Final Project Report

Carmel Pe'er Orion Yang

July 27, 2023

For a discrete random variable X if the distribution $p(x)$ is known, the informational entropy is defined by

$$H(X) = -\sum_x p(x)\log(p(x)).$$

Essentially, the entropy is an average of how surprising a particular value of X is based on the distribution $p(x)$. [1] However, the entropy of systems where $p(x)$ is unknown is more difficult to calculate. In such cases, it has been shown that if one divides a 1D dataset $x_{ii=1\ldots n}$ into two segments, a sample of length N and a dictionary of length M, and starting from random positions find the longest subsegments in the sample which match subsegments in the dictionary. We call this algorithm the pattern-matching (PM) algorithm.

Eventually, taking the average length of each of these segments $\langle l\rangle$, the entropy can be calculated with the following equation:

$$H \approx \frac{\log_2 M}{\langle l\rangle}$$

The inspiration for the PM algorithm derives from Lempel-Ziv 77 Factorization (LZ77), a data compression algorithm. One can show that LZ77 is asymptotically optimal, as in, the length of the compressed file gives the entropy. [2, 3] Based on this knowledge, one can prove that the PM algorithm and formula converge asymptotically to the entropy of a system. The difference between

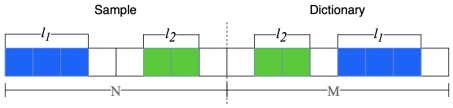
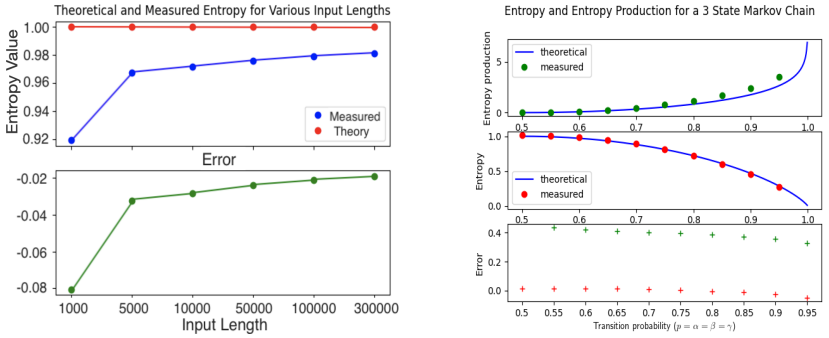


Figure 1: The input series is split into two segments: a sample of length N and a dictionary of length M. A window of size one is randomly placed within the sample, and it grows until a match in the dictionary can no longer be found. The length of the longest match for this window is recorded as l_1 and added to the list l The process is repeated multiple times to find other longest matches, denoted as l_2, etc., which are also included in the list l.

the PM algorithm and LZ77 is that the latter is full compression while the former only needs to accumulate enough lengths for an accurate estimate of the mean. Furthermore, the PM algorithm can be adjusted by simply reversing the dictionary to calculate the cross entropy of the system and its reverse. Then, by subtracting the entropy from the cross entropy, one obtains the Kullback–Leibler divergence (KLD) of a system and its reverse. The KLD can be used as a measure of the time reversal symmetry of the data which reveals the presence of any forcing in the system. We also call the KLD the entropy production when the distribution of the dictionary, q, is the reverse of p. Finally, note that this algorithm can be applied to data in higher dimensions but due to time constraints, our project focused only on 1D cases.

This summer, we implemented the PM algorithm in four different ways and applied it to neuron data. Two of the implementations were for discrete data and the other two were for continuous data. In each case, one of the implementations used calculations only in real space while the other used calculations in the Fourier space. Eventually, we used the discrete implementations to analyze spike train data from the first and second visual cortices (V1, V2) of various animals. Spike trains are data recorded based on neuron activation in response to stimuli, which are often transformed into discrete binary time series. [4] Through our analysis, we found a positive linear relationship between the firing rate of a neuron and its entropy production.



(a) (b)

Figure 2: Plots of theoretical and measured entropy for binary Bernoulli sequences and simple 3-state Markov Chains. Figure 2a shows theoretical entropy, measured entropy, and error calculated for increasing sequence length. Figure 2b shows theoretical entropy, measured entropy, theoretical entropy production, and measured entropy production and error calculated for increasing bias in underlying distribution of Markov sequence We see strong agreement between the theoretical and measured quantities.

