

An approach for analyzing spike train data
using dimensionality reduction

A Thesis
SUBMITTED TO THE FACULTY OF
THE
UNIVERSITY OF MINNESOTA
BY

Mary Sylvia Ebai Tambe

IN PARTIAL FULFILLMENT OF THE
REQUIERMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

Duane Nykamp (Advisor)

September 2018

© Mary Sylvia Ebai Tambe 2018

Acknowledgements

First, I would like to thank my thesis advisor Prof. Duane Nykamp for his guidance, invaluable advise and constructive criticism while writing this paper and throughout my academic journey. The door to Prof. Nykamp's office was always open whenever I ran into a trouble spot or had a question about my research or writing. He allowed this paper to be my own work, but steered me in the right the direction whenever he thought I needed it.

My very special thanks to the experts who were involved in the examination and evaluation for this research project: Prof. Richard McGehee and Prof. Galin Jones. Without their passionate participation and input, my plan A master's thesis could not have been completed successfully.

I would like to acknowledge Madeline Handschy as the second reader of this thesis, and I am gratefully indebted to her for her very valuable comments on this thesis.

Finally, I must express my very profound gratitude to my beloved husband, Dr. Norbert Tambe Ebai for providing me with unfailing love and continuous encouragement throughout my years in graduate school, and my precious children Kathleen and Neilton, who are the pride and joy of my life. This accomplishment would not have been possible without them. Thanks to my family and friends for encouraging me to follow my dreams. Special thanks to my father, John Mark Otim, who encouraged me to pursue farther studies and who has been by my side through thick and thin. Last but not least, words cannot express my deep gratitude for my Guardians Laveeda and Lynn Battle for their unwavering support and for encouraging me to believe in my self.

Dedicated to my beloved parents, Ms. Rose Asimwe and Mr. John Mark Otim

Abstract

Our perception of the world is influenced by the way our brains process information received from millions of neurons. Our senses are based on information sent to the brain in the form of sequences of stereotyped electrical impulses, or spikes. We attempt to answer a central question: how do we understand and analyze neural responses when their relationship to external variables such as stimuli, location or behavior is unclear?

Our main innovation is that we first ignore these external variables and instead look for structure solely within data representing neural activity such as spike trains. Our preliminary results on synthetic data show how the diffusion maps algorithm applied to data preprocessed in a novel way, captures the one-dimensional manifold corresponding to a simulated rat's movement around a track. Diffusion maps reveals the structure of a one-dimensional manifold within the neural activity, as would be expected from the fact that the neural activity is strongly correlated with the rat's position.

Contents

Acknowledgement	i
Dedication	ii
Abstract	iii
1 Introduction	1
2 Biological background	2
2.1 Structure of a neuron	2
2.2 Membrane potential	2
2.3 Generation of an action potential	3
2.4 Measuring neurons	3
2.5 Place cells and place fields	4
2.6 Spike trains	4
2.7 Raster plot	4
2.8 Poisson processes	6
3 Previous work	6
3.1 Single-unit multiple trial experimental design	6
3.2 Multiple single-unit single trial experimental design	7
3.3 Dimensionality reduction	7
3.3.1 Overview of spectral dimensionality reduction	8
3.3.2 Linear dimensionality reduction	8
3.3.3 Non-linear dimensionality reduction	9
3.3.4 Multidimensional scaling (MDS)	9
3.3.5 Spectral non-linear dimensionality reduction	10
3.3.6 Key graph theory concepts	10
3.3.7 Neighborhood graph	10
3.3.8 Basic idea of the diffusion maps algorithm	11
3.4 Spike train metrics	12
3.4.1 Cost-based spike metrics	12
3.4.2 Van Rossum spike metrics	13
4 Current Project	14
4.1 Our framework	14

4.2	Nature of spike data	15
4.3	The previous time function	15
4.4	Simulation	16
4.5	Analysis	17
4.6	Distance and similarity measure used in results section	18
5	Results and interpretation	18
5.1	Interpretation of our results	19
6	Future direction	19
	References	22
	Appendices	24
A	Linear algebra review	24
A.1	Definitions	24
A.2	Properties of SVD	24
B	Proximity measures	24
C	Graph Laplacians and their properties	25
D	Random walk on a graph	26
E	Diffusion maps algorithm	26

1 Introduction

Our perception of the world is influenced by the way our brains process information received from millions of neurons. Our senses are based on information sent to the brain in the form of sequences of stereotyped electrical impulses, or spikes. Thus spike trains are considered as the main mode of information transmission in the nervous system. Sometimes what neurons encode may be obvious, for instance when a neuron directly responds to a stimulus. However in general, the way neurons encode information is not known. We attempt to answer a central question: how do we understand and analyze neural responses when their relationship to external variables such as stimuli, location or behavior is unclear?

Our main innovation is that we first ignore these external variables and instead look for structure solely within data representing neural activity such as spike trains. Traditional approaches base analyses from the onset on the assumption of a particular relationship between the neural activity and some external variable. For example, when analyzing the response of visual neurons, one may start out by investigating the dependence of the activity on different visual stimuli. We avoid such apriori assumptions on the relationship between neural responses and any external variable. Once we have discovered a particular structure in the neural activity, we can then compare the structure from data to external variables.

Our primary tool for discovering structure within neural activity data is dimensionality reduction. Dimensionality reduction considers the problem of transforming a high-dimensional data set into a new low-dimensional data set in such a way that preserves, as much as possible, the underlying geometry of the data. In our analysis, we use a non-linear dimensionality reduction technique called diffusion maps which finds a low-dimensional model of the data by preserving relative distances between neighboring points on a data manifold. The discovered low-dimensional manifold can then be related to external variables in order to explore possible relationships between the neural activity and the external variables.

Our preliminary results on synthetic data show how the diffusion maps algorithm, applied to data preprocessed in a novel way, captures the one-dimensional manifold corresponding to a simulated rat's movement around a track. Diffusion maps reveals the structure of a one-dimensional manifold within the neural activity as would be expected from the fact that the neural activity is strongly correlated with the rat's position.

The paper is organized as follows: In section 2, we give some biological background on the structure of neurons, how they are measured and the nature of spikes or electrical signals transferred among neurons within the brain. In section 3, we describe some previous work such as classical and modern analyses used to model neural responses and outline their strengths and weaknesses, if any. In section 4, we give an overview of our current project, simulations and analyses used to model spike time data. We also introduce a novel method of preprocessing spike time data so as to discover structure in the data. We use dimensionality reduction to analyze our preprocessed data. In section 5, we report our results on synthetic data as a first step towards analysing real-world spike train data. Our preliminary results on synthetic data show that diffusion maps using spike time data preprocessed by our novel previous time measure, captures the one-dimensional manifold corresponding to a simulated rat's movement around a track. In section 6 we give a brief discussion of our results and outline our future directions.

2 Biological background

2.1 Structure of a neuron

A neuron is a specialized cell in the nervous system that receives, represents, and transmits information through a series of electrical pulses called action potentials or spikes. The neuron (see Figure 1) is the fundamental unit of brain function and is made of three major parts: the dendrites (which receive information from other neurons), the cell body or soma (which processes information) and the axon (which transmits information to other neurons)

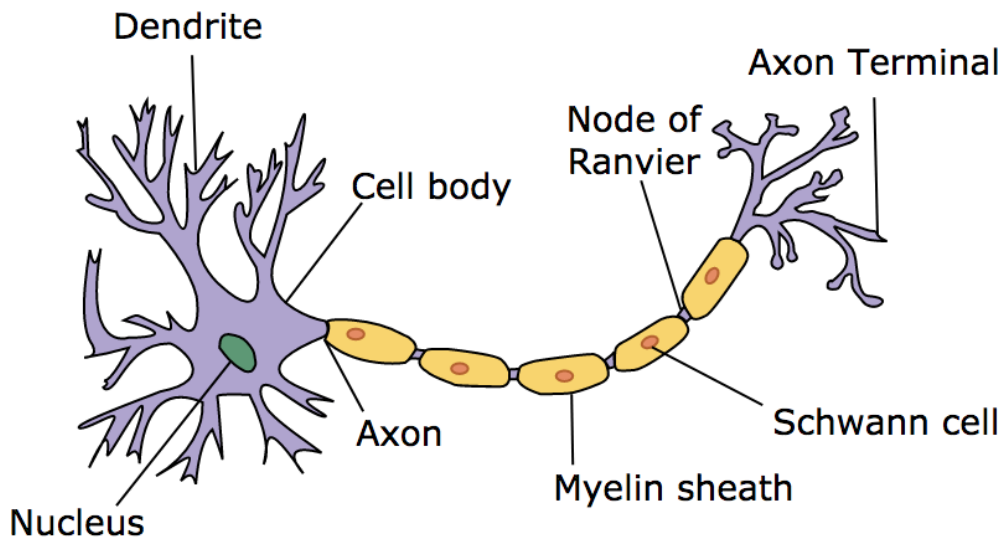


Figure 1: Structure of a neuron.

Taken from: <https://commons.wikimedia.org/w/index.php?curid=1474927>

The cell membrane is made up of phospholipids (fat) and separates the cell interior from the extracellular space. The lipid cell membrane is impermeable to charged ions but thin enough to allow interaction of separated charged ions through electrostatic forces. Thus the cell membrane acts as an electrical capacitor. Embedded in the cell membrane are Na^+ (sodium) and K^+ (potassium) ion exchange pumps which pump out three Na^+ ions for every two K^+ ions pumped in. As a result, Na^+ is more concentrated outside the cell than inside it, and the intracellular concentration of K^+ is substantially higher than that outside the cell. There are gated ion channels (trans-membrane proteins) embedded in the cell membrane which open or close enabling predominantly K^+ , Na^+ , Ca^{2+} (calcium), and Cl^- (chloride) ions to flow into and out of the cell. The ion channels act as conductors.

2.2 Membrane potential

A potential is a distribution of charge across the cell membrane. *Voltage* is a measure of the potential energy generated by separated charges and is measured in millivolts (mV). Ions flow into and out of the cell due to both voltage and concentration gradients. *Current* refers to the flow of charged ions into and out of the cell. A resting neuron contains a greater

number of negative charges on the inside than on the outside. This difference in separated charges is called the neuron's *membrane potential*. A neuron with a membrane potential of approximately -70mV is called *polarized*. This number is also referred to as the resting membrane potential.

