Data Analytics - Assignment-V

Anirban Biswas (Sr.No. - 14382)

November 5, 2018

Study on Perceptual Distance

The task for the first problem in the assignment is to measure the *Relative Entropy* distance for different pairs of images in the oddball detection experiment. In addition to that, the $L1$ distance and the *Average Search Delay* needs to be calculated. Both the distance metrics are to be evaluated using the firing rate data of the neurons. The second task is to find out the ratio of AM and GM of spread of the products of the distance metrics.

Part-I & II : Relation between Search Delay and Distance

The search time dataset has a total of 24 columns (4 sets with each set having 6 groups). The experiment is conducted on 24 persons and each person is shown 6 variations of a oddball-distractor pair, making row count for search data to be 144.

To calculate the average search time for a column c, the following equation is employed:

$$AverageSearchDelay_c = \frac{1}{144}\sum_{i=1}^{144}(SearchData_c^i - BaselineReactionTime)$$

In the above equation, the super-script i denotes the search data for the i^{th} row in the c^{th} column written as sub-script c. The baseline search time for this particular problem is 328 ms.

The neuronal firing data is used to measure the relative entopy and L1 distances. The relative entropy and L1 distances are obtained using the definition provided as in the lecture slides. The firing data has a total of 30 columns - 3 sets each containing 6 groups and one set having 12 groups. As per the instructions given, for each column of the first three sets we get in total 18 relative entropy values. The last set, which has 12 columns, we use *compound search* strategy to get the distance measures.

Compound Search Strategy : Here is the technique I followed to get the relative entropy (or L1 distance) values. Let us take the example of Bug-Worm

pair. The data for the column where the oddball is bug and distractor is worm, the observation data may come from any of the following four cases -

- Oddball Bug - Distractor Worm
- Oddball Big - Distractor Worm flipped
- Oddball Bug flipped - Distractor Worm
- Oddball Bug flipped - Distractor Worm flipped

All the four possible cases are equally likely. Hence, I have calculated the relative entropy for all these and took a simple average of them to obtain the final entropy value corresponding to the particular column. So, for all 12 columns corresponding to 4 sets, we'll be getting $\frac{12}{4} * 4 = 12$ data points.

Results

Finally, we have obtained 24 data points of average search time and their corresponding relative entropy distance (and L1 distance). Our hypothesis was that the reciprocal of search delay is proportional to the perceptual distance. The following two plots demonstrates the relationship of inverse search time(s^{-1}) vs $RelativeEntropy$ distance (and s^{-1} vs $L1$ distance).

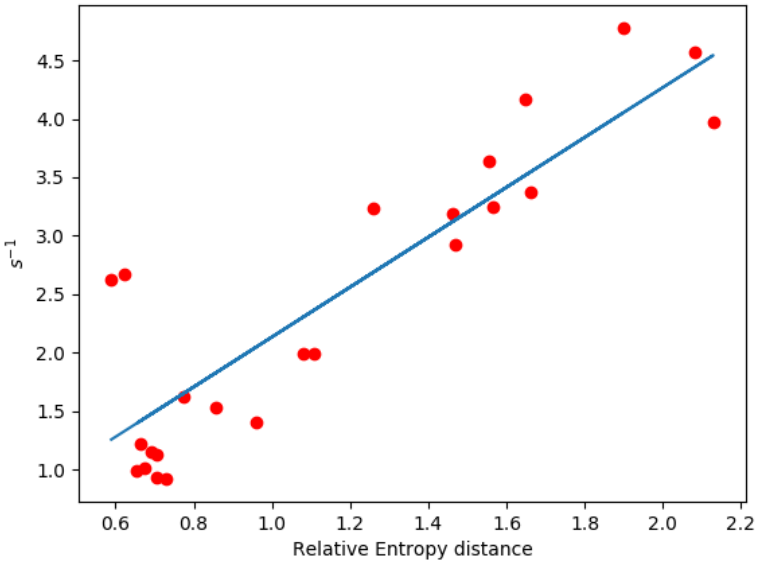

Figure 1: Scatter plot depicting the relationship between inverse search delay and relative entropy distance measure. The blue straight line best fits the points by minimizing squared error loss.

