A geometrical approach to spike train noise.

James B. Gillespie

A thesis submitted to the University of Dublin, Trinity College
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Supervisor
Conor Houghton

School of Mathematics,
Trinity College Dublin

December 11, 2012

‘If it wasn’t for the mist we could see your home across the bay... You always have a green light that burns all night at the end of your dock.’ [1]

[1] F. Scott Fitzgerald

Declaration

I, the undersigned, (known as the candidate) declare that:

1. this work has not been submitted as an exercise for a degree at this University or at any other University *and*

2. this is entirely the candidate's own work and all work by others which is discussed is duly referenced and acknowledged *and*

3. the candidate agrees that the Library may lend or copy the thesis upon request and that this permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

Summary

Mathematically, spike trains are elusive processes. They encode information, although how this information is contained in a spike train is still not clear. Same-stimulus spike trains display structural similarities, yet noise is also always present, bounding the precision with which a signal can be transmitted. Taking these varied factors into account is difficult, and here a geometrical approach is adopted in developing flexible mathematical descriptions of spike train properties.

Spike trains naturally form a metric space: a set with elements and a distance measure, known as a metric. However, having a metric space does not promise the existence of a set of coordinates to describe its elements, and in practice it is found that no natural spike train coordinates are available. This lack of coordinates makes the application of information theory, a widely used area in spike train analysis, challenging to implement. Here, models of spike train noise and information processing are developed and phrased in terms of metric distances. Such models do not rely on the existence of spike train coordinates.

This thesis is divided into four parts. First, the normed vector space of Gaussian channel capacity theory is rephrased in terms of metric quantities. This is tested on the metric space of spike trains. This results in the first model of spike train noise in a metric space. Second, the noise found in the distribution of spike times is investigated using an edit-distance metric. This distribution is found to be well modelled by a hyper-Laplace. Third, a method of generating artificial spike trains is examined. Lastly, the geometry of spike trains is seen to embed the neurons themselves in a manifold-like structure called a statistical model. Low dimensional embedding methods are used to visualise the neurons in this space.

When developing neuronal models one would like those models to be both mathematically consistent with the properties of spike trains, while remaining impartial to the nature of the temporal or rate encoding of information. The two metrics used to build the neuronal models in this work each contain a free parameter. This parame-

ter measures the presence of temporal and rate coding that exists in a particular neurons spike trains. It is not fixed manually, but by using an algorithm which chooses the parameter that best clusters a neurons same-stimulus spike trains. Hence, the models developed here remain agnostic to neuronal coding schemes. Rather, it is the outcome of the same-stimulus clustering that determines which coding scheme best represents the spike train data.

The fact that spike trains are naturally described in a metric space is generally overlooked when calculating the values of quantities such as channel capacity and information. The metric space restriction is addressed in this work through developing mathematical models based only on measurements of distance between spike trains. Therefore, in each case it is the geometry of the space of spike trains that governs the quantities measured from the dataset.

Acknowledgements

I am indebted to my supervisor Dr. Conor Houghton for his tolerance, direction and help.

I acknowledge the Irish Research Council of Science, Engineering and Technology for their Postgraduate Scholarship funding, and am grateful to the James S. McDonnell Foundation for additional funding.

I would like to thank all those whose useful suggestions and advice contributed to the resulting papers. Specifically, acknowledgments go to Dan McNamee, Garrett Greene and Louis Aslett for their hypothesis testing suggestions and proofing which helped the capacity paper, Hannah Julienne for proofing and Prof Jonathan Victor for his comments on the jitter paper. Also I am grateful to Dr Guy Billings, Dr Abhinav Singh, Dr Romain Brasselet and Prof Nicholas Lesica for the interesting discussions that benefited the Noise Space chapter. Additional thanks go to Dmitri Bykov for his metric space advice during those long coffee evenings in first year, Garrett Greene for what we will call the 'healthy debates' which helped this work, Cathal Cooney for something on matrices, and Josh Tobin for revolutionizing my coding.

I dedicate this thesis to my parents. From my earliest memories their love, help and encouragement have made this possible.