The first implementation we built was the most simple, acting on discrete data in real-space. We tested it on various sequences with known entropy such as Bernoulli, Poisson, and a simple 3-state Markov Chains. [5] Figure 2 shows

two plots comparing the calculated and theoretical entropy of Bernoulli and Markov. In Figure 2a, as the length of the sequence increases, the error decreases asymptotically to 0. This is expected because the bound the entropy estimator is asymptotically exact at infinite length sequences. In Figure 2b, we see that the measurement for entropy production also matches theoretical calculations [5]. The finite-size error increases as the distribution becomes more biased because the bias makes it less likely that the matching segments are exemplary of the entire sequence.

The next implementation used calculations in the Fourier space to find matches. Rather than iterate through the dictionary to find an exact match we can transform the dictionary and the sample to Fourier space with a Fast Fourier Transform (FFT) and compute the cross correlation by simply multiplying the sample by the conjugate of the dictionary. Once normalized, the cross correlation provides a quality value between 0 and 1 which can be compared to some threshold to find matches. Initially we found little sensitivity between the threshold pick and final entropy value but in future work we hope to verify this observation formally. While using a FFT to find matches is more expensive in the 1D discrete case, this implementation was useful practice for cases where using a FFT is faster or even necessary such as continuous cases or cases in higher dimensions.

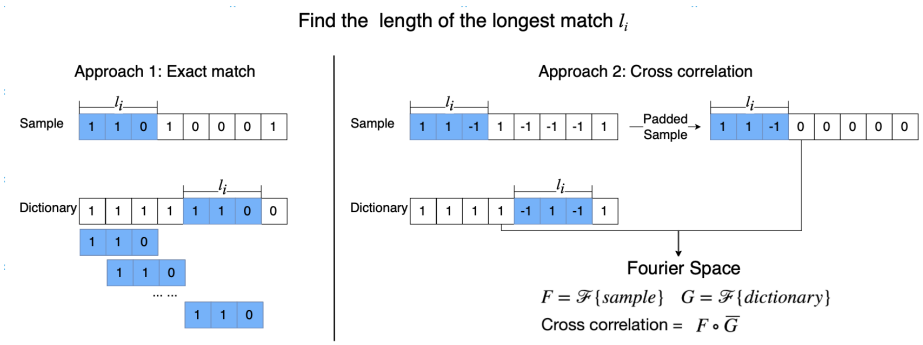
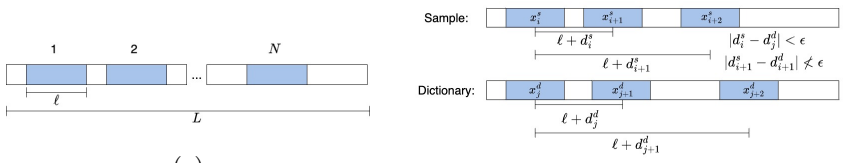


Figure 3: The real space PM algorithm searches for exact matches by exhaustively placing the sample segment in all positions within the dictionary. In contrast, FFT efficiently computes cross-correlations between the sample segment l_i and the dictionary in Fourier space.

For the continuous cases, the data we tested on was systems of N hard rods of constant length ℓ living on some larger length L. These systems are characterized by some density $\phi = \frac{N\ell}{L}$. A diagram of such a system is included in Figure 4a. These systems were of interest because their entropy is known analytically. While not generally possible, a real space version of the PM algorithm for hard rods. By intentionally picking the centers of rods as starting points, we know that the only mismatch between the sample segment and dictionary segment could be at the start of another rod. Using some error tolerance, ϵ, we



(a) (b)

Figure 4: Figure 4a shows an example of an L length hard rod system with N rods of length ℓ. Figure 4b shows a diagram of real space pattern matching on continuous hard rods. Starting from center x_i^s in the sample, we find that the distance between x_i^s and x_{i+1}^s is close enough to the distance between some x_j^d and x_{j+1}^d in the dictionary. We try to grow the match to the next center $x^s i + 2$ but find that it is not close enough to the difference between x_j^d and x_{j+2}^d.