2.3 Generation of an action potential

Dendrites contain chemically-gated ion channels, which open when a stimulus affects a sensory receptor, such as neurotransmitters binding to the dendrite receptors. As a result, current flows into the intracellular fluid, causing the membrane potential to be less negative or to be positive (depolarization of the neuron). This increase in the membrane potential causes the voltage-gated Na^+ channels, at the entry point of the axon, to open and thus more Na^+ flows into the cell down its electrochemical gradient. When a certain threshold is reached ($\approx -55\text{mV}$), an electrical pulse lasting a short duration ($\approx 1\text{ms}$), called an *action potential*, is released and is propagated over long distances along the neuron's axon. When the membrane potential rises, the voltage-gated K^+ channels open, allowing more K^+ to flow out of the cell, which causes the membrane potential to fall below the resting potential (hyperpolarization of the neuron). The neuron later returns to its resting potential after a refractory period, during which the likelihood of spiking is greatly reduced.

The axon terminal contains voltage-gated Ca^{2+} channels which open in response to an action potential, causing an influx of Ca^{2+} . This leads to the release of neurotransmitters (stored in the synaptic vesicles) into the synaptic cleft. The neurotransmitters then bind to the dendrite receptors of nearby neurons. A synapse is a specialized structure that facilitates communication between neurons. The neuron that sends off an action potential is called *presynaptic* and the one receiving the chemical message is called a *postsynaptic* neuron. Depending on the chemical properties of the binding neurotransmitters, action potentials fall into two broad categories. Excitatory post synaptic potentials (EPSP) result from excitation of a postsynaptic neuron, while inhibitory post synaptic potentials (IPSP) result from inhibition.

2.4 Measuring neurons

The electrical properties of biological cells are measured using electrodes, which allow electrical current to pass through them when they come into contact with electrolytes. Due to the small size of cells, microelectrodes are typically used to measure single unit spiking activity. A *single unit* refers to a single action potential-generating neuron, whose spikes are clearly isolated by a recording microelectrode [1]. Neurons are measured using two methods: *extracellular recording* in which an electrode is inserted in the extracellular space near the cell body and *intracellular recording*, a process where an electrode is inserted inside the cell body (see Figure 2). Intracellular electrodes record membrane potential by comparing the potential on the inserted electrode to that of a reference electrode placed in the extracellular fluid surrounding the cell body. Extracellular recordings can be processed (via spike sorting algorithms) to obtain spike times.

Multi-electrode arrays enable simultaneous single-unit recordings from multiple brain sites. The data set we analyze is based on extracellular multiple single-unit single-trial recordings in which spike times of single units have been isolated using suitable spike sorting algorithms.

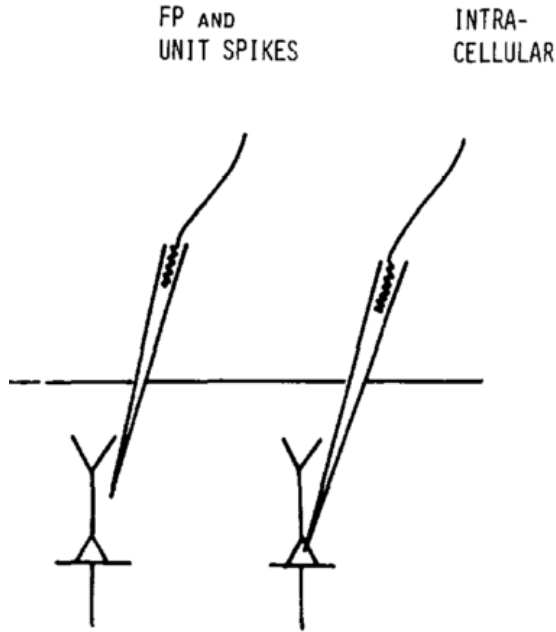


Figure 2: Extracellular (left) and intracellular (right) recordings, adapted from figure 2 of [1].

2.5 Place cells and place fields

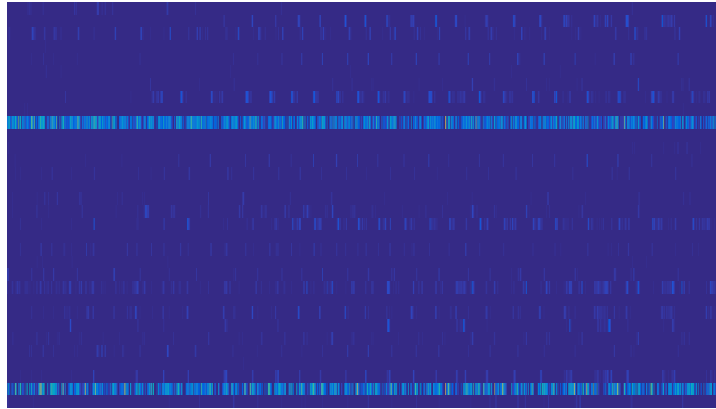
Place cells are neurons found in the CA1 and CA3 region of the rat hippocampus, whose firing rate is strongly modulated by the position of the rat within its environment. Hence place cells appear to be encoding location [2, 3]. The region where a place cell is highly likely to spike is called a place field.

2.6 Spike trains

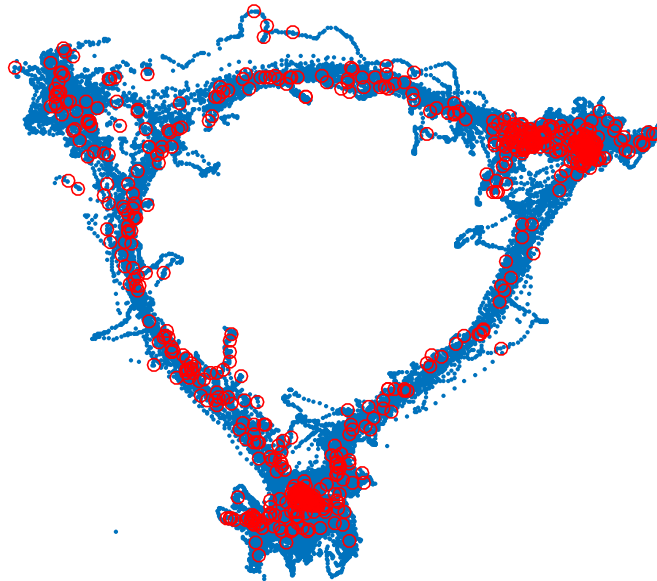
Experimentalists obtain information that neurons carry about an organism's environment by measuring from neurons. Spike trains are considered as the main mode of information transmission in the nervous system. A spike train is an ordered sequence of recorded times at which a neuron fires an action potential (spike) [4].

2.7 Raster plot

A raster plot is a graphical representation of spike trains. In this graph, a short vertical line is used to show the time at which a spike occurred in a recorded voltage trace. Figure 3(a) shows a raster plot of real-world data from the CA1 region of the rat hippocampus. The y-axis represents the neuron label while the x-axis represents spike times.



(a) Raster plot for 32 neurons from the CA1 region of the rat hippocampus



(b) Firing activity of 32 neurons whose raster plot is shown in (a). The location of each dot indicates the rat's position in space. The color indicates whether or not a single neuron fired.

Figure 3: Raster plot (a) and firing activity (b) of 32 neurons from the CA1 region of the rat hippocampus based on real-world data provided by the Redish Lab. The data was collected during a behavioral experiment in which the subject was running around a circular maze.

The goal of our analysis is to investigate the relationship between spike trains (neural activity) and other variables such as stimuli and location. It is known that place cell firing is closely correlated with the animal's location in its environment [2, 3, 5].

2.8 Poisson processes

We represent a spike train as a sequence of random events, that is, as a point process where the random events correspond to spike times. We focus on the simplest point process: a Poisson process, where the numbers of events in non-overlapping intervals are independent random variables. If the probability per unit time (the instantaneous event rate r) is constant, the Poisson process is called a *homogeneous Poisson process*. On the other hand if the instantaneous firing rate varies with time, the Poisson process is called a *non-homogeneous Poisson process*.

3 Previous work

In this section, we review some approaches for analyzing neuronal responses. First, we review the classical single-unit multiple trial approach based on estimating the firing rate and highlight the strengths and weaknesses of this approach. Second, we review the dimensionality reduction approach as a means of addressing the problem of trial-to-trial variability of neural responses. Our approach is also dimensionality reduction. Finally, we review two spike train metrics that have been used to study dissimilarity measures between spike trains. We review spike train metrics because the dimensionality reduction algorithms we use require a distance matrix as an input.

3.1 Single-unit multiple trial experimental design

Traditionally, spike trains are obtained by recording from a single sensory neuron during multiple presentations of the same stimulus over a fixed period of time. The neural responses are decoded by counting the total number of spikes in a given time window following onset of the stimulus (firing rate) [6]. This approach is based on the rate-coding hypothesis which states that the rate of spike generation varies with stimulus intensity. The goal of these classical experiments is to answer the question: Given a set of stimulus features, what is the neural response (spike train) elicited by the presented stimulus? The neuron is then characterized by plotting the firing rate as a function of given stimulus feature values. This plot is referred to as a tuning curve of the cell. In modern single-unit multiple trial recordings, the mean firing rate is estimated by averaging neural responses across repeated trials.

Experimenters developed substantial scientific theories based on single-unit multiple trial recordings. For instance, it is known that activity patterns from sensory neurons in the motor cortex of primates are tuned to the direction of the subject's arm movements [7], that neurons in the visual cortex of primates are tuned to the orientation of a stimulus [8], that place cells in the CA1 region of the rat hippocampus are tuned to the animal's position in the environment [2]. However, methods of analyzing single-unit multiple trial recordings require trial averaging in order to estimate the firing rate. This often results in the smoothing over of rapid fluctuations in the responses, which may lead to loss of temporal information in spike trains, thus yielding incorrect interpretations of the underlying neural mechanisms. Moreover, neural responses vary substantially between trials even on presentation of the same stimulus due to many variables. This trial-to-trial variability makes single-unit multiple trial analyses hard. In addition, there are some internal neural mechanisms such as olfaction,

cognition and decision making that cannot be controlled by researchers, as can other forms of stimuli. [9, 10, 11, 12, 13]. Such observed phenomena therefore could instead be analyzed using single trial recordings from multiple single units. We refer to this alternative option as multiple single-unit single trial experimental design.