Contents

Chapter 1

Introduction

The central nervous system is a vast dynamic electrical and chemical network. Its core component is the neuron, which processes and transmits information. This information is communicated from neuron to neuron in the form of sets of spikes in electrical membrane potential, known as spike trains. Accurately interpreting the encoding of this information remains a difficult task.

Understanding spike train information encoding is fundamental to understanding the brain function. Here, a metric space approach is taken to calculate the maximum information a neuron can reliably transmit, and metric analysis is used to explore and generate noise models associated with a neuron's spike trains. This will eventually lead to the development of a manifold which embeds networks of neurons on its surface. Key to all this is the idea that such spike train models naturally arise from the metric space geometry associated with the spike trains themselves.

1.1 Neurons, spikes and synapses

A neuron is a cell capable of changing the voltage difference existing between its inner and outer membrane in response to certain inputs. There are typically 100 billion neurons in the human brain, where each neuron is connected to 10,000 other neurons on average, see Fig. 1.1B. A neuron receives input from its dendritic tree, which conducts an incoming signal, see Fig. 1.1A. The output is carried by its axon to all the other cells it is connected to, where the axon conducts the outgoing signal, branching along the way.

Differences in voltage between the inside and outside membrane of the cell are created and maintained by ion pumps [3]. These are situated in the cell membrane itself. These sophisticated pumps take ions from one side of the membrane and transport them to the other. One such pump, the sodium-potassium ion pump, transports two potassium ions in through the cell membrane and three sodium ions out. This results in a large concentration of potassium ions in the cell and a large concentration of sodium ions outside. Due to the sodium ions there is a net positive extracellular voltage difference relative to the intracellular membrane. The extracellular potential is taken to be 0mV relative to the intracellular potential of approximately -70mV.

A spike is triggered if the membrane potential is increased, known as depolarization, to a certain threshold level. This level is typically around -55mV. Once this critical level is reached, ion gates in the membrane open and positively charged sodium ions flow from the outside to inside of the membrane. Firstly, due to the influx of positive charge, the inside of the membrane becomes less negative. Then, as positive charge continues to build it will typically reach a positive value of $+50$mV, the peak of the spike. At this stage the sodium gates close, and positively charged potassium ions flow to the outside of the membrane, until the resting potential is once again reached. The process for generating a spike takes less than a thousandth of a second. This depolarization, which we record as a voltage spike, travels down the axon of the cell.

The synapse is the junction between an axon of one neuron and a dendrite of another, see Fig. 1.1C. At a synapse the plasma membrane of the signal passing neuron comes into close proximity with the membrane of the signal receiving neuron. Then, depending on the type of synapse, either a chemical neurotransmitter or in a limited number of cases an electrical current is passed from axon to dendrite, allowing the spike to continue to the next neuron.

The pre-synaptic spike causes the second neuron to alter the dendritic membrane voltage. The magnitude of this change depends on both the synapse and the voltage dynamics of the second neuron itself. Many network models of neurons assign each synapse a certain strength, or weighting, defined as the amplitude of the change in the post-synaptic electrical potential due to the pre-synaptic potential. A higher synaptic weighting between two neurons will increase the probability of one neuron causing the other to fire, thus synaptic weightings isolate subsets of neurons within any network which will typically cause each other to fire. Certain neurons, inhibitory neurons, decrease membrane potential, thus reducing the probability that the neuron will fire.

If one neuron provokes another to fire then the synapse connecting them is strengthened, and similarly a failure to fire leads to a weakening in the synapse. Thus the set of weightings is itself changing, with the most probable propagation of spikes throughout the network determined by the set of synapses with the highest connection strengths. In this way electrical potential differences, called spikes, are passed through the network.

1.2 A typical electrophysiological experiment

As all datasets used throughout this work come from the auditory forebrain of the adult male zebra finch, an overview of the methods used by external laboratories to collect this extracellular data is given.