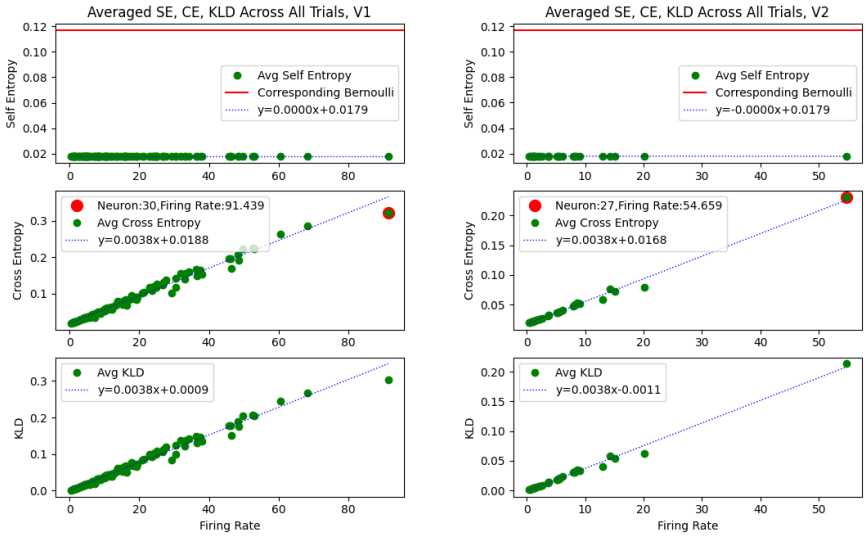
matched segments based on distances between rods. We then built an implementation using the a non-uniform fast Fourier transform (NUFFT) that can be more generally applied to continuous data. Due to time constraints, we were unable to fully study this implementation.

We investigate the time reversibility of neuron activity using spike trains from V1 and V2 neurons of anesthetized macaques exposed to a grating stimulus [4]. Time reversal symmetry is quantified using the KLD which is calculated by subtracting the self entropy of spike train from the cross-entropy between spike trains and their reversed version. To explore if reversibility is correlated with neuron activity (related to firing rate) or not, we plotted KLD vs. firing rate for all the neurons in different areas. Firing rate is defined as

$$v_k = \frac{n_k^{sp}}{T},$$

where n_k^{sp} is the spike count within a time window T , representing average spike frequency during trial k. Neurons are assumed to behave similarly under different stimuli. We compute average KLD separately for V1 and V2 neurons across all trials and observe a positive linear relationship between KLD and firing rate. This relationship holds when KLD is averaged across the same stimulus type. These findings suggests that higher firing rates lead to less reversible dynamics, indicating stronger correlations among the more frequent spikes, a fact hitherto unreported in neuroscience literature.

Future work on this project includes verifying our observations from the neural data and checking if a relationship between firing rate and entropy production is present in any other animals. We have already begun this step and have already found similar relationships. Additionally, we are curious about whether this relationship is also observable in any other areas of the brain such as deeper visual cortices. We would also like to continue work on the continuous implementations of the algorithm in both real and Fourier space. Given the time frame of this project, we were unable to analyze these to the same



(a) Neural Activity in V1

(b) Neural Activity in V2

Figure 5: The red line corresponds to the entropy of a Bernoulli distribution with the same proportion of 0s and 1s as the spike train. As anticipated, neurons exhibit lower self-entropy compared to Bernoulli distributions because the latter assumes independence between symbols, representing the most random scenario. The neurons' lower self-entropy implies stronger correlations in the spike train data during the experimental period.

extent as the discrete implementations. More work to verify their accuracy and parameter sensitivity is an important future step. Eventually, this algorithm should be implemented and studied in higher dimensions.

References

[1] J. V. Stone, *Information Theory: A Tutorial Introduction*. Sheffield, UK: Sebtel Press, first edition ed., 2015.

[2] J. Ziv and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Transactions on Information Theory*, vol. 24, pp. 530–536, Sept. 1978.

[3] S. Martiniani, "The other side of entropy," Nov. 2022.

[4] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal dynamics: from single neurons to networks and models of cognition*. Cambridge, United Kingdom: Cambridge University Press, 2014.

[5] S. Ro, B. Guo, A. Shih, T. V. Phan, R. H. Austin, D. Levine, P. M. Chaikin, and S. Martiniani, "Model-free measurement of local entropy production and extractable work in active matter," *Phys. Rev. Lett.*, vol. 129, p. 220601, Nov 2022.

Appendix

The proof behind the PM algorithm shows that given a list of maximum length matching segments, the probability of each length should follow a Gompertz (or truncated Gumbel) distribution. Throughout the summer we used this fact to investigate the performance of our implementations. In Figure 6 we see strong