3.2 Multiple single-unit single trial experimental design

The use of multi-electrode [14] recording technologies has enabled multiple single-unit single trial recording from various brain structures. However, trying to identify all possible activity patterns corresponding to a single neuron within the recorded neural population leads to an amplification in the number of variables to be considered while modeling collective neural responses. Consequently, biologically motivated assumptions have to be made in order to model population activity. For instance, a common approach in analyzing population activity is to assume that the activity patterns of one neuron are independent of other neurons which simplifies analyses as in the single-unit experimental design. However, in the dynamical systems perspective, neurons belong to an underlying connected network within the brain which may lead to correlated responses between neurons [15]. In addition, among a population of neurons that encode features of a stimulus, the population activity is correlated with the features of the stimulus [7, 8]. This suggests that relaxing the independence assumption by taking into account correlations among population activity patterns may be useful in studying neuronal response variability. Lately, dimensionality reduction [16], has been suggested as a tool for studying collective neural activity patterns. The technique can be applied to multiple single-unit single trial recordings so as to obtain a low dimensional model of the data which captures similar activity patterns among neuronal populations.

In order to better analyze neural response variability, our approach is to use dimensionality reduction on synthetic data as a first step to analyzing a set of spike trains recorded from multiple neurons in the CA1 region of the rat hippocampus in a single trial. Our focus is to study collective neural activity patterns using a low dimensional model which captures similar activity patterns among neural populations. The low dimensional model should also enable us to approximately recover the location of the animal (rat) based on the fact that place cell firing is correlated with position. Moreover, reconstructing information about an organism’s environment based on neuronal responses enables the experimenter to interpret information in a way that an organism is likely to interpret it [17]. For instance, it may be possible to predict what the animal is “thinking about” at a particular moment in time based on our low dimensional model.

3.3 Dimensionality reduction

In this section, we review some linear and non-linear dimensionality reduction techniques. We focus on spectral dimensionality reduction methods where the low dimensional model is obtained from spectral decomposition of an $n \times n$ positive semidefinite (PSD) proximity matrix (see appendices A and B). We start by defining dimensionality reduction and outlining steps of spectral dimensionality reduction. Next, we give a brief description of PCA and the diffusion maps algorithm for dimensionality reduction. In later sections, we compare the performance of PCA and diffusion maps on our spike train data set.

3.3.1 Overview of spectral dimensionality reduction

Let $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ be a set of n data points, each of which is associated with N features, so that each \mathbf{x}_i is a point in a high dimensional space \mathbb{R}^N . For convenience, we often view the set of data points \mathbf{X} as a matrix, $\mathbf{X} \in \mathbb{R}^{n \times N}$. Assume that the data points lie on or near an underlying l -dimensional manifold embedded in the high dimensional space where l is much smaller than N . Dimensionality reduction considers the problem of transforming the high dimensional data set \mathbf{X} into a new data set $\mathbf{Y} = \{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ of n points, each of which is associated with a smaller set of l features, which are possibly new or a combination of the original N features, and such that each $\mathbf{y}_i \in \mathbb{R}^l$ is a low dimensional representation of $\mathbf{x}_i \in \mathbb{R}^N$. In addition the transformation must preserve, as much as possible, the underlying geometry of the data.

Spectral dimensionality reduction (SDR) refers to all dimensionality reduction methods which obtain a low dimensional model of \mathbf{X} by carrying out four main steps [18].

- i) A distance $b_{ij} = \text{dist}(\mathbf{x}_i, \mathbf{x}_j)$, between data points is chosen and a real number $l, 1 \leq l < N$ representing the desired dimensionality is fixed.
- ii) From the distance, an $n \times n$ PSD proximity matrix is computed (see appendix B).
- iii) Spectral decomposition is carried out on the generated PSD matrix and the top l eigenvectors $\{\mathbf{v}_1, \dots, \mathbf{v}_l\}$ of the proximity matrix form the columns in a new matrix, $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_l] \in \mathbb{R}^{n \times l}$.
- iv) Viewing the i^{th} row of \mathbf{V} as a point \mathbf{y}_i in \mathbb{R}^l , a new set of n points $\{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ is obtained such that each $\mathbf{y}_i = (\mathbf{v}_1(i), \dots, \mathbf{v}_l(i)) \in \mathbb{R}^l$ is a low dimensional representation of $\mathbf{x}_i \in \mathbb{R}^N$.

3.3.2 Linear dimensionality reduction

Let $\mathbf{X} \in \mathbb{R}^{n \times N}$ be a matrix of n data points in a high dimensional space \mathbb{R}^N . Linear dimensionality reduction refers to the problem of finding a low dimensional model using a linear transformation of the data. The low dimensional model of the data \mathbf{Y} , is obtained using the relation $\mathbf{Y} = \mathbf{M}^T \mathbf{X}$ where M is an $n \times l$ matrix with $l \ll N$. A common spectral linear approach is principle component analysis (PCA) [19]. PCA is a linear technique for finding the directions of maximum variance in the data. The main assumption in PCA is that the high dimensional data lie on or near a low l -dimensional linear subspace embedded in some high dimensional space \mathbb{R}^N . The low dimensional model that describes the data is found via spectral decomposition of the sample covariance matrix as follows:

- i) Let \mathbf{X} be the centered data formed by subtracting the mean from each point.
- ii) Summarize the correlation relationships between the zero-mean data points by computing the sample covariance matrix $\frac{1}{n} \mathbf{X} \mathbf{X}^T$.
- iii) Find the spectral decomposition of $\mathbf{X} \mathbf{X}^T$ or use SVD to find $\mathbf{X} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$ (see appendix A).
- iv) Let \mathbf{V}_l denote the first l columns of \mathbf{V} corresponding to the top l singular values of \mathbf{X} .

v) The low dimensional model \mathbf{Y} is obtained by setting $\mathbf{Y} = \mathbf{V}^\top \mathbf{X}$.

3.3.3 Non-linear dimensionality reduction

Methods like PCA assume that the underlying manifold on which the data lie is globally linear. This implies that the shortest distance between any two points while traveling along the data manifold is a straight line. However, any dimensionality reduction technique must preserve the underlying geometry of the data manifold. In particular, points that are close together in the high dimensional space should be mapped closer to each other in the low dimensional space whereas points that are far apart in the high dimensional space must remain far apart in the embedded space.

To accomplish this goal, we must ignore long distances in the embedded space, as long distances computed in the embedded space may be the result of shortcuts that ignore the non-linear geometry of the manifold. Figure 4 illustrates the limitations of linear techniques on highly curved manifolds like the swiss roll and helps to highlight the importance of non-linear dimensionality reduction techniques.

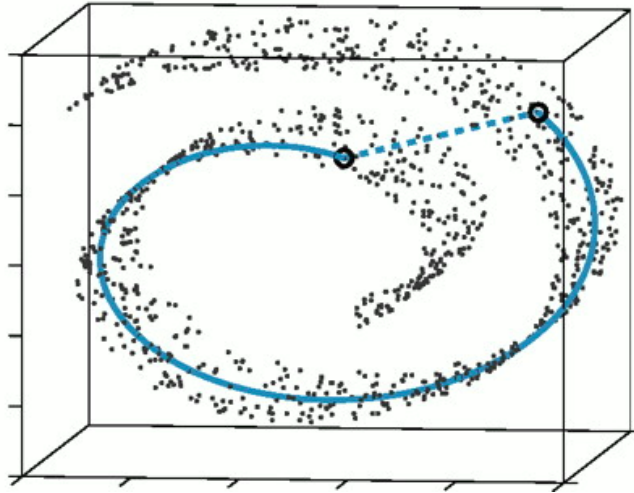


Figure 4: For a linear method like PCA, the two circled points on the swiss roll would be close together even though they are far apart when using geodesic distances. Taken from [20].

3.3.4 Multidimensional scaling (MDS)

This idea of preserving relative distances between points in both the input space and embedded space is based on the multidimensional scaling (MDS) approach. Assume that an $n \times n$ matrix, $\mathbf{B}=(b_{ij})$, of pair-wise distances (not necessarily Euclidean) or similarities (see appendix B), $\mathbf{S}=(s_{ij})$, between data points, $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, is given and n is large.

MDS [21, 22] considers the the problem of finding a low dimensional model of the high dimensional data by searching for a configuration of n points $\{\mathbf{y}_1, \dots, \mathbf{y}_n\}$ in $\mathbb{R}^l, l \ll n$, where each \mathbf{y}_i is a low dimensional representation of \mathbf{x}_i , and such that relative pair-wise distances between points are preserved. Specifically, the Euclidean distances between the

configuration points, $\|\mathbf{y}_i - \mathbf{y}_j\|_2$, must be as “close” as possible to the given distances, b_{ij} , that is, $\|\mathbf{y}_i - \mathbf{y}_j\|_2 \approx b_{ij}$, for all $1 \leq i, j \leq n$.

3.3.5 Spectral non-linear dimensionality reduction

Spectral non-linear dimensionality reduction (SNLDR) techniques are a class of MDS methods which find a low dimensional model of the data by preserving relative distances between neighboring points on a data manifold. These are the non-linear dimensionality reduction algorithms that we will consider. Examples of SNLDR methods include ISOMAP [20], locally linear embedding (LLE) [23], Laplacian eigenmaps (LE) [24] and diffusion maps [25], in addition to many others. Unlike linear dimensionality reduction techniques, SNLDR methods do not make a priori assumptions about the underlying geometry of the data manifold. SNLDR methods model local neighborhood relations between data points by building a graph on the data [26]. We only review the diffusion maps algorithm.

