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

Spring 2023

Problem Set 8

This problem set is due on **Friday, April 28, 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 → ps8_data.mat". When you load the .mat file, you will find the following variables:

Xplan: a 728×97 matrix of real data, where each row is a spike count vector across 97 simultaneously-recorded neurons[1]. This is exactly the same data analyzed in Problem Set 3, but we are providing you here with the spike counts rather than the raw spike trains. As in Problem Set 2, the spike counts are taken in a 200 ms bin during the plan period. There are 91 trials for each of 8 reaching angles, for a total of 728 trials. Trials 1 to 91 correspond to reaching angle 1, trials 92 to 182 correspond to reach angle 2, etc.

Xsim: a 8×2 matrix of simulated data, where each row is a data point.

1. Visualization of high-dimensional neural activity using PCA

   In Problem Set 3, we classified neural activity from 97 neurons across 8 different reaching angles with greater than 90% accuracy. This indicates that there were 8 well-separated clusters of data points in 97-dimensional space. However, we did not attempt to visualize these clusters due to the high dimensionality of the data. Here, we will apply PCA to these same data (**Xplan**) to gain some intuition about why we were

---

[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.

able to classify so well in Problem Set 3. The data points are $\mathbf{x}_n \in \mathbf{R}^D$ $(n = 1, ..., N)$, where $D = 97$ is the data dimensionality and $N = 728$ is the number of data points.

(a) **(5 points)** Plot the square-rooted eigenvalue spectrum. If you had to identify an elbow in the eigenvalue spectrum, how many dominant eigenvalues would there be? What percentage of the overall variance is captured by the top 3 principal components?

(b) **(10 points)** For the purposes of visualization, we'll consider the PC space defined by the top $M = 3$ eigenvectors. Project the data into the three-dimensional PC space. Plot the projected points in Matlab using 'plot3', and color each dot appropriately according to reaching angle (there should be a total of 728 dots). Use your mouse to rotate the three-dimensional plot in Matlab. Show a view in which the clusters are well-separated.

(c) **(5 points)** Define a matrix $U_M \in \mathbf{R}^{D \times M}$ containing the top three eigenvectors (i.e., PC directions), where $U_M(\mathtt{d}, \mathtt{m})$ indicates the contribution of the $\mathtt{d}$th neuron to the $\mathtt{m}$th principal component. Show the values in U_M by calling 'imagesc(U_M)'. (Note: Also call 'colorbar' to show the scale.) Are there are any obvious groupings among the neurons in each column of U_M, as hypothesized in class?

2. From PCA to PPCA and FA

We will compare PCA, PPCA, and FA on a toy problem using the data in Xsim. The data points are $\mathbf{x}_n \in \mathbf{R}^D$ $(n = 1, ..., N)$, where $D = 2$ is the data dimensionality and $N = 8$ is the number of data points. We will project the data into a $M = 1$ dimensional space.

(a) **(10 points)** Create one plot containing all of the following for PCA:

- Plot each data point $\mathbf{x}_n$ as a black dot in a two-dimensional space.
- Plot the mean of the data $\boldsymbol{\mu}$ as a big green point.
- Plot the PC space defined by $\mathbf{u}_1$ as a black line. (Hint: This line should pass through $\boldsymbol{\mu}$.)
- Project each data point into the PC space, and plot each projected data point $\hat{\mathbf{x}}_n$ as a red dot. (Hint: The projected points should lie on the $\mathbf{u}_1$ line.)
- Connect each data point $\mathbf{x}_n$ with its projection $\hat{\mathbf{x}}_n$ using a red line. (Hint: The red lines should be orthogonal to the PC space. To see this, you will need to call 'axis equal' in Matlab.)

(b) **(30 points)** Implement the EM algorithm for PPCA in Matlab, and run the algorithm on the data in Xsim. Plot the log data likelihood ($\sum_{n=1}^{N} \log P(\mathbf{x}_n)$) versus EM iteration. (Hint: The log data likelihood should increase monotonically with EM iteration. You should run enough EM iterations to see a long, flat plateau.)

(c) **(5 points)** Using the parameters found in part (b), what is the PPCA covariance $(WW^{\mathrm{T}} + \sigma^2 I)$? If you did part (b) correctly, the PPCA covariance should be very similar to the sample covariance.

(d) **(10 points)** Create one plot containing all of the following for PPCA:

- Plot each data point $\mathbf{x}_n$ as a black dot in a two-dimensional space.
- Plot the mean of the data $\boldsymbol{\mu}$ as a big green point.
- The PC space found by PPCA is defined by W, which in this case is a two-dimensional vector. Check that PC space defined by W is identical to that found by PCA. Plot the PC space as a black line. (Hint: If there is some small discrepancy between the PC spaces found by PCA and PPCA, run more EM iterations for PPCA.)
- Project each data point into the PC space using PPCA, and plot each projected data point $\hat{\mathbf{x}}_n = W E[\mathbf{z}_n \mid \mathbf{x}_n] + \boldsymbol{\mu}$ as a red dot. (Hint: The projected points should lie on the line defined by W.)
- Connect each data point $\mathbf{x}_n$ with its projection $\hat{\mathbf{x}}_n$ using a red line. Why are the red lines no longer orthogonal to the PC space?

(e) **(10 points)** Implement EM algorithm for FA. You should be able to do this with only a small modification to your PPCA code. Run the algorithm on the data in **Xsim**. Plot the log data likelihood ($\sum_{n=1}^{N} \log P(\mathbf{x}_n)$) versus EM iteration. (Hint: The log data likelihood should increase monotonically with EM iteration. You should run enough EM iterations to see a long, flat plateau.)

(f) **(5 points)** Using the parameters found in part (e), what is the FA covariance $(WW^{\mathrm{T}} + \Psi)$? If you did part (e) correctly, the FA covariance should be very similar to the sample covariance.

(g) **(10 points)** Create one plot containing all of the following for FA:

- Plot each data point $\mathbf{x}_n$ as a black dot in a two-dimensional space.
- Plot the mean of the data $\boldsymbol{\mu}$ as a big green point.
- The low-dimensional space found by FA is defined by W, which in this case is a two-dimensional vector. Plot the low-dimensional space as a black line. (Hint: This line should pass through $\boldsymbol{\mu}$.) Why is the low-dimensional space found by FA different from that found by PCA and PPCA?
- Project each data point into the low-dimensional space using FA, and plot each projected data point $\hat{\mathbf{x}}_n = W E[\mathbf{z}_n \mid \mathbf{x}_n] + \boldsymbol{\mu}$ as a red dot. (Hint: The projected points should lie on the line defined by W.)
- Connect each data point $\mathbf{x}_n$ with its projection $\hat{\mathbf{x}}_n$ using a red line.