display the first two PCA components over time.
- Plot the entire raw data with the detected spikes and compare the waveform shapes of the different clusters.

These visualization tools will allow you to gain a deeper understanding of the data and provide valuable insights that can help improve your analysis.

3 Analysis of Single Neuron Activity



Understanding how individual neurons encode and process information is a fundamental question in neuroscience. One way to study neural responses is by analyzing patterns of action potential firing times, which serve as a code for conveying information. Single neuron analysis is a powerful tool for investigating the information processing capabilities of individual neurons, and two commonly used metrics are d-prime and mutual information. In this question, we will explore how d-prime and mutual information can be used to analyze the information processing capabilities of a single neuron in response to face and non-face stimuli.

We will use a dataset of trials from several neurons that were presented with face and non-face stimuli. The `face-data` file contains multi-unit activity data from 40 recording sites in the inferior temporal cortex of macaque monkeys, in response to visual stimuli. The data is structured as a 3D array with dimensions of $77 \times 40 \times 900$, representing the number of observations (stimuli), number of neurons, and time samples, respectively. The time axis spans 900 samples, from 200 milliseconds before stimulus onset to 700 milliseconds after, and the data was sampled at a rate of 1 kHz.

The `face-data-labels` file contains the categories of the presented stimuli. Each entry in the labels file corresponds to one of the 77 possible stimulus presentations in the data file. The label values are 0 for human face stimuli, 1 for monkey face stimuli, and 2 for non-face stimuli.

The data in `face-data` represents the spiking activity of each neuron, with 1s representing the occurrence of an action potential. To reduce inter-trial variability, repeated trials of the same stimulus were averaged, resulting in the data containing double numbers instead of just ones and zeros.

Figure 3: 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.

Visualization

Move an average window throughout the time axis, with the length of 50 samples and stride of 1, to reduce the noise of data and calculating the firing rate. Then, plot the time series of each neuron's response in time. Stratify the observations by the type of stimulus. See if you can find any difference in the time series. Include some of the best plots in your report.

Mutual Information

Mutual information is a concept used in information theory that measures the amount of information that two variables share. In the context of single unit analysis, mutual information can be used to quantify the amount of information that a neuron is encoding about a specific stimulus. Essentially, mutual information is a mea-

sure of the statistical dependence between the firing patterns of a neuron and the stimulus that is presented to it. By calculating the mutual information between a neuron's firing patterns and a stimulus, researchers can gain insight into the information processing capabilities of individual neurons and how they encode and process information. Mutual information is often used in conjunction with other metrics, such as d-prime, to gain a more comprehensive understanding of how neurons process information.

Refer to the fourth chapter of [3] for more details on using mutual information on neuronal data. Then, answer the following questions:

- What are some of the limitations and sources of bias for mutual information when applied in real-world practice?
- Using the time window averaged signal from the previous section, apply mutual information to each time bin's firing rate and estimate the amount of face information in each neuron's response. Then, average the mutual information across neurons and plot the results, including the mean and standard error of the mean.
- Compare the timing of information emergence in neurons with the onset of the rise in firing rate plots. What can be inferred from this comparison?

d-prime

D-prime is a commonly used metric in single unit analysis to quantify the information content of a neuron's response. It measures the separation between the probability distributions of the neural responses to two different stimuli, typically a signal and a noise stimulus. D-prime is calculated by taking the difference between the means of the two distributions and dividing it by the standard deviation of their pooled variance. A higher d-prime value indicates a stronger ability of a neuron to discriminate between the two stimuli, and therefore a higher information content in the neural response. D-prime is a useful measure for evaluating the sensitivity and selectivity of individual neurons in response to different stimuli, and can be applied in a wide range of sensory and cognitive tasks.