3.3.6 Key graph theory concepts

The non-linear dimensionality reduction algorithms we consider are based on graph theory. Hence, we first introduce three key graph theory concepts that we need: a weighted undirected graph, the degree matrix and the graph Laplacian.

A graph is a tuple $G = (V, E)$ where $V = \{v_1, \dots, v_n\}$ is a finite set of points called vertices or nodes and E is a finite collection of edges connecting pairs of vertices. The graph G , is undirected if the edges between vertices are bidirectional. We can make a weighted graph by assigning a weight or number w_{ij} , to an edge $e_{ij} \in E$, between a pair of vertices $(v_i, v_j) \in V$. A connectivity matrix or similarity matrix \mathbf{W} , of G , is the matrix whose (i, j) -th entries are edge weights w_{ij} . The degree d_i , is the sum of weights w_{ij} , of all edges connected to a vertex $v_i \in V$. A diagonal matrix \mathbf{D} , with degrees, d_i , on its diagonal is called the degree matrix. The unnormalized graph Laplacian is the matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$.

3.3.7 Neighborhood graph

We can approximate small distances between points in the input space using the usual Euclidean distance between nearby points in the embedded space as in linear approaches. However, long distances in the input space are not necessarily equal to long distances in the feature space (see figure 4). One way to address this problem is to ignore long distances in the embedded space by building a graph $G = (V, E)$ on the data. We ignore long distances by placing an edge $e_{ij} \in E$ only between neighboring vertices $(v_i, v_j) \in V$. The result is called a neighborhood graph.

A particular type of neighborhood graph is the ϵ -neighborhood graph constructed as follows. Suppose we are given a data set, $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$ and assume that the pairwise distances, $b_{ij} = \text{dist}(\mathbf{x}_i, \mathbf{x}_j)$, between points are known. Identifying each data point with a vertex $v_i \in V$, on a graph $G = (V, E)$, we can build a weighted undirected graph on the data by placing an edge $e_{ij} \in E$, between vertices $(v_i, v_j) \in V$, if x_i and x_j are sufficiently close i.e., $\text{dist}(x_i, x_j) < \epsilon$. We weight the edge $e_{ij} \in E$, by a similarity measure s_{ij} , derived from the given distances (see appendix B). The resulting ϵ -nearest neighborhood graph represents the local distances among points.

For diffusion maps, the neighborhood graph is modified to a fully connected graph, but where the weights are approximately zero whenever the distance between data points is large. To do this, we use a similarity function that captures local neighborhood relations between points and at the same time decays to zero so fast that we do not have to truncate the weights. In particular, in diffusion maps, we use the similarity function

$$w_{ij} = \exp\left\{-\frac{\text{dist}^2(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}\right\}.$$

In this way, σ plays the role of ϵ , in that it specifies the width of the neighborhood.

3.3.8 Basic idea of the diffusion maps algorithm

The diffusion maps algorithm consists of several steps which are outlined in appendix E. Below is a brief summary of the diffusion maps algorithm.

- 1) Given the pairwise distance $\text{dist}(\mathbf{x}_i, \mathbf{x}_j)$, compute the weight matrix $\mathbf{W} = (w_{ij})$, between data points by setting

$$w_{ij} = \exp\left\{-\frac{\text{dist}^2(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}\right\}.$$

- 2) Let $d_i = \sum_{j=1}^n w_{ij}$ be the degree of the i^{th} node and compute the degree matrix $\mathbf{D} = \text{diag}(d_1, \dots, d_n)$.
- 3) Define a random walk on the graph of the data (see appendix D) by specifying $p_{ij} = \frac{w_{ij}}{d_i}$, the probability of going from vertex v_i to v_j in one-step. Organize the pair-wise transition probabilities in an $n \times n$ transition probability matrix $\mathbf{P} = \mathbf{D}^{-1}\mathbf{W}$ with $\mathbf{P} = (p_{ij})$.
- 4) Normalize \mathbf{P} to obtain a PSD similarity matrix \mathbf{S} given by $\mathbf{S} = \mathbf{D}^{\frac{1}{2}}\mathbf{P}\mathbf{D}^{\frac{1}{2}}$.
- 5) Compute the SVD of \mathbf{S} and write $\mathbf{S} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^{\text{T}}$ where, $\mathbf{\Sigma} = \text{diag}(\lambda_0, \dots, \lambda_{n-1})$ with ordered eigenvalues, $\lambda_0 \geq \lambda_1 \geq \dots \geq \lambda_{n-1}$ and corresponding eigen vectors $\mathbf{V} = [v_0, v_1, \dots, v_{n-1}]$.
- 6) The l -dimensional embedding of the i^{th} data point \mathbf{x}_i in the low dimensional space \mathbb{R}^l is the map

$$\mathbf{x}_i \mapsto \begin{bmatrix} \lambda_1^t \phi_1(i) \\ \lambda_2^t \phi_2(i) \\ \vdots \\ \lambda_l^t \phi_l(i) \end{bmatrix}.$$

where $\{\phi_0, \phi_1, \dots, \phi_{n-1}\}$ are the right singular vectors of \mathbf{P} and λ_i^t , are obtained from the t -step transition probability matrix \mathbf{P}^t (see appendix E).

3.4 Spike train metrics

So far, the dimensionality reduction algorithms we have reviewed require a distance matrix as an input. Since our data consists of spike trains, we need a measure of quantifying how similar or dissimilar two spike trains are. A common approach for quantifying neuronal response variability is through specification of a similarity or dissimilarity measure between pairs of spike train data [27, 28, 29, 30, 31]. In line with conclusions reached by Victor and Purpura [28, 29], and van Rossum [30], a short distance between two spike trains approximately represents similar neuronal inputs while a large distance between spike trains roughly represents divergence between different neuronal inputs.

How spike trains encode information is not known. In certain instances, information may be encoded through the precise time at which spikes occur (temporal coding) where as in others, it is encoded through the number of spikes in a given interval (rate coding) [6]. As a remedy for loss of temporal information due to trial averaging, the observation duration is often divided into non-overlapping time intervals called bins. The shortcoming of spike binning in studying temporal patterns is that two different spike trains often yield identical binning patterns whenever spikes fall in the same bin.

Several spike train measures have been designed to overcome the problem of binning. For instance, the edit-length metric [28, 29] is based on minimizing the cost of transforming one spike train into another by deleting, inserting or shifting a spike. Another measure, the van Rossum distance [30, 31], refers to any metric induced on the space of spike trains by transforming a spike train into a continuous function using a smoothing kernel and then using the standard L^2 distance on the corresponding function space as the dissimilarity measure. An additional measure, the correlation-based distance [32] is based on filtering the spike trains using a Gaussian kernel and then using the normalized dot product between spike trains as a similarity measure. We only review the cost-based and Van Rossum spike train metrics. Both metrics allow one to control the sensitivity to differences in spike timing versus spike count.

3.4.1 Cost-based spike metrics

The Victor and Purpura (VP) metric [28, 29], is defined as the minimum cost of transforming one spike train into another, using a set of three elementary operations: deleting a spike, inserting a spike and moving or shifting a spike. Define a metric d between spike trains A and B as follows:

$$d^{VP}(A,B;q) = \min_{S(A,B)} \sum_k c_q(T_k, T_{k+1}). \quad (1)$$

Then the VP metric is the spike train metric in (1), together with the three elementary operations above. The quantity $c_q(T_k, T_{k+1})$ denotes the cost of transforming a spike train T_k to a spike train T_{k+1} . The minimum is taken over the set $S(A, B)$, of all possible sequences of spike trains $\{T_k\}$, obtained from elementary operations, that transform the spike train A to the spike train B . In the VP metric, the cost of inserting or deleting a spike is set to 1.

The non-negative quantity q denotes the cost (in 1/seconds) of shifting a spike, specifying the relative cost of the shifting operation compared to inserting or deleting. In this way, one can use q to specify the relative cost of discrepancies in spike count versus spike timing. The effect of q on the relative importance of spike timing is illustrated by a simple example

of two spike trains $A_1 = \{t^A\}$ and $B_1 = \{t^B\}$, each containing one spike. In this case (1) implies that we can write

$$\begin{aligned} d^{VP}(A_1, B_1; q) &= \min\{q|t^A - t^B|, 2\} \\ &= \begin{cases} q|t^A - t^B| & \text{if } |t^A - t^B| < 2/q \\ 2 & \text{otherwise.} \end{cases} \end{aligned}$$

Consequently, if the two spike times differ by an amount less than $2/q$, then the cost of shifting a spike varies linearly with the difference between the two spike times. Otherwise the cost 2, of deleting or inserting a spike is cheaper. In the extreme case when $q = 0$, the distance depends only on the number of spikes in each spike train and not on spike timing. Hence for our simple example of spike trains with one spike each, $d(A_1, B_1) = 0$ when $q = 0$. If q is very small, then shifting a spike in time does not significantly affect the distance between two spike trains. Hence the precise timing of a spike does not matter and the distance is essentially a measure of difference in spike count. However, if q is very large then small differences in spike timing lead to large distances so that the algorithm is sensitive to spike timing.

The VP metric was initially designed for analysis of spike trains corresponding to a single neuron, obtained on multiple presentations of different known stimuli. Based on Victor and Purpura's work, spike trains are considered to have similar post synaptic effects if they are similar, as measured by $d^{VP}(A, B; q)$.

