Introduction to Computational Neuroscience
Practice II: Data Analysis - Continuous and Spiking data

Aqeel Labash, Ardi Tampuu, Dainel Majoral, Ilya Kuzovkin, Raul Vicente

February 11, 2018

1 Introduction

In this practice session we go through how to work with recordings from brain areas (EEG) and individual cells (intracranial recording, spiking data). In the lecture we also covered some other ways of recording brain activity, such as fMRI and MEG. We do not have the time to look at the data from these imaging techniques, but if you find them fascinating, you can work with them in your course project.

2 EEG data

Those who made it to the psychology lab visit on Monday, already know how recording EEG (Electroencephalography) signal looks like. For those who were not present: electrodes are placed on different parts of your head and electrical signal is recorded from them. Certain electrodes (on the earlobes, face) do not record the brain activity but are instead used as a reference to filter out noise caused by to muscle activity and electrical signals in the room. When all the electrodes are attached the person looks like this:

After some preprocessing (filtering out noise with the help of reference electrodes), this is how an usual EEG recording of one channel (signal from one electrode is called a channel) looks like:
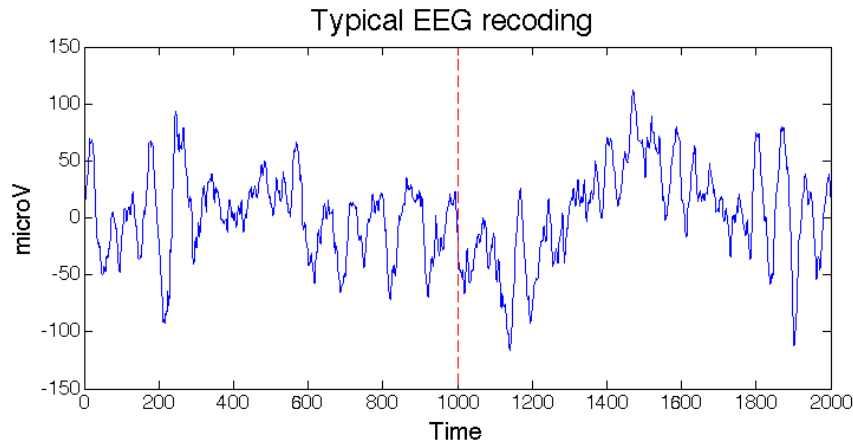
Figure 1: On the x axis there is time, on the y axis strength of the recorded signal in μV. The dashed line at the time point $x = 1000$ is the moment when the *stimulus* (picture) was shown to the test subject.

Exercise 1: Event Related Potential (ERP) (1pt)

Event related potentials refer to changes in the brain activity in response to certain events. For example, we expect some change in activity of some regions after a person hears a sound.

On the plot above, the test subject was presented with a stimulus at T=1000. Nevertheless, from the plot above one cannot conclude that showing the stimulus had any considerable effect on the subject's brain activity (as measured by this specific electrode). Even though there seems to be a voltage increase starting at T=1200 and ending at the T=1500, it could be a random event not related to the stimulus. Clearly the signal is very noisy.

To remove the noise and reach conclusions on the effect of the stimulus, scientists conduct the same experiment several times and then average the results. In each trial the noise is different and will cancel out if we average over trials. Therefore the brain response, if it is there, will reveal itself much more clearly.

In the `data` folder you have file `erptrials.csv`. One row is one trial recorded for 2 seconds with sampling rate of 1000 Hz. The stimulus was shown at the time point 1000 (1 second). There are 79 trials in this file. Your task is to plot an average of all 79 trials to see that there is a clear ERP response or not. Also add a red vertical line at T=1000 like on Figure1. Respond to the following questions: i) How many milliseconds after the stimulus presentation does the ERP happen? ii) How long does it take for the EEG signal to return to normal range?

In your response you are supposed to include the code used for generating the plot and the plot itself, with title and axis names.

