18-698 / 42-632
Neural Signal Processing
Prof. Byron Yu — Spring 2023

Problem Set 5

This problem set is due on **Tuesday, April 4, 11:59pm**.

Be sure to show your work and include all **Matlab code** and plots. **Any Matlab questions without code will receive no credit.**

If you have questions, please post them on the Piazza Q&A webpage, rather than emailing the course staff. This will allow other students with the same question to see the response and any ensuing discussion.

Please submit your work as a **single PDF file** on Gradescope, which is linked from Canvas. When preparing your solutions, please complete each problem on a **separate page**. Gradescope will ask you select the pages that contain the solution to each problem.

Submissions can be written in LaTeX or they can be handwritten and photocopied using a scanner or smartphone camera. Handwritten work should be clearly labeled and legible.

The dataset for the following problems can be found on Canvas under "Files → Data sets → ps5_data.mat". When you load the .mat file, you will find the following variables:

`RealWaveform`: 10 seconds of real electrode data[1] sampled at 30 kHz. Values are in μV.

`InitTwoClusters_1, InitTwoClusters_2, InitThreeClusters_1, InitThreeClusters_2`: each is a 31 x K matrix, where the kth column is the initialization of the kth cluster center, $\boldsymbol{\mu}_k$ ($k = 1, \ldots, K$).

1. **(10 points)** Data pre-proccessing

   Plot the entire waveform. You will notice a low frequency component in the signal, known as the local field potential (LFP). We are not interested in analyzing the LFP during spike sorting. Remove the LFP using a high-pass filter, stopping frequencies below 250 Hz. You may find the following code useful for designing a simple high-pass filter:

```
x = RealWaveform;
f_0 = 30000; % sampling rate of waveform (Hz)
f_stop = 250; % stop frequency (Hz)
```

---

[1] The neural data have been generously provided by the laboratory of Prof. Krishna Shenoy at Stanford University. The data are to be used exclusively for educational purposes in this course.

```
f_Nyquist = f_0/2; % the Nyquist limit
n = length(x);
f_all = linspace(-f_Nyquist,f_Nyquist,n);
desired_response = ones(n,1);
desired_response(abs(f_all)<=f_stop) = 0;
x_filtered = real(ifft(fft(x).*fftshift(desired_response)));
```

Plot the filtered waveform. Was the low-frequency component removed?

You can listen to the neurons with this command (only supported on some architectures):

```
sound(x*.97/max(abs(x)),f_0);
```

2. **(15 points)** Spike detection

   Set a threshold, $V_{\text{thresh}} = 250\ \mu\text{V}$, and determine all of the times that the signal crosses from below to above the threshold. Plot the threshold as a line across your plot of the high-pass filtered waveform from Problem 1.

   Take 1 ms snippets of the waveform beginning 0.3 ms before each threshold crossing. Each snippet should have 31 samples; the tenth sample should be less than V_{thresh}, and the eleventh sample should be greater than V_{thresh}. In a new figure, make a "voltage versus time" plot containing the following:

   - all of the threshold-crossing waveform snippets, and
   - the threshold as a horizontal line.

   Your result should look similar to Figure 3 from Lewicki's review paper, with the key difference that your plot should show multiple spike shapes (instead of just one). You will also notice a few non-stereotyped traces resulting from either different neurons spiking near the same time or noise exceeding threshold.

3. Clustering with the K-means algorithm

   Implement the K-means algorithm in MATLAB, and use it to determine the neuron responsible for each recorded spike.

   Treat each snippet as a point $\mathbf{x}_n \in \mathbf{R}^D$ ($n = 1, ..., N$), where $D = 31$ is the number of samples in each snippet, and N is the number of detected spikes. In this problem, we will assume that there are $K = 2$ neurons contributing spikes to the recorded waveform. Initialize the cluster centers using `InitTwoClusters_1`.

   (a) **(25 points)** For each cluster ($k = 1, 2$), create a separate "voltage versus time" plot containing the following:

- the cluster center $\boldsymbol{\mu}_k$ returned by K-means as a red waveform trace (i.e., the prototypical action potential for the kth neuron),
- all of the waveform snippets assigned to the kth neuron, and
- the threshold as a horizontal line.

(b) **(20 points)** Plot the objective function J versus iteration number (as in Figure 9.2 in *PRML*). How many iterations did it take for K-means to converge?

4. **(10 points)** As discussed in class, K-means guarantees convergence to a local optimum. Thus, it is possible to converge to different local optima with different initializations. Repeat Problem 3 where the cluster centers are initialized using `InitTwoClusters_2`. Is the local optimum found here the same or different as that found in Problem 3?

5. With real data, we don't typically know how many neurons contribute spikes to a given recorded waveform. We will soon discuss in class how to infer the number of neurons from the data, a procedure which typically requires running the clustering algorithm on the same data for different values of K. In this problem, we will assume that there are $K = 3$ (rather than $K = 2$) neurons contributing spikes to the recorded waveform.

   (a) **(10 points)** Repeat Problem 3 with $K = 3$ and initializing using `InitThreeClusters_1`. How does the local optimum found here differ from that found in Problem 3?

   (b) **(10 points)** Repeat Problem 3 with $K = 3$ and initializing using `InitThreeClusters_2`. Is the local optimum found here the same or different as that found in part (a)?