3.4.2 Van Rossum spike metrics

The Van Rossum (VR) metric [30, 31] refers to any metric that is obtained via a two step method: First, any given spike train $T = \{t_1, \dots, t_n\}$ is mapped to an infinite dimensional vector space of continuous functions, $L^2[0, T]$, by smoothing (or filtering) the spike train with a kernel function K , via the mapping: $T \mapsto f(t) = \sum_{i=1}^n K(t - t_i) \in L^2[0, T]$. Second, the distance between any two filtered spike trains, $f, g \in L^2[0, T]$, is computed using the L^p norm given by

$$d(f, g) = \left\{ \int_0^T |f(t) - g(t)|^p dt \right\}^{\frac{1}{p}}. \quad (2)$$

Examples of most commonly used kernels include the boxcar window,

$$\begin{cases} \frac{1}{\Delta t} & \text{if } -\frac{\Delta t}{2} \leq t \leq \frac{\Delta t}{2} \\ 0 & \text{otherwise} \end{cases},$$

the Gaussian kernel

$$K(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{t^2}{2\sigma^2}\right),$$

the decaying exponential kernel

$$K(t) = \begin{cases} \frac{1}{\tau} e^{-\frac{t}{\tau}} & \text{if } t \geq 0 \\ 0 & \text{otherwise,} \end{cases}$$

and the Laplace kernel

$$K(t) = \frac{1}{\tau} e^{-\frac{|t|}{\tau}}$$

where $\sigma, \Delta t, \tau$ are positive free parameters, referred to as the bandwidth, and control the width of the kernels. The kernels can be viewed as representing the effect of a given spike across time. The kernel bandwidths determine how much variability in the spike times is incorporated into the distance measure. In general, when the bandwidth is large, the corresponding metrics approximate the spike count rate. On the other hand, when the bandwidth is small, the corresponding metrics are sensitive to small changes in spike times.

4 Current Project

4.1 Our framework

The goal of our current project is to analyze real-world spike time data without assuming a particular relationship between spikes and external variables. For example, one target data set is obtained by measuring single units (place cells) from the CA1 region of the rat hippocampus during a behavioral experiment in which a rat is performing a spatial task along a circular track (see figure 3). As place cells fire predominantly when the rat is in particular regions [33, 5], neural activity is correlated with space. However, our goal is an analysis approach that does not assume such correlation but instead discovers structure in the neural data alone. Such an analysis would then be applicable to spike time data sets where such a strong relationship with external variables is not evident.

We hypothesize that the high dimensional spike time data from neural activity may lie along a low dimensional manifold as the rat is moving along a one-dimensional track. Therefore, we use dimensionality reduction to learn the underlying structure of the data (see section 3.3). We then check to see how the low dimensional structure learned from the data itself corresponds to external variables such as the animal’s position along the track.

To do dimensionality reduction, we need a distance measure (see section 3.4) to model pairwise relations between spike trains since our data is spikes. The dimensionality reduction algorithms we use require a summary of pairwise distances between spike trains as an input in order to learn the low dimensional model.

We propose a novel method of preprocessing spike times by looking at the time since the previous spike, which we refer to as the previous time measure (see section 4.3). The previous time function (see section 4.3) captures the history of neural activity even in long intervals between spikes.

Since we do not have a suitable measure of error to analyze the performance of Diffusion maps and PCA on our real-world spike data, we design a synthetic experiment (simulation) as a first step towards developing and testing the effectiveness of our preprocessing technique on spike data (see section 4.4-4.5). To determine the performance of Diffusion maps and PCA on synthetic data, we carry out our analyses using two different synthetic data sets: a clean firing rate data set and a data set consisting of spikes generated as a Poisson process. The generated spike data is then preprocessed using our previous time measure before carrying out dimensionality reduction. We use the clean firing rate data set as our reference to determine how well both algorithms perform on the synthetic data.

Our preliminary results on synthetic experiments show that spike time data preprocessed by the previous time measure capture the simulated four and a half laps taken by a rat around a circular track. We use the Diffusion maps algorithm to obtain a low dimensional model of the synthetic data and compare the algorithm’s performance with that of PCA.

The sections that follow are organized as follows: In section 4.2 and 4.3, we define the previous time function and illustrate it graphically. In addition, we give a description of our simulation and analyzes used. In section 5, we show our results of diffusion maps and PCA on synthetic data sets. Finally in section 6, we give a brief discussion on our results and suggest possible future directions.

4.2 Nature of spike data

The spike data set consists of multiple single-unit single-trial spike trains,

$$T^{\text{spike}} = \{\{t^j\}, 1 \leq j \leq N\}$$

recorded from N neurons where, $\{t^j\} = \{t_1^j, \dots, t_{n_i}^j\}$, represents a sequences of n_i recorded times at which the spikes of the j^{th} neuron occurred.

4.3 The previous time function

Previous time refers to the time since the last spike of a given neuron. The previous time function captures the history of the neuron’s activity even in long intervals between spikes. Given any time t , define the previous time function of the j^{th} neuron, denoted by $P^j(t)$ as follows. Let

$$t_{\text{prev}}^j(t) = \max\{t_i^j \mid t_i^j < t, 1 \leq i \leq n_i\} \text{ for all spike trains } \{t^j\} \in T^{\text{spike}},$$

where t_i^j denotes the i^{th} spike of the j^{th} neuron. The previous time function, $P^j(t)$, is given by,

$$P^j(t) = t_{\text{prev}}^j(t) - t. \tag{3}$$

Figure 5 below illustrates how the previous time function is obtained from the firing activity of a given neuron.

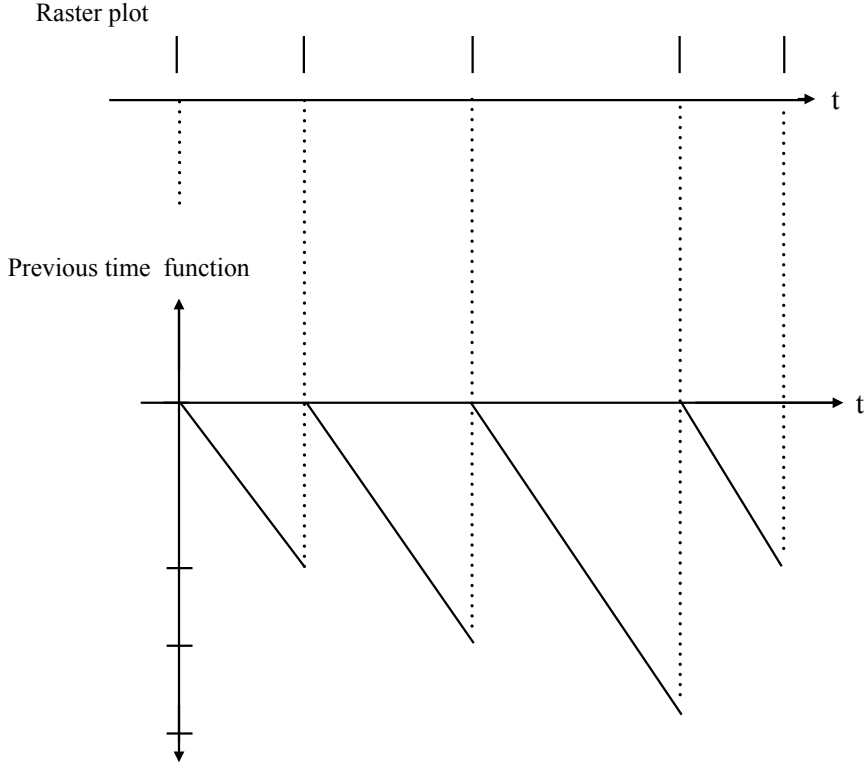


Figure 5: The previous time function $P^j(t) = t_{\text{prev}}^j(t) - t$, based on a raster plot of a single neuron. The ticks on the raster plot represent the spike time of a neuron. The previous time function captures the history of neural activity even in long intervals between spikes. $P^j(t)$ is equal to zero when a neuron fires a spike and then moves away from zero with a slope of negative one to a height equal to the difference between two adjacent spike times, below the horizontal axis.

4.4 Simulation

We consider a population of $N = 32$ neurons (idealized place cells) encoding a one-dimensional circular variable, $\theta(t)$, which represents the position of a rat, at any time t along a circular track, during a behavioral task (cf. figure 3). In the simulation, we let the rat move at a constant speed c , setting $\theta(t) = ct \pmod{2\pi}$, where time t , is in seconds. The quantity $c = \frac{2\pi}{T_{\text{lap}}}$, is the speed of the animal, and T_{lap} is the time taken for the animal to make one lap around the circle. From $\theta(t)$, we model the place field of the j^{th} neuron using a Gaussian

$$g^j(\theta) = f_{\text{bg}} + f_{\text{max}} \exp\left(-\frac{\text{dist}^2(\theta - \phi_j)}{2\epsilon_j^2}\right) \quad (4)$$

where f_{bg} is the background firing rate that is independent of the underlying stimulus, f_{max} denotes the maximum firing rate of each neuron, ϵ_j is the width of the j^{th} neuron's place field (ϵ_j is chosen uniformly at random in $[0.01, 1]$) and ϕ_j is the preferred position of the

j^{th} neuron or the center of the j^{th} place field. The centers of the place fields are equally spaced, that is, we set

$$\phi_j = \frac{j2\pi}{N}, \text{ for } 1 \leq j \leq N.$$

Intuitively, $g^j(\theta)$, models the likelihood that the rat’s neuron fires given the animal’s position $\theta(t)$, in space. The Gaussian function indicates that the rat’s place cell is likely to fire maximally at the center of the cell’s place field which mimics a characteristic of a real-world rat’s place cells. Our model assumes that each neuron in the simulated rat’s brain has a single place field. This is a realistic assumption as most place cells have one place field (see section 6 on possible future directions).

Given any two angles, $\theta_1, \theta_2 \in [0, 2\pi]$, the quantity $\text{dist}(\theta_1, \theta_2)$ denotes the shortest distance between two points on a unit circle, given by

$$\text{dist}(\theta_1, \theta_2) = |((\theta_1 - \theta_2 + \pi) \bmod 2\pi) - \pi|.$$

Our simulation assumes that information about the animal’s environment is encoded by the firing rate function

$$R^j(t) = g^j(\theta(t)).$$

Thus given a rat position $\theta(t)$, all the N neurons in the simulated rat’s brain fire according to the firing rate functions $R^j(t)$, for $j = 1, \dots, N$.

4.5 Analysis

We set the start and end time of the synthetic experiment to be 0 and T respectively. We then divide the interval $[0, T]$, into equal sub-intervals of width Δt . We carry out our analyses using two different data sets: a clean firing rate data set and a data set consisting of spikes generated as a Poisson process. We use two data sets because we do not have a suitable measure of error for analyzing the performance of diffusion maps and PCA. Hence, we use the clean firing rate data set as our reference to determine how well both algorithms perform on the spike time data set.