Exercise 2: Frequency Analysis (2pt)

The most popular operation that you can do with continuous brain data such as EEG is converting it from the *time domain* to the *frequency domain*. Due to the fact that any function can be represented as a sum of sinusoids (Fourier transform) we can decompose our signal into such sinusoids and observe from which frequency components it consists. In terms of the brain data such transformation makes particular sense, because of the *brain rhythms* – different frequencies of the firings of the neurons are related to the different kinds of mental activity[1].

In this exercise we will plot the *power spectrum* (like the two plots below) and see how *alpha wave* (brain oscillation at 8-13Hz) emerges when the test subject's eyes are closed. In the `data` folder you can find two files: `eyes_open.csv` and `eyes_closed.csv`. The data is recorded from

---

[1] http://en.wikipedia.org/wiki/Electroencephalography#Wave_patterns

the channel `Pz` (you can google where it is located). One row contains 4 seconds of the signal, sampling rate is 512. Each file has 15 recordings.

Your task is to perform Fourier analysis on both datasets, plot power spectra and compare the results. Do it as follows:

1. Plot some of the signals just to see how they look like.

2. For each recorded signal (2048 data points):

    (a) Use `fft(signal)` to compute power spectrum of this signal. You will get a vector of complex numbers.

    (b) Use `abs(result_of_fft)` to obtain the magnitude[2]. `abs(result_of_fft)`^{2} will give you *power*.

3. Sum together the 15 power spectrum distributions, that you got from the previous step.

4. And divide the resulting vector by 15 to obtain the average.

5. Plot it. You will see that the right part of the graph is mirror image of the left part. Discard the right part.

6. Your x axis goes from 1 to 1024, which does not correspond to actual frequencies. Compute correct x axis as follows:

```
dt = 1/512;              % time step length in ms
df = 1/4;                % frequency step in Hz, 4 corresponds to
                         % the length of recording
fNQ = 1/dt/2;            % fNQ is the maximal frequency, in our case it is
                         % 256 if you have discarded part of the data in step 5
xaxis = (0:df:fNQ-df);   % points for your X axis, should be of the same length
                         % as the vector of frequencies
```

7. And plot it again

```
plot(xaxis, powers_average);
```

Now your axis goes from 0 to 256 (or 512 if you did not discard the right side of the graph) with step of 0.25. This corresponds to the frequency range `pow` function produced.

After making the plot more beautiful and focusing only on the range from 0 to 30 Hz you should obtain a result that looks something like this:

[2]`http://www.mathworks.se/help/matlab/ref/abs.html`
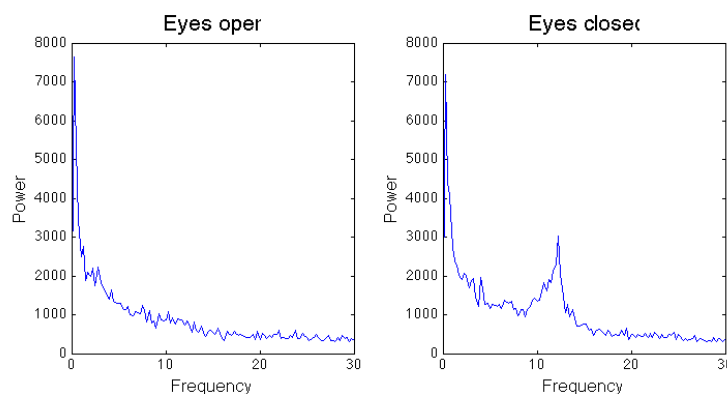
3 Part II: Spiking Data

I the first part we looked at continuous brain data: strength of the EEG signal was changing over the time. Another very popular form of data, obtained when we insert electrodes inside the brain and record the activity of individual neurons, is *spiking* data. The concept is very simple: we attach a sensor to the neuron and whenever the neuron emits and action potential (aka "fires") we write a "1" into our data file, otherwise we write a "0". The resulting data shows us for each time point (in our case each millisecond) whether the neuron has fired or not.