A male zebra finch learns its song during puberty. Female zebra finches lack the ability to sing, so the finch learns a large part of its song from its father. However, no two birds have the same song, and external factors will also effect the development of its song. The song can be seen to consist of vocal units called syllables, where a group of vocal units is known as a motif. The finch's song will not alter once learned, and is typically a few seconds in length, with song fine detail of the order of a few milliseconds.

Two days prior to the electrophysiological recordings a number of preparations are carried out on the finch. First, locations where electrodes will penetrate the finch brain have to be marked. At this stage a steel head support pin must also be fitted by gluing to the skull. The bird is anesthetized with isoflurane in O2. Skin and the top layer of skull underneath is then removed and the bifurcation point of the midsagittal sinus, commonly used as a coordinate reference point for field L brain regions, is used to locate regions for electrode penetration. These regions are marked with ink in areas 1.5 mm lateral and 1.2 mm anterior to the bifurcation point.

On the day of recording the bird is anesthetized using urethane. This involves three intramuscular injections of 20% urethane (70-90 micro litres) administered half an hour apart. The bird's head is held in place using posts attached to the stereotax, which is essentially a metal clamp that can be varied in position. The lower region of skull and dura matter is removed from the regions previously marked. Two tungsten electrodes are then inserted 1200-2500 micrometers into these exposed regions using a micromanipulator. One of the electrodes forms a reference ground electrode, dipped in NaCl solution and inserted into CSF, the opposite hemisphere of the brain.

The bird is then placed in a sound attenuated chamber, a small padded room, facing a loudspeaker through which a conspecific song stimulus is played. The bird is positioned facing the speaker, with the speaker approximately 20cm away from its beak, and at the same level as the speaker. Neuronal responses are recorded at approximately 100 micrometer interval depths.

Every time a stimulus is played, the extracellular output is recorded in the following way. The electrode readings are sent to a preamp, which is used to boost the gain of the signal before it is filtered. This signal is then filtered using a d.c. amplifier. The resulting filtered signal is between 300-5000 Hz. The component of the signal below 300 Hz comprises the local field potential, which is discarded for the purposes of this experiment, as is the component above 5000 Hz which has a high level of electronic noise.

The filtered signal is then sent to three sources for recording and direct outputting. First, the soundcard is used to record the waveforms of the spikes. Second, the oscilloscope is used to view the spike waveforms. Third, the audiomonitor is used to listen to the spikes as they are recorded.

It is interesting to note that the spikes can have quite distinct sounds. For instance, the spike from an inhibitory neuron tends to have a sharper pitch to it than does the spike from an excitatory neuron.

Spike sorting algorithms are then applied to the raw waveform data in order to separate a spike from a particular neuron from those of its neighbouring neurons. The specifics of the actual sorting algorithms are under constant development, but in general different spike sorting methods share common techniques. To begin, select the set of spikes that are larger than a certain threshold voltage value. Observe, either using an algorithm or directly by viewing, stacked spike waveforms for time range 1 ms before peak and 1 ms after peak. Isolate 'similar' waveforms from the data. For instance one can use software to 'draw' an outline of a waveform and similar ones can be selected computationally from the stack. If, for a now sorted spike train, 99% of inter-spike intervals in a spike train are greater than 1 ms, then the spike train is classified as a single unit cell.

It should be noted that noise distorts the spike waveforms, so trying to select spikes from a particular neuron according only to waveform will be unreliable. Combating this is difficult and depends on both the software used and the intuition of the physiologist.

1.3 Information in spike trains

Spikes encode information relating to the nature of the stimulus that caused the neuron to fire. This information is propagated throughout sections of the nervous system. At each stage the information is processed, that is, signals are sent to the set of neurons to which it is connected. The weightings of the synapses largely control the neuronal paths this information can take. While certain neurons, sensory neurons, have to pass information about the stimulus, other neurons, motor neurons, will act on this information. Motor neurons project their axons outside the central nervous system, allowing them to directly or indirectly control muscles. Thus, neurons allow for the brain function to both identify and respond to a stimulus.

Over a period of time, a list of times at which a neuron released a spike can be compiled, and the resulting list is a spike train, see Fig. 2.2. In general, information is viewed as being contained in spike trains rather than spikes.