First, we sample from the firing rate function $R^j(t)$, for each time point $t \in [0, T]$, to obtain one data point $(R^1(t), R^2(t), \dots, R^N(t)) \in \mathbb{R}^N$.

To obtain the second data set, we generate spikes for a single trial. The spike times $\{t_1^j, \dots, t_{n_i}^j\}$, of the j^{th} neuron are generated as a non-homogeneous Poisson process at a rate $R^j(t)$, in the interval $[0, T]$. Next, we define the previous time function (see section 4.3) based on the generated set of spike trains. For each time point $t \in [0, T]$, we sample from the previous time function $P^j(t)$, to obtain one data point $(P^1(t), P^2(t), \dots, P^N(t)) \in \mathbb{R}^N$.

From Figure 5, we see that the previous time function is undefined at the beginning of the experiment where there is no recorded spike. In our analysis, we insert a spike before the start of the experiment by computing the mean previous time obtained after running one simulation of the previous time data. This ensures that the previous time function is defined for all time (see discussion in section 6 for future directions).

At each time point, $t \in [0, T]$, the position of the animal, $\theta(t)$, is associated with a previous time function, $P^j(t)$, and a firing rate function, $R^j(t)$. We analyze both data sets using the diffusion maps algorithm and PCA. Note that the value of $\theta(t)$ does not enter these analyzes.

4.6 Distance and similarity measure used in results section

Given two time points t_1 and t_2 , we write the corresponding previous time vectors

$$\mathbf{P}(t_1) = (P^1(t_1), P^2(t_1), \dots, P^N(t_1)) \quad \text{and} \quad \mathbf{P}(t_2) = (P^1(t_2), P^2(t_2), \dots, P^N(t_2)) \quad \text{in} \quad \mathbb{R}^N.$$

We use the l_1 norm to compute the distance between the two vectors:

$$d(\mathbf{P}(t_1), \mathbf{P}(t_2)) = \sum_{j=1}^N |P^j(t_1) - P^j(t_2)|. \quad (5)$$

Similarly, we use the l_1 -norm to compute the distance between two firing rate vectors:

$$d(\mathbf{R}(t_1), \mathbf{R}(t_2)) = \sum_{j=1}^N |R^j(t_1) - R^j(t_2)|. \quad (6)$$

We use the l_1 norm instead of the l_2 norm because the latter emphasizes large differences between data points which diverges from our goal of learning similar brain states based on pairwise distances between spike trains.

We form pairwise similarities, w_{ij} , between each data point using the Gaussian kernel which we eventually use as weights in the diffusion maps algorithm. The weights are given by,

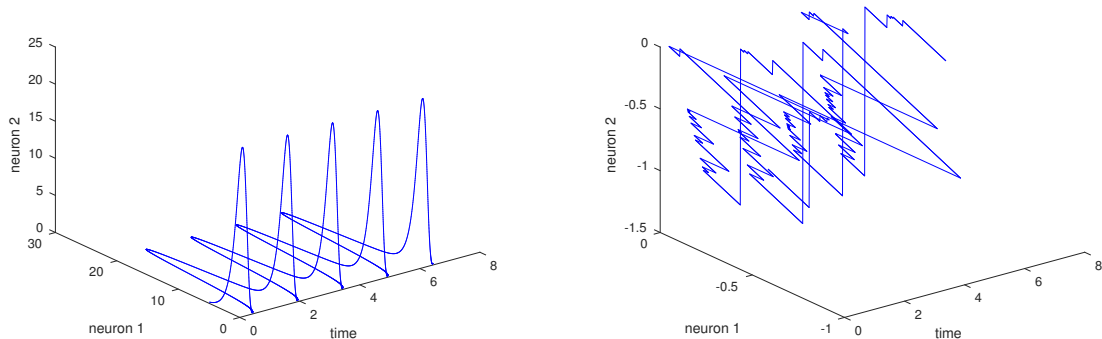
$$w_{ij} = e^{-\frac{\text{dist}^2(\mathbf{x}_i, \mathbf{x}_j)}{2\sigma^2}}$$

where $\text{dist}(\mathbf{x}_i, \mathbf{x}_j)$ is given by equations (5) and (6).

5 Results and interpretation

In this section, we report results on synthetic data using diffusion maps. We compare our results with those obtained via PCA, a popular linear dimensionality technique. We apply PCA and diffusion maps to both the clean firing rate data set and the spike time data set preprocessed with the previous time measure to obtain previous time data (see Figure 6).

Figure 7 (left column) shows that both diffusion maps and PCA capture the animal's motion around the circular track when applied on firing rate data. This is not surprising since we are using clean firing rate data that encodes the animal's position around the track. However, when using previous time data, diffusion maps appears to outperform PCA (see the right column of Figure 7). We see that analyzing previous time data using diffusion maps captures the simulated four and half laps taken by the animal around the circular track while PCA fails to reveal that the rat took four and a half laps around the track. Only diffusion maps reveals the structure of a one-dimensional manifold within the neural activity, as would be expected from the fact that the neural activity is strongly correlated with the rat's position.



(a) Simulated firing rate data (FR)

(b) Simulated previous time data (Prevtime)

Figure 6: (a) Example of the firing rate of two neurons showing the modulation of the firing rates as the rat completes four and a half laps around a simulated circular track. (b) Example of the previous time of two neurons obtained after preprocessing simulated spike times using the previous time measure. The pattern produced by the previous time measure is not clear in such a plot.

5.1 Interpretation of our results

Analyzing synthetic spike time data preprocessed by the previous time measure captures the animal’s position around the simulated circular track while using diffusion maps. We regard this observation as a potentially exciting area for future investigation since previous time data contains less information compared to using clean firing rate data. Even though we are using synthetic data, these preliminary results show that pre-processing spike times using the previous time measure should be explored in future analyses of real-world spike train data from the CA1 region of the rat hippocampus.

6 Future direction

First, at the beginning of the experiment, there are no recorded spike times. However, there must be at least one spike before the previous time function is defined. This raises a problem on how to address the boundary condition at the beginning of the experiment. As it stands, we have addressed this problem by first running the experiment once to obtain the mean of the sampled previous time data, and afterwards, we insert a spike before the beginning of the experiment based on the computed previous time average, and then use that to simulate new previous time data. We think that this is not the best approach, as the mean may not be the best representation of the underlying distribution. We plan to find another way of addressing the boundary condition for the previous time vectors. For instance, we plan to insert a spike before the beginning of the experiment based on samples drawn from a correctly estimated underlying model of the data.

Second, we currently do not have a measure of error for analysing the effectiveness of our approach. We tried using Shannon mutual information on real-world data. However, we found that high mutual information may be caused by spatial noise, such as changes in the animal’s head direction rather than indicating that the analysis captured the animal’s movement around the track. We plan to use the Fisher information as a goodness of

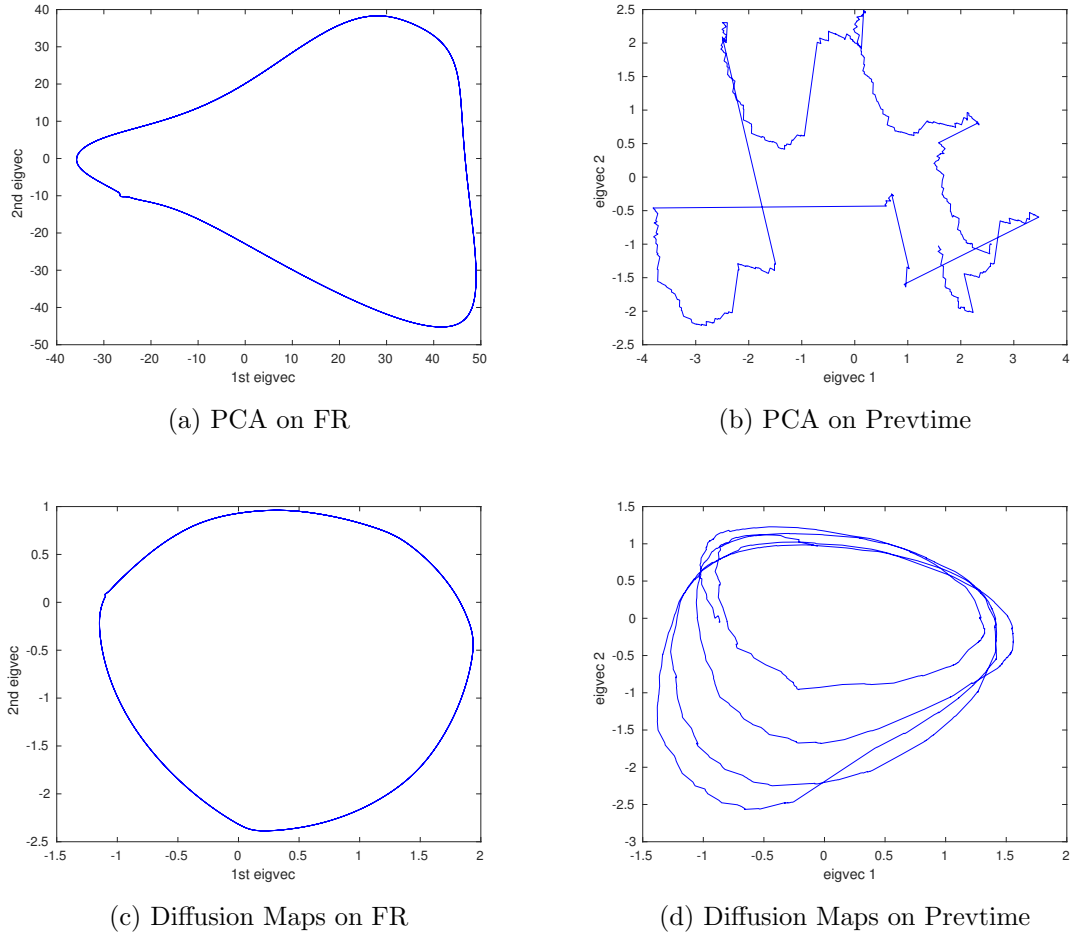


Figure 7: Performance of PCA (first row) and Diffusion maps (second row) on simulated firing rate data (figure 6 a), and previous time data (figure 6 b). A single point on any curve represents the value of the first and second eigen vector at any time t . (a, c) Each circle corresponds to the four and a half laps made by the rat as the animal goes around a circular track. All the laps are superimposed on top of each other because the results in the left column are obtained after applying PCA and Diffusion maps on clean firing rate data (with no noise). (b) Example showing that PCA applied to previous time data fails to reveal the four and a half laps taken by the rat around a circular track. (d) Example showing that Diffusion maps applied to previous time data recovers the four and a half laps taken by the rat around the circular track. Since the rat's position in space is a circular curve parametrized in time, the top two eigen vectors of diffusion maps on previous time data capture the circular motion of the rat.

measure in future work since it can take advantage of the fact that results lie in a continuous space. In addition, Fisher information is highest when the variance of the estimated model parameter is very small. This could yield a more reliable estimate of how well we captured a one-dimensional manifold that corresponds to the animal's position along the track.