Dataset

The dataset we will be working with has the spiking data recorded while the mouse was presented with a stimulus: moving bars on the screen, which can have different orientations. On Figure 2 you can see an example: white and black bars are tilted (*orientation*). The bars also move in the direction perpendicular to the tilt during the experiment. The spiking activity is recorded from 72 neurons and we want to find whether different neurons react differently to moving bars with different orientations. Have a look at Figure 3 – spiking pattern of this particular neuron seems to indicate that this neuron is more active when the bar is tilted 45° clockwise from the horizontal position. And its activity

Figure 2: Example of the stimulus. During the actual experiment the bars are moving.

fades when the orientation changes away from 45. We would like to rediscover from the data, that indeed different neurons respond differently to different orientations.
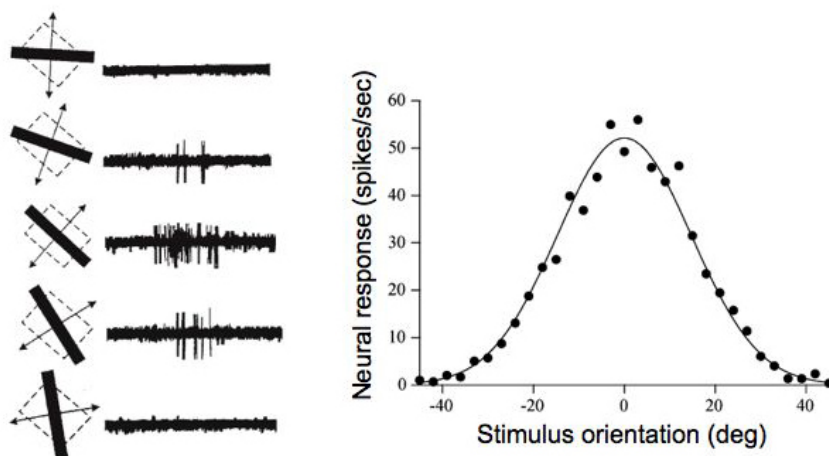

Figure 3: Neuronal response to different orientations of the bar.

NB: Note that in the dataset we have 13 stimuli (13 orientations), but the first one should be ignored (it has orientation $-1°$). So you should only use the other 12.

Exercise 3: Raster plots (2pt)

Let us start by plotting some spiking data. Under the `data/lgn` folder you have recordings from 72 neurons of a mouse. `Matlab/Octave` users can use .mat files in the `matlab` folder, the same data is available in the plain text format in the `plain` folder.

`Matlab` files have names `mlgnori_NN.mat` where NN is the number of the neuron. When you load it to your environment you will find that each file has a structure with two elements: `spktimes` and `stim`. The first one is $M \times N \times T$ matrix where M is the number of different stimuli, N is the number of trials and T is the time in milliseconds. So this is a 3-dimensional matrix containing values 0 (no spike) and 1 (spike) for every millisecond in every trial for every

stimulus. A value at position (m,n,t) will therefore tell you if this neuron was active at time t, in trial n of stimulus m. The second element, `stim`, describes the stimulus. Open it and you will see what information is in there.

If you do not use `Matlab`, then it is easier for you to use the plain text files. They have names `neuron_NN_stimulus_SS.csv` where `NN` is the number of the neuron and `SS` is the number of the stimulus. Inside each file one line represents one trial. For each millisecond it has value 0 (no spike) or 1 (spike). Files with names `stimulus_SS.csv` describe the stimulus: they hold four values:

1. Time before the stimulus (in seconds).

2. Duration of the stimulus (in seconds).

3. Time after the stimulus (in seconds).

4. Orientation of the stimulus (in degrees).

Hint: There is some code in the *code* folder . **Your task is** to take any of the neurons and plot all trials as a raster plot (see Figure 4). You will notice that neuronal response to the stimulus varies a lot (lot of noise!), this is why you usually need several trials.
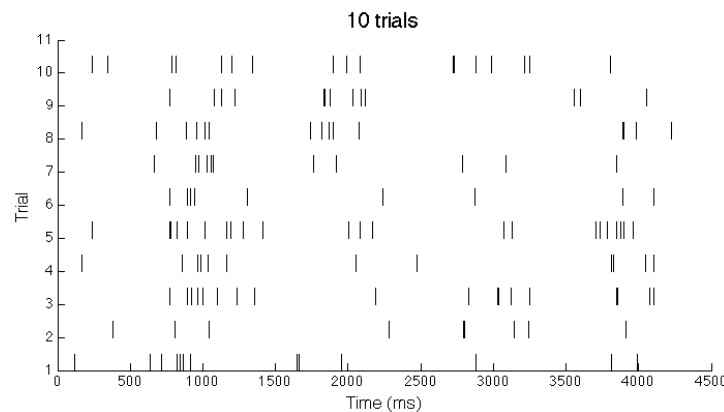

Figure 4: Raster plot of 10 trials on the same neuron.

On the X axis we have time and on the Y axis trials. A vertical bar indicates a spike in that trial. To draw a vertical line in `Matlab` you can use

```
line([t t], [tr tr+0.5], 'Color', 'k');
```

where `t` is the time and `tr` is the number of the trial. If necessary modify the length and width of the bars to make the raster plot more readable.

The next step is to **create two more raster plots**. The first one will illustrate the behaviour of all the neurons under the same stimulus, while the other one will have responses of the same neuron but to the different stimuli.

Create the following two plots:

1. Raster plot, where on X axis we have time and on Y axis we have all 72 neurons. Vertical bars are the responses to a stimulus (choose any) by the corresponding neuron in any of the trials (!). Please note that in our dataset different recordings have different length, but this should not be a problem. **(neuron to neuron variability)**

2. Raster plot, where on Y axis we have all 12 stimuli and bars are the responses from the same neuron (choose any). **(variability across stimuli)**

5

Hint: *For the first of the two plots we need to average spiking data over the trials so that each neuron would be represented by only one vector, not 10. Simplest way to do that is just to add trials together. Let us say that you have 10 trials, and each of them is a vector of 4500 time points. You just sum those vectors together to obtain one vector of length 4500. After that replace all values which are grater than 1 with 1 (meaning if there was a spike in that time point in at least one of the trials).*

Exercise 4: Tuning Curve as Rose Diagram (2pt)

Hint: There is some code for this exercise in *code* folder.

From the lecture you must be familiar with the term *receptive field*. *Tuning curve* is a plot that helps to describe the receptive field of a neuron with respect to some variable - how strongly does the neuron respond (how often it fires) if we give the variable different values. Figure 3 describes how a neuron's response varies for different orientations of the bar, the plot on the right is the tuning curve of this neuron (with respect to orientation).

For orientations of bars there is a really neat way to visualize the tuning curve. This visualization method is called *rose chart* or *angle histogram*. You can see an example on Figure 5. The idea is that the values are placed on the circumference on the circle and the length of the sector is determined by the number of times the value appears in the list. It is like a histogram drawn in circle.

