Spectral Clustering for Identifying Large-Scale Neuron Connectivity Project Pre-Proposal

Jackson Cornell [1]

Abstract

Brain-Computer Interfaces (BCI) are becoming increasingly widespread and complex. Algorithms must keep up with this increasing complexity and be able to process and identify patterns in large sets of data being streamed by thousands of electrodes. In this paper, I model the data as a large graph, and apply spectral clustering algorithms to identify connectivity between neurons of the brain.

1. Introduction

1.1. About Me

My name is Jackson Cornell and I'm a first year Masters student at the University of Florida (UF), studying in the Electrical & Computer Engineering Department. My area of focus is in statistical signal processing and machine learning, and how they can be used to solve a wide range of applications such as satellite communications, finance, and brain-computer interfaces. This project will focus on the latter, namely, how clustering machine learning algorithms can be applied to neuronal measurements being modelled as a graph.

1.2. Papers

The following is a list of papers I plan to use to assist me on developing the algorithms and results:

- ***Identifying Functional Connectivity in Large-Scale Neural Ensemble Recordings: A Multiscale Data Mining Approach*** Will be the main reference, I will use the data from this experiment as well as the general experimental design. (Eldawlaty et al., 2009)
- ***On Spectral Clustering: Analysis and an Algorithm*** Will be the main spectral clustering algorithm I will implement. (Ng et al., 2002)
- ***Spectral Graph Theory*** Will be used as a reference for representing graphs in a matrix form and that can be processed by machine learning algorithms. (Butler & Chung, 2006)
- ***A Tutorial on Spectral Clustering*** Will be the main reference on how to specifically cluster on graphs and how to evaluate algorithm performance. (Luxburg, 2007)
- ***On the Use of Dynamic Bayesian Networks in Reconstructing Functional Neuronal Networks from Spike Train Ensembles*** Will be an additional reference for modelling neuron connectivity with probabilistic graphs. (Eldawlaty et al., 2010)
- ***CaImAn an open source tool for scalable calcium imaging data analysis*** Will be a reference on how to process the data set. (Giovannucci et al., 2019)

2. Background

The brain is an amazingly complex structure, which means mapping it is an incredibly arduous task. One method for doing so is by detecting the flow of neuronal spiking, we can map neuron connectivity associated with a certain stimulus or task. By modeling each neuron as a node in a graph, and each neuron connection as a vertex, a network of neurons can be mapped from detected spikes. The graph can be subsequently processed by using a technique know as Spectral Clustering to cluster closely-linked neurons, and thus create a map of which groups of neurons fire based on which stimulus or task. The application of this methodology with the given calcium imaging data will be a novel approach.

2.1. Detecting Spiking from Calcium Imaging

I will be working with Dr. Oweiss from UF's Department of Electrical & Computer Engineering to obtain and process the calcium imaging data. Calcium imaging is done using fluorescence microscopy to capture the density of calcium ions as a neuron fires. I will be using the CalmAn software to process the calcium imaging data and convert it into a 2D grid of neuron activity (where the values in the grid will change over time). From here, it is only a matter of representing each grid point (that is, a neuron) as a spike train. (Giovannucci et al., 2019) (Eldawlaty et al., 2009)

2.2. Modelling Neurons as Bayesian Graph Networks

After representing the the P neurons as a spike train, which we will denote $s_p(t)$, the challenge then becomes transforming this high-dimensional and time-varying data as a graph that spectral clustering can be performed on. To do this, we will first perform a wavelet decomposition of $s_p(t)$, which effectively results in each neuron spike train being decomposed into J separate timescales s_p^j. each entry of s_p^j can be thought of as the probability of neuron p firing at time j, effectively giving a probabilistic representation of the data.