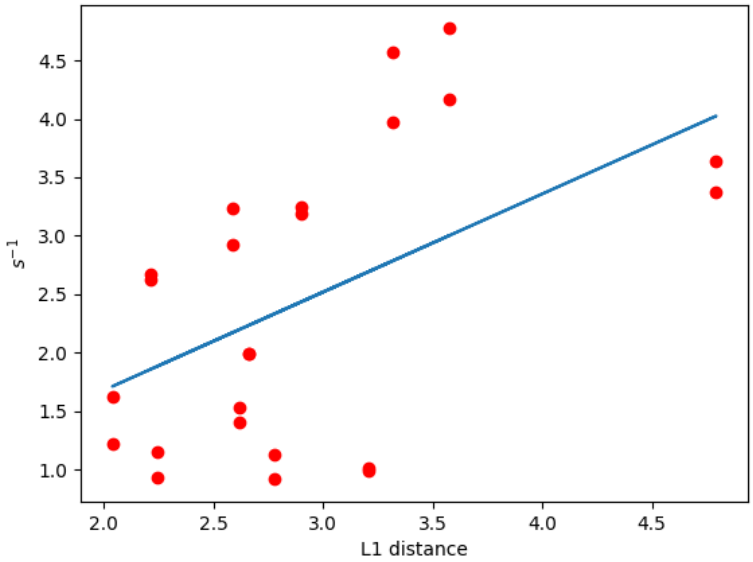

Figure 2: Scatter plot depicting the relationship between inverse search delay and L1 distance measure. The blue straight line best fits the points by minimizing squared error loss.

By looking at the above two figures, it is very clear that the straight line fit is much better for relative entropy metric. Hence, it is evident that **relative entropy is a better metric than l1 distance**.

The second problem asks for the AM/GM measure of the spread of the products : *search delay* $\times$ *relative entropy* and *search delay* $\times$ *L1 distance*. For the relative entropy case, the value turns out to be **1.129** and for l1 distance the value is **1.042** . As the AM/GM ratio is a measure of the spread of data, and the value is lower for the former, we can confidently say that **relative entropy is a better distance metric compared to L1 distance**.

Part-III : Fitting Gamma Distribution

The task of this problem is to fit a **Gamma Distribution**, $\Gamma(\alpha, \beta)$, on the search times and to estimate the shape(α) and rate(β) parameter of the probability distribution. Additionally, comparison between the CDF of the original distribution and CDF obtained from a subset of data points in search time data needs to be done.

To solve the first part of this problem, I have randomly selected half of the groups and calculated the mean and standard deviation for each of them. So,

we have 12 pairs of *(mean, standard deviation)* values. These values are plotted using *matplotlib* package. Here is how it looks like :

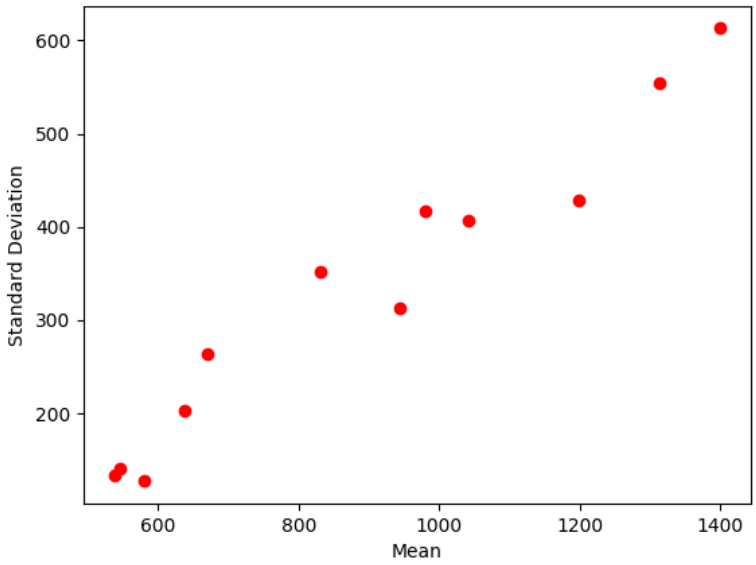

Figure 3: Scatter plot of mean vs. standard deviation of search delays. The plot follows a linear trend - standard deviation varies almost proportionately as mean changes.

Estimation of Shape parameter : To estimate the shape parameter, a straight line fitting is done on the previous plot by minimizing the squared error loss. This is done with the help of *numpy.polyfit* method. The slope of the straight line gives us the shape parameter. The estimated value for **shape parameter is 3.716**. Hence, $\boxed{\alpha = 3.716}$

Estimation of Rate parameter : Next, from each of the remaining 12 groups in search time data, half of the samples are randomly selected and these points are used to estimate the rate parameter. The estimation is done by using the minimization of the squared error loss of variance (as a linear function of mean) using *numpy.ployfit*, because for gamma distribution, $Mean = \frac{\alpha}{\beta}$ and $Variance = \frac{\alpha}{\beta^2}$. Hence, $\beta = \frac{Mean}{Variance}$.