In our case it allows to represent our data in a much more natural way because different orientations form a circle. In our case the lengths of the sectors correspond to sum of spikes in this orientation. There are rose charts for some of the neurons on Figure 5. We can clearly see that neuron 8 reacts more to the orientation of 0°, the neuron 6 is most active in the range of 270° to 330° and so on.
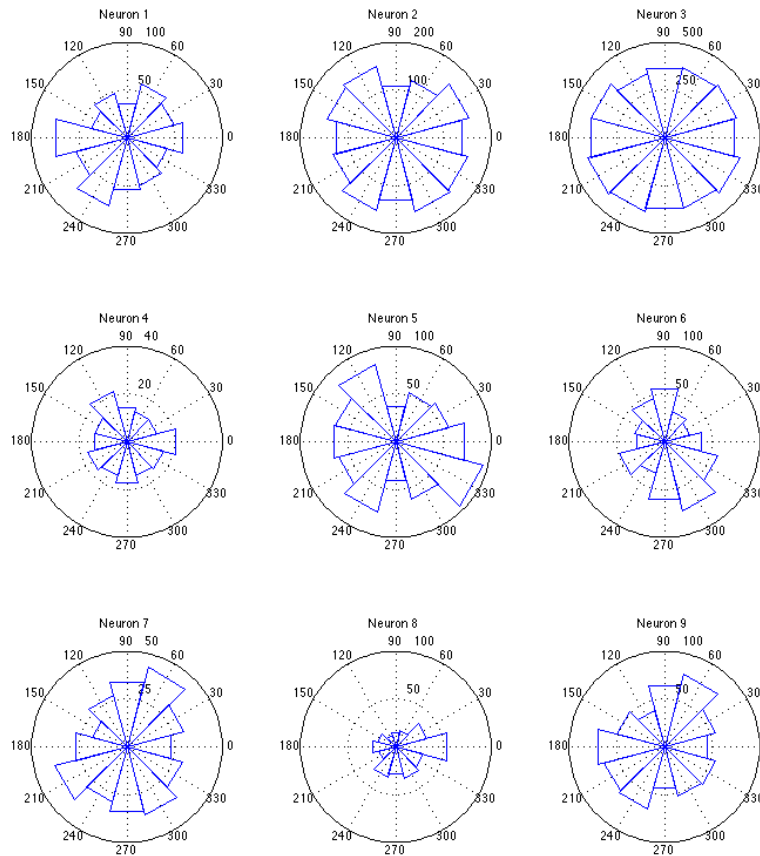

Figure 5: Rose diagram: the number of spikes for each of the 12 orientations.

Your task is to produce similar plot for any 9 neurons (not the same ones as on previous figure). To produce a plot for one neuron do the following:

1. Create an array `A` where for each spike you will store the orientation when the spike occurred.

2. For each orientation (stimulus) do:

   (a) Count the number of times the neuron fired.
   (b) Insert current orientation (in degrees) into vector `A` as many times as the neuron has fired.

3. Draw the plot. In `Matlab` you can do:

   (a) Array with the angles of bins (needed for the `rose` function):
   ```
   angles = degtorad([0, 30, 60, 90, 120, 150, 180, 210, 240, 270, 300, 330]);
   ```
   (b) Call `rose(deg2rad(A), angles)`

4. Also, look through the diagrams you obtain and point out which neuron has the most specific tuning curve and which direction it prefers.

———————————————— End of obligatory exercises ————————————————-

Exercise 5*: What else our neurons are tuned to? (Bonus: up to 1pt)
In this session we have seen an example of how neurons are "tuned" to respond to very specific stimuli. You task is to find other interesting examples of stimuli our neurons are tuned to react to. Are there special neurons, which fire when you look at a human face? Neurons which react on the temperature? Hunger? Numbers?

Find the most interesting examples (from humans, animals, insects). Write at least half a page (images, charts, plots are recommended, but do not count in this half a page).

Exercise 6*: Post-Stimulus Time Histogram (PSTH) (Bonus 1pt)
Another useful analysis tool is a histogram, where on X axis we have time points (or time ranges) and on Y axis the number of spikes that occurred during given time range. It is called *Post-Stimulus Time Histogram (PSTH)*.

1. Choose any neuron, any stimulus.

2. Take an average over all trials as we did before.

3. Choose time window, for example 20ms or 50ms.

4. Plot a histogram, where on the X axis we have time windows and on the Y axis the numbers of spikes that occurred during those windows.

Please submit a `pdf` report with the answers to the questions, plots and comments about your solutions. Also submit the code for the programming exercise(s). Pack those into the `zip` archive and upload to the course web page.