Next, it is important to decompose this large-dimensional data into a $P \times P$ similarity matrix W that represents the interactions of neurons. First, a $P \times P$ correlation matrix Σ^j is computed for each timescale. Σ^j is then compressed into a P^2x1 vector, to form a $P^2 \times J$ matrix R. Singular Value Decomposition (SVD) is then applied to R, and the first Q dominant modes of the SVD are kept. W is then constructed by summing the Q $P \times P$ matrices, weighted by their corresponding eigenvalue, thus resulting in a $P \times P$ matrix that can be interpreted as a graph of the neuronal connections. (Eldawlatly et al., 2009) (Eldawlatly et al., 2010)

2.3. Spectral Clustering

The theory behind spectral clustering is that the eigenvalues (otherwise known as the spectrum) of a given graph reveals underlying structures. We are given a the weighted adjacency matrix W, where each row/column represents a vertex, and each entry in the matrix represents the weighted edge between the vertices. Additionally, the degree matrix D is a diagonal matrix where each entry along the diagonal is simply the number of edges connected to the corresponding vertex. Using W and D, we can construct the Laplacian matrix $L = D - W$, whose eigenvalues holds the spectral information of the graph. (Luxburg, 2007) (Butler & Chung, 2006)

There are two main categories of spectral clustering algorithms used in the literature: those who consist of traditional k-means or kNN clustering schemes, and those who consist of minimum cut clustering schemes. We will focus on the former group, of which nearly all algorithms consist of the same basic procedure. A matrix U is constructed from the eigenvectors of the Laplacian matrix L. The rows y_i of the matrix U are then used as the data points for which a clustering algorithm is applied to. The procedure for the algorithm I will be implementing, Normalized Spectral Clustering, is as follows:

(Ng et al., 2002)

Algorithm 1 Normalized Spectral Clustering

Input: weighted matrix W of size $P \times P$ and number of clusters K
Construct edge matrix D from W
Construct symmetric Laplacian matrix L
Compute $U \in \mathbb{R}^{P \times K}$, where U is the first K eigenvectors of L $u_1, ..., u_K$
Construct $T \in \mathbb{R}^{P \times K}$ where each member in T is defined as $t_{ij} = \dfrac{u_{ij}}{(\Sigma_k u_{ik}^2)^{1/2}}$
Let $y_i \in \mathbb{R}^K$ be a vector corresponding to the i^{th} row of T for $i = 1, ..., P$
Apply K-means clustering where y_i is the input data

2.4. Evaluation

To evaluate the effectiveness of the algorithm, I will first follow the procedures outlined in 8.1 in (Eldawlatly et al., 2009) to simulate coupled neurons. Algorithm 1 will then be applied to the simulated data, and its output clusters evaluated. If the clusters correspond to the coupled neurons, we know the algorithm is working as expected. The clustering parameter K will be chosen based off the eigen gap heuristic, wherein the first K values of λ should be small, but λ_{K+1} should be large (Luxburg, 2007). Adjusted rand index will be used to evaluate the accuracy of the resulting model.

References

Butler, S. and Chung, F. *Handbook of Linear Algebra*. CCR Press, 2nd edition, 2006.

Eldawlatly, S., Jin, R., and Oweiss, K. G. Identifying functional connectivity in large-scale neural ensemble recordings: A multiscale data mining approach. In *Neural Computation 21*, pp. 450–477, 2009.

Eldawlatly, S., Zhou, Y., and Oweiss, K. G. On the use of dynamic bayesian networks in reconstructing functional neuronal networks from spike train ensembles. In *Neural Computation 22*, pp. 158–189, 2010.

Giovannucci, A., Friedrich, J., Gunn, P., Kalfon, J., Brown, J., Koay, S., Taxidis, J., Najafi, F., Gauthier, J., Zhou, P., Khakh, B., Tank, D., Chklovskii, D., and Pnevmatikakis, E. Caiman an open source tool for scalable calcium imaging data analysis. *eLife*, 2019.

Luxburg, U. A tutorial on spectral clustering. *Statistics and Computing*, 17(4), 2007.

Ng, A., Jordan, M., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing 14*, pp. 849–856, 2002.