The concept of spike train noise should be addressed, as often noise can be interpreted in a number of different ways. For instance, some electrophysiologists could regard noise as the spontaneous firing of a neuron with no stimulus present, while others might regard it as being the variability found between spike trains corresponding to the same stimulus. During this thesis, the latter interpretation is used. Neurons in zebra finch field L have been identified to encode conspecific song, [49, 44, 51, 50, 41], hence the variability found between spike trains corresponding to the same song sheds no additional information and can be regarded as noise.

It remains unclear as to how information is actually encoded within a set of spikes. Neuronal networks are in general extremely complex, with extensive interconnectivity. The responses of cells are found to depend on both their type and network location. There are two obvious methods of encoding information within a single spike train. The first is rate coding [1], where information is contained in the number of spikes generated during a certain time interval. The second is temporal coding [2], where instead information is contained in the timing of the spikes.

Whether single spike trains encode information temporally or on a rate basis remains a fundamental question in neuroscience. The temporal features of a spike train are generally discarded and focus is given to accurately modelling the rate. This has been used to impressive effect, for example in modelling the responses of motor neurons in brain machine interface. However evidence for temporal coding also exists, for instance in the spike coincidence detection found in the barn owl auditory system. Also, the speed at which information is passed throughout the visual system leaves the idea of a purely rate based coding system improbable. So by ignoring temporal coding features we seem to be leaving out some of the natural properties of spike trains. A compromise is a coding scheme that is a combination of both rate and temporal. This combination could itself vary depending on the changing properties of the neuron.

When a neuron receives input from a population of other neurons, rather than from just one, more possibilities exist to transmit information and the problem becomes more complex. For example, information could be transmitted by means of a rank code, [5], where information is contained in the relative timing of spikes received from different neurons during a certain interval. This allows for faster information processing, and is consistent with reaction times found in the visual system. Another example is of vector population coding, [6], where each neuron in a network represents movement in a certain direction, turning each firing rate into a vector, with the sum of these vectors encoding the direction of motion. During this thesis we examine the simpler case of single spike trains rather than population codes.

We are left with the idea that, for the purposes of information processing, different neurons function in different ways. During this research all data used is from the auditory part of the zebra finch forebrain, a region where spike trains exhibit apparent temporal properties. However, rather than assuming either temporal or rate coding, it would seem best to develop models which in some way remain agnostic to respective coding beliefs, but which could instead be tailored to a particular neurons coding scheme by fixing some 'coding parameter'. A spike train metric allows for such a model.

1.4 Spike train metrics

The set of spike trains form a metric space [31]. A metric space is a set of elements along with a method of measuring distance between the sets elements. Additionally this distance measure $d(x, y)$, where x and y are elements of the set, satisfies four properties. The first property is called positivity, where $d(x, y) \geq 0$. The second is non-degeneracy, that is, $d(x, y) = 0$ if and only if $x = y$. The third is symmetry, where $d(x, y) = d(y, x)$. Last is the triangle inequality, $d(x, y) \leq d(x, z) + d(z, y)$ for any element of the set z.

So a metric is a method for measuring the distance between elements of a set, and can be used to quantify the dissimilarity between two spike trains. Here, the concept of distance is an abstract one: the distance between two spike trains is directly related to how different the two trains are from one another. Now the word 'difference' itself is ambiguous, conventionally it is taken to mean either a difference in the number of spikes from train to train, or differences in the positions of the spikes.

There are two types of metrics in common use: an edit-length distance metric [27] and a kernel based metric [25]. An edit length distance metric morphs one spike train into another through some combination of operations, such as adding a spike, deleting a spike, or shifting a spike. Each operation has a specific cost, and the total cost of turning one spike train into another is defined to be the distance between the two spike trains. A kernel based metric uses a kernel to map each spike train to a function. The Euclidean L^2 distance between two function mapped spike trains can then be measured, and this is taken to be the distance between the two spike trains.