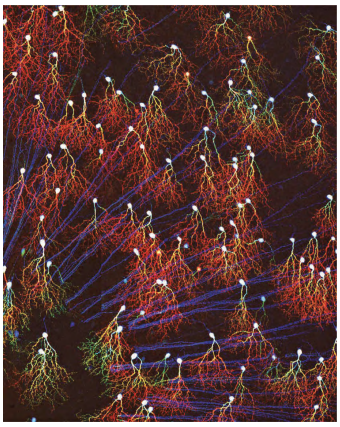
- What is noise in the context of single neuron analysis, and how does it impact the interpretation of experimental results?
- What is the theoretical basis for using d-prime as a measure of neural sensitivity to a stimulus?
- What are some limitations of using d-prime as a measure of neural sensitivity?
- Estimate the d-prime over time for given neural signals using face labels (face vs. non-face). Then again, average the

d-prime across neurons and plot the results, including the mean and standard error of the mean.
- Compare the onset and peak timings of d-prime, mutual information and firing-rate?
- Are the d-prime and mutual information results completely congruent?

Overview

- Is there a relationship between the d-prime or mutual information of a single neuron's response to a stimulus and the overall firing rate of the neuron?
- How does changing the stimulus intensity affect the d-prime of a single neuron's response to the stimulus?
- Can the d-prime of a single neuron's response be used to predict the presence or absence of a particular stimulus in a population of neurons?
- Does the d-prime of a single neuron's response to a stimulus vary across different levels of background noise?

4 Analysis of Population Activity



Population decoding is a technique commonly used in neuroscience to infer the stimulus or action being performed by an animal based on the activity of a population of neurons. The basic idea behind population decoding is that different stimuli or actions evoke different patterns of neural activity, and these patterns can be decoded to infer the underlying stimulus or action. This approach is particularly useful for studying the neural basis of perception, cognition, and motor control, as it allows researchers to infer the animal's mental state or behavior from the activity of multiple neurons simultaneously. Population decoding has been used to study a wide range of neural systems, from simple sensory systems in insects to complex cognitive and motor systems in primates.

Population decoding is a commonly used method in neuroscience that allows us to infer the stimulus or behavior from the collective activity of a population of neurons. It is a statistical method that uses the spiking activity of multiple neurons to decode or reconstruct the sensory or behavioral information that is being processed by the brain.

The basic idea of population decoding is to train a statistical model, often a linear classifier or regression, to predict the stimulus or behavior from the spiking activity of multiple neurons. The model is trained on a set of known stimuli or behaviors, with the goal of learning the relationship between the spiking activity and the stimulus or behavior.

Once the model is trained, it can be used to decode or reconstruct the stimulus or behavior from the spiking activity of the same or different population of neurons. This decoding process involves applying the statistical model to the spiking activity to infer the most likely stimulus or behavior given the observed activity.

Population decoding has been used to study a wide range of sensory and cognitive processes, including visual perception, auditory perception, motor planning, decision-making, and memory retrieval. It has also been used to develop brain-machine interfaces, where the spiking activity of neurons is used to control external devices such as prosthetic limbs.

One of the main advantages of population decoding is that it allows us to study the neural code at the population level, which can reveal more complex and subtle relationships between the spiking activity and the stimulus or behavior than can be revealed by single neuron analysis. However, population decoding is also subject to certain limitations, such as the assumption of linearity and the need for large amounts of data to achieve high decoding accuracy.

- What are some of the different types of classifiers that can be used for population-level decoding, and how do they differ in terms of their strengths and limitations?
- How can different features of neural activity patterns, such as spike counts or spike timing, be used as inputs to classifiers for decoding?
- What are some of the performance metrics used to evaluate the accuracy of classifiers for population-level decoding, and how are they calculated?
- What are some of the factors that can impact the performance of classifiers for population-level decoding, such as the number of neurons, the size of the neural population, or the stimulus conditions?
- Train a support vector machine on each time window, and evaluate the performance of classifier there. Visualize the results.
- Processes like random initialization of machine learning algorithms and stochasticity of train-test splitting can add unreliablity to classification output. To account for that, one can run the code with various random "seed"s, and then average these separate runs. Provide a time serie of classifier performance over time estimated by this manner[1]
- Train an LDA classifier on the data. Compare the results with that of the SVM [2].

1: Don't forget to overlay the mean with empirical confidence interval! It is almost useless to report the mean of a distribution while ignoring its variation.

2: classifier performance, timing of onset, timing of peak

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 pages 2, 4).

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

[3] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. MIT press, 2005 (cited on page 8).