We plan to preprocess our real-world spike time data using the previous time measure in order to look for structure in data itself so as to see how the structure corresponds to external variables. The data set consists of spike times obtained by measuring from $N = 32$ single units (place cells) from the CA1 region of a rat hippocampus during a behavioral experiment in which the rat is performing a spatial task along a circular track (see figure 3).

Finally, we plan to look at other types of distance measures for quantifying neural response variability such as variants of the kendal tau distance or a weighted combination of some existing distances instead of the l_1 distance.

An improved approach such as the one we envision and outline here may provide some insight on how to analyze spike time data when the relationship between spikes and external variables is unknown. By looking for structure in the spike time data itself, a researcher may be able to see how the underlying structure corresponds to external variables such as a stimulus or a location without imposing apriori ideas on the relationship between these variables and the observed response.

References

- [1] Donald R Humphrey and Edward M Schmidt. Extracellular single-unit recording methods. *Neurophysiological techniques*, pages 1–64, 1990.
- [2] J. O’Keefe and J. Dostrovsky. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain Research*, 34(1):171–175, 1971.
- [3] John O’Keefe and D H Conway. Brain Hippocampal Place Units in the Freely Moving Rat : Why They Fire Where They Fire. *Experimental Brain Research*, 31(4):573–590, 1978.
- [4] P Dayan and LF Abbott. *Theoretical neuroscience*. 2001.
- [5] Neil Burgess, Michael Recce, and John O’Keefe. A model of hippocampal function. *Neural Networks*, 7(6-7):1065–1081, 1994.
- [6] Lf Abbott and Peter Dayan. Theoretical Neuroscience. *Computational and Mathematical Modeling of Neural . . .*, 60(3):489–95, 2001.
- [7] a P Georgopoulos, J F Kalaska, R Caminiti, and J T Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *J.Neurosci.*, 2(11)(11):1527–1537, 1982.
- [8] D. H. Hubel and T N Wiesel. Receptive Fields and Functional Architecture of monkey striate cortex. *Journal of Physiology*, 195:215–243, 1968.
- [9] J J Hopfield. Pattern recognition computation using action potential timing for stimulus representation. *Nature*, 376(6535):33–36, 1995.
- [10] A. David Redish. Vicarious trial and error. *Nature Reviews Neuroscience*, 17(3):147–159, 2016.
- [11] Seychelle M Vos, Elsa M Tretter, Bryan H Schmidt, James M Berger, and Cell Biology. HHS Public Access. 12(12):827–841, 2015.
- [12] Matthew T Kaufman, Mark M Churchland, Stephen I Ryu, and Krishna V Shenoy. Cortical activity in the null space: permitting preparation without movement. *Nature neuroscience*, 17(3):440–448, 2014.
- [13] Ofer Mazor and Gilles Laurent. Transient dynamics versus fixed points in odor representations by locust antennal lobe projection neurons. *Neuron*, 48(4):661–673, 2005.
- [14] D. R. Kipke, W. Shain, G. Buzsaki, E. Fetz, J. M. Henderson, J. F. Hetke, and G. Schalk. Advanced Neurotechnologies for Chronic Neural Interfaces: New Horizons and Clinical Opportunities. *Journal of Neuroscience*, 28(46):11830–11838, 2008.
- [15] Krishna V Shenoy, Maneesh Sahani, and Mark M Churchland. Cortical control of arm movements: a dynamical systems perspective. *Annual review of neuroscience*, 36:337–359, 2013.

- [16] John P Cunningham and Byron M Yu. Dimensionality reduction for large-scale neural recordings, nov 2014.
- [17] William Bialek and Fred Rieke. Reliability and information transmission in spiking neurons. *Trends in Neurosciences*, 15(11):428–434, 1992.
- [18] Neil D. Lawrence. A Unifying Probabilistic Perspective for Spectral Dimensionality Reduction: Insights and New Models. 13:1–31, 2010.
- [19] I T Jolliffe. *Principal Component Analysis and Factor Analysis*. Principal Component Analysis. Springer New York, 1986.
- [20] J B Tenenbaum, V de Silva, and J C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science.*, 290(5500):2319–2323, dec 2000.
- [21] Cox M Cox T. *Multidimensional scaling*. 2000.
- [22] K V Mardia. *Multivariate analysis*. Probability and mathematical statistics. Academic Press, London ; New York, 1979.
- [23] Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.
- [24] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.
- [25] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.
- [26] Ulrike Von Luxburg. A tutorial on spectral clustering. (August 2006):395–416, 2007.
- [27] E N Brown, R E Kass, and P P Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nat.Neurosci.*, 7(5):456–461, 2004.
- [28] J. D. Victor and Keith P Purpura. Nature and precision of temporal coding in visual cortex: a metric-space analysis. *Journal of neurophysiology*, 76(2):1310–26, 1996.
- [29] Jonathan D Victor, Keith P Purpura, and Jonathan D Victor. No Title. (212):1–60, 1998.
- [30] M C W Van Rossum. Communicated by Jonathan Victor. *Neural Computation*, 763:751–763, 2001.
- [31] Conor Houghton and Jonathan Victor. Measuring representational distances—the spike-train metrics approach. *Visual Population Codes—Toward a Common Multivariate Framework for Cell Recording and Functional Imaging*, pages 391–416, 2010.
- [32] S Schreiber, J M Fellous, D Whitmer, and P Tiesinga. A new correlation-based measure of spike timing reliability. 54:925–931, 2003.
- [33] John O’Keefe and Jonathan Dostrovsky. The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat. *Brain research*, 34(1):171–175, 1971.

Appendices

A Linear algebra review

A.1 Definitions

Definition 1 An $n \times n$ symmetric matrix \mathbf{A} is positive semidefinite (PSD) if the quadratic form

$$\mathbf{x}\mathbf{A}\mathbf{x}^\top \geq 0, \text{ for all non-zero vectors } \mathbf{x} \in \mathbb{R}^n$$

Given any $n \times n$ matrix \mathbf{A} , the matrices $\mathbf{A}\mathbf{A}^\top$ and $\mathbf{A}^\top\mathbf{A}$ are symmetric PSD.

Definition 2 Let $\mathbb{R}^{n \times m}$ denote the vector space of $n \times m$ matrices with real entries. Given any $n \times m$ matrix \mathbf{A} with real entries, there exist orthonormal matrices $\mathbf{U} \in \mathbb{R}^{n \times n}$ and $\mathbf{V} \in \mathbb{R}^{m \times m}$ such that $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ with $\mathbf{\Sigma} = \text{diag}(\sigma_1, \dots, \sigma_k)$ where $k = \min(n, m)$ and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k \geq 0$. The numbers $\sigma_i, 1 \leq i \leq n$ are called the singular values of \mathbf{A} .

A.2 Properties of SVD

Given any $n \times m$ real matrix A of rank r , there exist matrices $\mathbf{U} \in \mathbb{R}^{n \times r}$ and $\mathbf{V} \in \mathbb{R}^{r \times r}$ satisfying $\mathbf{U}^\top\mathbf{U} = \mathbf{V}^\top\mathbf{V} = \mathbf{I}$ such that $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}_1\mathbf{V}^\top$ with $\mathbf{\Sigma}_1 = \text{diag}(\sigma_1, \dots, \sigma_r)$ and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$. This factorization of \mathbf{A} is called the truncated SVD of A . The rank of A is equal to the number of non-zero singular values of A . If \mathbf{U} and \mathbf{V} are made up of column vectors so that $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_r]$ and $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_r]$ then $\{\mathbf{u}_1, \dots, \mathbf{u}_r\}$ and $\{\mathbf{v}_1, \dots, \mathbf{v}_r\}$ are called the left and right singular vectors of A respectively.

The matrix A admits the SVD expansion

$$\mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^\top.$$

The real numbers $\{\lambda_1, \dots, \lambda_r\}$ where $\lambda_i = \sigma_i^2$ for all $i = 1, \dots, r$ are the non-zero eigenvalues of both symmetric matrices $\mathbf{A}^\top\mathbf{A}$ and $\mathbf{A}\mathbf{A}^\top$. The right singular vectors of \mathbf{A} are the eigenvectors of $\mathbf{A}^\top\mathbf{A}$. The left singular vectors of \mathbf{A} are the eigenvectors of $\mathbf{A}\mathbf{A}^\top$. The spectral decomposition of $\mathbf{A}^\top\mathbf{A}$ is given by $\mathbf{A}^\top\mathbf{A} = \mathbf{V}\mathbf{D}_1\mathbf{V}^\top$ and that of $\mathbf{A}\mathbf{A}^\top$ is given by $\mathbf{A}\mathbf{A}^\top = \mathbf{U}\mathbf{D}_2\mathbf{U}^\top$ where \mathbf{U} and \mathbf{V} are the SVD factors upto a sign, \mathbf{D}_1 and \mathbf{D}_2 are diagonal matrices containing the r singular values of A .

B Proximity measures

Definition 3 A proximity measure characterizes how close two objects are, using either a similarity or dissimilarity measure between them [21]. A similarity measure characterizes how similar two objects are while a dissimilarity measure or distance characterizes how dissimilar two objects are.

