AI 1104: Programming for AI

Assignment 2

Due on: 24 April 2024, 11:59 PM IST.

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Read each question carefully, and answer all parts. Write well-documented codes with contextual variable names. Put the codes (.py or .ipynb) and plots (.png, .jpg, .jpeg, .pdf) of all the questions into one single folder, compress this folder, and upload the .zip file. You should name the .zip file as `AI1104_assignment_2_<your roll number>.zip`.

Data set credits: Sripati and Olson, 2010.
Assignment credits: Prof. Rajesh Sundaresan, Department of ECE, IISc.
See here for a graduate-level course on Data Analytics taught at IISc. This assignment is part of the module on visual neuroscience and oddball detection in the course.

1 Data Set Description

1. **Search times data (.csv file):**

   This file contains data on search times (in ms) for the image pairs shown in Figure 1.

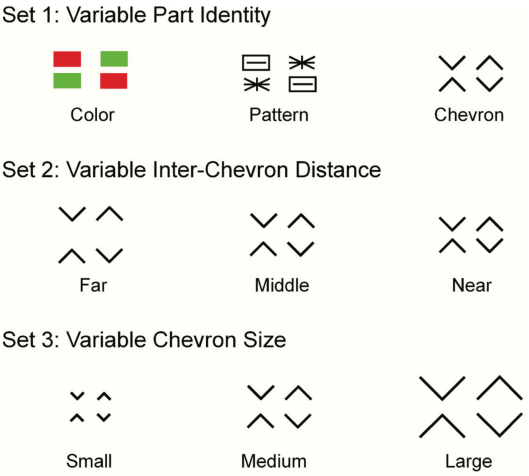

Figure 1: Collection of images on which the search times and firing rates data was collected.

   (a) There are three sets of data. Each set has six groups of experiments. Each group is for a fixed oddball image against its distracting image.

   (b) Column A contains search times of six individuals for the "Set 1 - Colour" pair in Figure 1. The oddball image is the picture on the left in the figure (Set 1 - Colour, left image with red on top of green). This column is labeled Oddball L. Column B contains the search times when the oddball image is the one on the right (Oddball R, green on top). Each individual was presented this oddball test 12 times in random order. The data is similarly arranged for columns C through R (Sets 1 through 3).

2. **Firing rates data (.csv file):**

   This file contains data on the firing rates obtained from rheseus macaque monkeys that were shown each of the 18 images in Figure 1 separately for 200ms.

   (a) Each set has a **different number** of neurons whose firing rates were recorded; the exact number of neurons corresponds to the number of rows of numerical data for that set.

(b) Consider columns A, B. Column A records the average firing rates for 174 neurons when the image "Set 1 - Colour - left image with red on top" was displayed standalone for 200ms, while Column B refers to the average firing rates of 174 neurons when the image "Set 1 - Colour - right image with green on top" was displayed standalone for 200ms.

(c) Column M contains the average firing rates of the appropriate number of neurons (equal to the number of rows of numerical data available in column S) when the image "Set 3 - Small - L" was displayed standalone, while Column N is for the image "Set 3 - Small - R".

2 Assignment

1. Computation of average search delays, relative entropy distance per neuron, and ℓ_1 distance per neuron.

   (a) For each image pair in Figure 1, compute the average search delay. Remember to subtract 328 ms (the so-called *baseline reaction time* of human subjects) from each value in the data set in your computations. Store the average search delays in an array named `average_search_delays`.

   (b) Given an oddball image i with firing rates $\boldsymbol{\lambda}_i = [\lambda_i(1), \ldots, \lambda_i(N)]^\top$ and a distracter image j with firing rates $\boldsymbol{\lambda}_j = [\lambda_j(1), \ldots, \lambda_j(N)]^\top$, where N is the number of neurons from which the firing rates are recorded, the *relative entropy distance per neuron* between images i and j is defined as

   $$D_{ij} \coloneqq \frac{1}{N} \sum_{n=1}^{N} \left[\lambda_i(n) \log \frac{\lambda_i(n)}{\lambda_j(n)} - \lambda_i(n) + \lambda_j(n) \right]. \tag{1}$$

   The logarithm in (1) is natural logarithm.

   For each image pair in Figure 1, compute the relative entropy distance per neuron. Store the values in an array named `relative_entropy_distances`.

   (c) Given an oddball image i with firing rates $\boldsymbol{\lambda}_i = [\lambda_i(1), \ldots, \lambda_i(N)]^\top$ and a distracter image j with firing rates $\boldsymbol{\lambda}_j = [\lambda_j(1), \ldots, \lambda_j(N)]^\top$, where N is the number of neurons from which the firing rates are recorded, the ℓ_1 *distance per neuron* between images i and j is defined as

   $$L_{ij} \coloneqq \frac{1}{N} \sum_{n=1}^{N} |\lambda_i(n) - \lambda_j(n)|. \tag{2}$$

   For each image pair in Figure 1, compute the ℓ_1 distance per neuron. Store the values in an array named `L_1_distances`.

2. Fitting a straight line passing through the origin.

   (a) With relative entropy distances on the x-axis and **inverse** of average search delays on the y-axis, find the best straight line passing through the origin that fits the data. Print the slope of this line.

   (b) With ℓ_1 distances on the x-axis and **inverse** of average search delays on the y-axis, find the best straight line passing through the origin that fits the data. Print its slope.

   Which of the above two lines is a better fit?

3. Fitting a Gamma distribution to the search delays.

   A general note on Gamma distribution: The Gamma distribution Gamma(a, b) has two parameters (shape a and rate b). It also has the property that the standard deviation to mean ratio is not arbitrary, but tied to the shape parameter a. If the shape parameter is 1, we get the Exponential distribution. If the shape parameter is 2, we get a random variable that is the sum of two exponential random variables with the identical rate parameters, and so on. In general, the shape parameter need not be an integer.

   The mean and standard deviation of Gamma(a, b) are respectively $\mu = a/b$ and $\sigma = \sqrt{a}/b$.

   (a) Randomly select half the number of columns in the search times data set. For each column selected, compute the mean and standard deviation of its values (after subtracting 328 ms from the search times). Store the mean values in an array named `mean_search_delays` and the standard deviation values in an array named `std_search_delays`. Plot the standard deviation of the search delays against their means. From the plot, estimate the shape parameter and print its value.

(b) On each of the columns that did not contribute to the shape parameter estimation, randomly select one half of the samples. Using all the selected samples, estimate the rate parameter and print its value.

(c) From each of the columns used for 2(b), collect the remaining half of the samples that did not contribute to the rate parameter estimation. Accumulate all these samples into an array named `search_delays`. Plot the empirical cdf of the samples in `search_delays`. On the same figure, plot the Gamma(a, b) cdf, say G, where a is the shape parameter estimated in 2(a) and b is the rate parameter estimated in 2(b). Print the value of the so-called *Kolmogorov–Smirnov test statistic*

$$\mathsf{KS}(F, G) = \max_x |F(x) - G(x)|,$$

where the maximum is over all x belonging to the array `search_delays`.

Note: If you plan to use `stats.scipy.gamma` to plot the Gamma CDF, be sure to substitute the `scale` field with the **inverse** of the rate parameter.