The estimated value for **Rate parameter is 0.00163**. $\boxed{\beta = 0.00173}$

Empirical CDF and Kolomogorov-Smirnov Statistic

In the third part of the problem, first we have plotted the cumulative distribution function (CDF) using those data points which we have not used yet. Also, the CDF of gamma distribution with the estimated shape and rate parameters is plotted in the same curve.

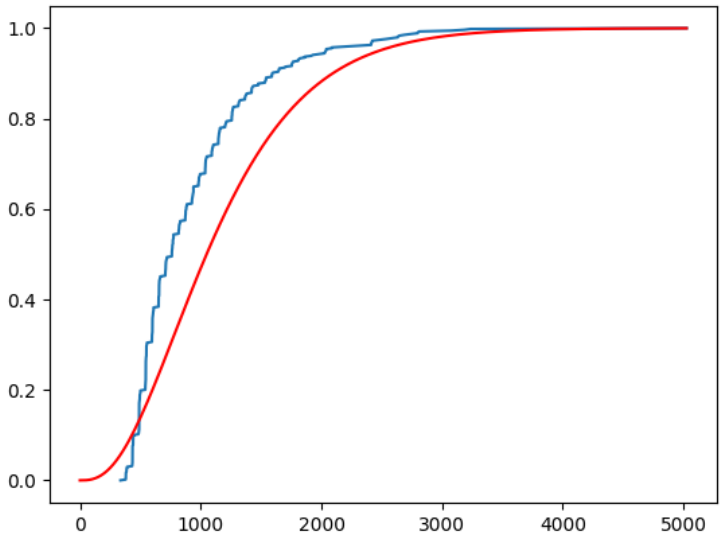

Figure 4: CDF of gamma distribution. The blue curve is generated from the remaining data samples. The red curve corresponds to the CDF of gamma distribution using the estimated shape and rate parameter.

In statistics, the Kolmogorov–Smirnov test (K–S test or KS test) is a non-parametric test to quantify a distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution, or between the empirical distribution functions of two samples.

In our experiment, to comapare the CDF of the empirical gamma distribution and gamma distribution using estimated shape and rate parameter, I have used *scipy.stats.ks_2samp* API. The Kolmogorov-Smirnov statistic obtained is as follows :

statistic=0.19814814814814818, pvalue=8.925723250937718e-10

As the K-S statistic is small, then we can accept the hypothesis that the distributions of the two samples are the same. Therefore we can say with a high confidence that search delays follow a gamma distribution.

Part-IV : Expected Search Time under controlled scenario

The task for this question is to find out the lower bound on the expected search time, given image 0 is the oddball at location l (hypothesis $h = (l, 0, 1)$) under the special circumstance when the subject can control where to look at the beginning of each time slot.

Solution Approach

As mentioned in the slide 15 of lecture 3, the expected search time under a policy π when the oddball being 0 and the distractor being 1 is lower bounded by the following inequality :

$$E_{\pi}^{0}[\tau] \geq \frac{D(P_{\pi}^{0}||P_{\pi}^{1})}{\max_{\lambda} \sum_{a=1}^{K} \lambda_a D(p_a^0||p_a^1)}$$

Essentially to minimize the expected search time the denominator needs to be maximized which in turn suggests that we need to find the vector $\lambda = \{\lambda_a : 1 \leq a \leq K\}$. Here K is the number of actions the subject can take. In this particular problem, the subject can look at any of the six places in the image, hence $K = 6$.

Given the adversary places the oddball at different locations each time (i.e. the placement of oddball among the distractors is not purely random), and the subject can choose where to look in a controlled manner, it is always beneficial to look at a different location from those already visited. In the case, when the subject look at a position independent of what places he has already looked, $\lambda_a = \frac{1}{6}, \quad a \in \{1, 2, 3, 4, 5, 6\}$.

But when the searcher has control and the adversary opt for the previously mentioned strategy, $\lambda = \{\frac{1}{6}, \frac{1}{5}, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, 1\}$. This way the denominator of the inequality increases and the expected search time gets reduced. Therefore, it is a better strategy to remember the previously visited places and take action based on that instead of looking randomly independent of earlier locations.