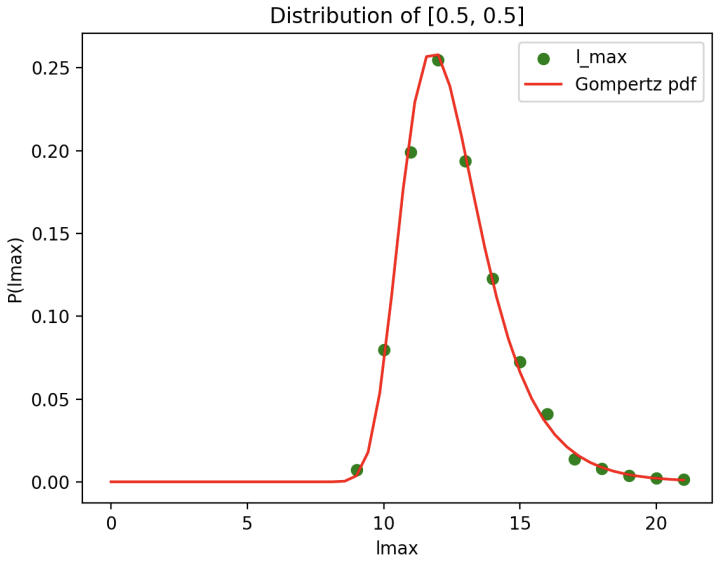


Figure 6: Probability of each maximum length measured by PM algorithm on a [0.5, 0.5] Bernoulli sequence. The probabilities almost perfectly match up with the Gompertz PDF.

agreement between the calculated probabilities and those from the Gomperts PDF. This aligns with the low error observed between the PM calculated entropy and the theoretical entropy of a Bernoulli sequenced governed by a [0.5, 0.5] distribution.

Another time we utilized these distributions was when working on the continuous implementations. One challenge was that with computers there is still

some level of discretization. Even if this discretization is very fine, it means that the exact value calculated by the PM algorithm will be inaccurate. However, we can still observe the curve of calculated entropy by ϕ and compare that to the shape of the theoretical entropy by ϕ. Comparing these curves can be difficult because the PM algorithm performs differently for different ϕ. We were able to observe this by plotting the Kernel Density Estimator of the list of matching lengths found and a truncated Gumbel distribution with the same mean and variance.

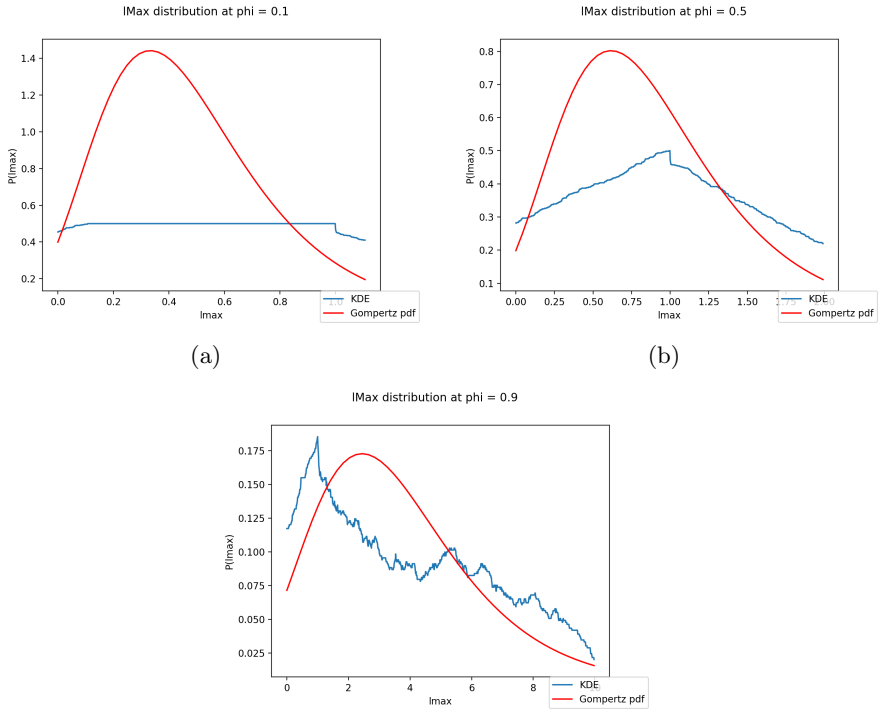


Figure 7: These show the various maximum length segment distributions compared to a Gompertz pdf with the same mean and variance. In theory, if the algorithm is unbiased, the distributions should match. We see that they are more similar at higher phi but do not match exactly.

Based on the graphs in Figure 7 we are able to observe that the algorithm likely performs better at higher phi but is nowhere near agreement at any phi. There are several possible sources of error. As previously stated, the PM estimate is asymptotically exact at infinite length sequence so finite N and finite number of nodes used for the NUFFT introduces some error. Additionally, because this version of the continuous implementation starts all matches at rod centers, there is some bias due to correlations between starting points. Future

work with the Fourier space continuous implementation seeks to minimize the impact of the finite effects and correlation bias.