Definition 4 Let X be a set of n objects, $\{x_1, \dots, x_n\}$, that are not necessarily vectors. A distance, d , on X , is a real-valued function $d : X \times X \rightarrow \mathbb{R}$ which assigns to each pair of objects, (x_i, x_j) , a real number, d_{ij} , satisfying two conditions, for all i and j from 1 to n .

i) $d_{ij} \geq 0$ and $d_{ij} = 0$ if and only if $i = j$

ii) $d_{ij} = d_{ji}$

A metric is a distance function, d , which satisfies the triangle inequality: $d_{jk} \leq d_{ij} + d_{ik}$, for all i, j, k

Definition 5 A similarity measure $s : X \times X \rightarrow \mathbb{R}$ is a function which assigns to each pair of objects, (x_i, x_j) , a real number, s_{ij} , such that for all i and j , we have:

i) $0 < s_{ij} \leq s_{ii}$

ii) $s_{ij} = s_{ji}$

The similarity measure, s_{ij} , can be obtained from a distance, d_{ij} , using any of the three operations: $s_{ij} = c - d_{ij}$ or $d_{ij} = 1/s_{ij} - c$, or $d_{ij} = s_{ii} + s_{jj} - 2s_{ij}$, where c is a constant.

A matrix of distances between pairs of objects, $D = (d_{ij})$, is called a distance matrix or dissimilarity matrix and that of similarities between pairs of objects, $S = (s_{ij})$, is called a similarity matrix.

C Graph Laplacians and their properties

- 1) The unnormalized graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$ of a connected, weighted, undirected graph G , with n vertices is positive semidefinite and hence has a basis of n non-negative eigenvalues with corresponding eigenvectors.
- 2) The smallest eigenvalue of \mathbf{L} is $\mathbf{0}$ with corresponding eigenvector $\mathbf{1}$.
- 3) The second eigenvalue of \mathbf{L} is positive if and only if the graph G is connected.

There are two other types of graph Laplacians, namely the symmetric normalized graph Laplacian

$$\mathbf{L}_{sym} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{W} \mathbf{D}^{-\frac{1}{2}} \quad (7)$$

and the graph Laplacian related to a random walk on a graph which is not symmetric

$$\mathbf{L}_{rw} = \mathbf{D}^{-1} \mathbf{L} = \mathbf{I} - \mathbf{D}^{-1} \mathbf{W} \quad (8)$$

where \mathbf{D} is the degree matrix and \mathbf{W} is the similarity matrix.

D Random walk on a graph

Let $G = (V, E)$ be a connected, weighted and undirected graph with a set of n vertices, $V = \{v_1, \dots, v_n\}$. A stochastic process that jumps from one vertex, v_i , to another vertex, v_j , on the graph, G , is called a random walk on G . Denote the random walk on G by the set $\{X(t)\}_{t \in \mathbb{N}}$. A matrix \mathbf{M} is called a stochastic transition matrix or a transition probability matrix, if the entries in each row of \mathbf{M} are non-negative and add up to one. We require that the transition probability of jumping from vertex v_i to vertex v_j on G is proportional to the edge weight, w_{ij} , between the two vertices. Let $p_{ij} = \text{Prob}(X(t+1) = j | X(t) = i)$ be the transition probability of jumping from vertex v_i to vertex v_j in one step. Define p_{ij} by

$$p_{ij} = \frac{w_{ij}}{d_i} \quad (9)$$

where, d_i , denotes the degree of the i^{th} vertex, and set $\mathbf{P} = (p_{ij})$. Since the degree matrix $\mathbf{D} = \text{diag}(d_1, \dots, d_n)$, contains the sum of the i^{th} row of the adjacency matrix, \mathbf{W} , we know that the matrix

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W} \quad (10)$$

is indeed a transition probability matrix. This follows from the fact that $d_i > 0$ (because G is connected) and $w_{ij} \geq 0$ since it is a similarity. Moreover, the definition of \mathbf{D} implies that the rows of \mathbf{P} sum to one. \mathbf{P} is a probability transition matrix implies that it satisfies the equation $\mathbf{P}\mathbf{1} = \mathbf{1}$ where $\mathbf{1}$ is the constant vector of all ones. Thus 1 is an eigenvalue of \mathbf{P} with corresponding constant eigenvector $\mathbf{1}$.

Observe that \mathbf{P} is precisely the random walk graph Laplacian, \mathbf{L}_{rw} , in equation (8). Given that the random walk starts at node i , so that $X(0) = i$, the probability that the random walk is at vertex v_j after t steps is given by

$$\text{Prob}(X(t) = j | X(0) = i) = p_{ij}^t$$

Thus the probability distribution of the random walk at time t , given that it started from vertex v_i is given by the i^{th} row of \mathbf{P}^t , i.e.

$$\text{Prob}(X(t) | X(0) = i) = \mathbf{1}^\top \mathbf{P}^t = \mathbf{P}^t(i, :) \quad (11)$$

However, we cannot apply spectral decomposition to \mathbf{P} , because it is not symmetric hence not PSD. This requires a normalization using the adjacency matrix, \mathbf{W} , in order to get a PSD similarity matrix \mathbf{S} .

The eigenvalues, λ_k , of \mathbf{P} satisfy $|\lambda_k| < 1$ for all k .

E Diffusion maps algorithm

The diffusion maps algorithm is given by the following steps.

- 1) Given an $n \times n$ distance matrix $D = (d_{ij})$, build a graph $G = (V, E)$ on the data by identifying the each data point as a vertex v_i from the vertex set, $V = \{v_1, \dots, v_n\}$, on G .

- 2) Compute the similarity matrix or weight matrix, $\mathbf{W} = (w_{ij})$, between the vertices v_i using weights

$$w_{ij} = e^{-\frac{\text{dist}^2(x_i, x_j)}{2\sigma^2}} = e^{-\frac{d_{ij}^2}{2\sigma^2}}$$

- 2a) The weight matrix, \mathbf{W} can be any PSD kernel (similarity) and d_{ij} is any distance appropriate for the data. The bandwidth or tuning parameter, σ restricts transitions between points to a $\sqrt{\epsilon}$ neighborhood.
- 2b) Classically, the diffusion maps algorithm uses Euclidean distance $d_{ij} = \|x_i - x_j\|_2^2$ in the Gaussian kernel above.
- 3) Compute degree matrix $\mathbf{D} = \text{diag}(d_1, \dots, d_n)$ where d_i is the degree of the i^{th} node.
- 4) Define the random walk on the graph by specifying the transition probabilities, defining

$$p_{ij} = \frac{w_{ij}}{d_i}.$$

- 5) Obtain the random walk graph Laplacian or transition probability matrix \mathbf{P} by defining

$$(p_{ij}) = \mathbf{P} = \mathbf{D}^{-1}\mathbf{W}. \quad (12)$$

- 6) Since \mathbf{P} is not symmetric, normalize (12) using \mathbf{D} to obtain a PSD similarity matrix \mathbf{S} given by

$$\mathbf{S} = \mathbf{D}^{\frac{1}{2}}\mathbf{P}\mathbf{D}^{-\frac{1}{2}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}. \quad (13)$$

- 7) Apply spectral decomposition to \mathbf{S} in (13), write

$$\mathbf{S} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^\top, \quad (14)$$

and order the eigenvalues, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$, where, $\mathbf{\Sigma} = \text{diag}(\lambda_1, \dots, \lambda_n)$. Write $\mathbf{S} = [\mathbf{v}_1, \dots, \mathbf{v}_n]$.

- 8) From (13), write

$$\mathbf{P} = (\mathbf{D}^{-\frac{1}{2}})\mathbf{S}(\mathbf{D}^{\frac{1}{2}}) = (\mathbf{D}^{-\frac{1}{2}}\mathbf{V})\mathbf{\Sigma}(\mathbf{D}^{\frac{1}{2}}\mathbf{V})^\top.$$

- 9) Let $\mathbf{\Phi} = \mathbf{D}^{-\frac{1}{2}}\mathbf{V} = [\phi_1, \dots, \phi_n]$ and $\mathbf{\Psi} = \mathbf{D}^{\frac{1}{2}}\mathbf{V} = [\psi_1, \dots, \psi_n]$ to get

$$\mathbf{P} = \mathbf{\Phi}\mathbf{\Sigma}\mathbf{\Psi}^\top.$$

The bases $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are a biorthogonal system, and hence satisfy $\mathbf{\Phi}\mathbf{\Psi}^\top = \mathbf{\Phi}^\top\mathbf{\Psi} = \mathbf{I}_{n \times n}$, which implies that $\phi_j^\top \psi_k = \delta_{ij}$.

- 10) Observe that $[\phi_1, \dots, \phi_n]$ and $[\psi_1, \dots, \psi_n]$ are the right and left singular vectors of \mathbf{P} , respectively. Hence we can write for any time t ,

$$\mathbf{P}^t = \mathbf{\Phi}\mathbf{\Sigma}^t\mathbf{\Psi}^\top = \sum_{k=1}^n \lambda_k^t \phi_k \psi_k^\top. \quad (15)$$

- 11) Equation (15) shows the expansion of \mathbf{P}^t in terms of the basis vectors $\{\psi_k\}$, thus by the observation in equation (11), it follows that the diffusion map for the i^{th} vertex v_i is given by

$$v_i \mapsto \mathbf{P}^t(i, :) = \begin{bmatrix} \lambda_1^t \phi_1(i) \\ \lambda_2^t \phi_2(i) \\ \vdots \\ \lambda_n^t \phi_n(i) \end{bmatrix}.$$

- 12) Since $|\lambda_k| < 1$, for a sufficiently high power of t , most λ_k^t are very small and hence can be discarded. Also $\mathbf{P}\mathbf{1} = \mathbf{1}$ implies that the first constant vector yields a trivial solution and can be discarded. Hence a diffusion map ϕ_t for an embedding of the i^{th} vertex v_i of a graph G in low dimension $p \ll n$ is the map

$$\phi_t(v_i) = \begin{bmatrix} \lambda_2^t \phi_1(i) \\ \lambda_3^t \phi_2(i) \\ \vdots \\ \lambda_{p+1}^t \phi_{p+1}(i) \end{bmatrix}.$$