Key to the metric space approach is the notion that distances between spike trains are natural and relatively easy to define. A distance such as the van Rossum metric, or the Victor-Purpura metric, also possesses a very useful property: both contain a free parameter which reflects the fraction of temporal and rate coding associated with the neuron. This free parameter, generally a timescale, is chosen in such a way as to ensure spike trains corresponding to the same stimulus have shorter measured distances than spike trains corresponding to different stimuli. This is based on the reasonable notion that spike trains corresponding to the same stimulus should have a more similar structure to those corresponding to different stimuli.

The free parameter can be seen as 'tuning the metric to the neuron'. In the case of the kernel based method the free parameter alters the shape of the kernel function, implicitly interpolating between coding schemes. For the cost based metric varying the value of its free parameter alters the weighting of the edit operations, also interpolating between coding schemes. The importance of the free parameter lies in the fact that general metric models of neuronal activity can be built which are not dependent on a particular assumption of coding scheme, but rather adapt to the apparent coding scheme of the data itself.

1.5 The discrete approach

Generally, metrics are not commonly used in the analysis of spike trains. Instead spike trains are converted into discrete sequences, which are then analysed using information theory, [7].

To apply the discrete approach one must firstly divide the time axis into a discrete set of time bins, each of width δt say. The time bin size is typically chosen to be of the same length as the absolute refractory period of a neuron. Thus, while spikes can occur at any time, the discrete approach turns the spike train into a *time series*

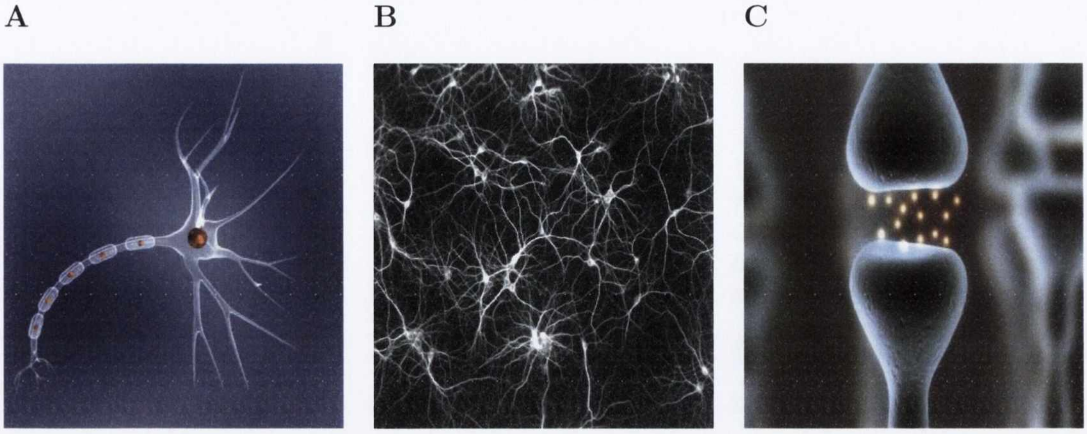

Figure 1.1: A typical neuron, network of neurons and synapse. **A**: Branching to the right are the dendrites, and to the left the axon. Center is the cell body, or soma. **B**: A simplified network of neurons. A neuron is on average connected to 10,000 other neurons, although in some neurons such as the Purkinje this number is over 100,000. **C**: A synapse, where the signal is sent from the presynaptic neuron to the postsynaptic receptor by means of a chemical neurotransmitter. Pictures taken from: `www.mqworks.com, www.anthropic-principle.org and www.defd.colorado.edu` respectively.

of uniform time intervals.

The spike train is then converted into a binary sequence: if a spike time occurs within a certain time bin a 'one' is assigned to that bin, else it is assigned a 'zero' [4, 24, 20, 8]. This binary sequence is then split into a set of short intervals, each of length T say, called *words*. Word length is typically chosen to reflect the behavioural response time of the neuron to the stimulus.

With sufficient data the distribution of words can be estimated. Information theory is based around the calculation of quantities dependent on underlying probability distributions and so can be applied to these spike train probabilities. If, for example, the normalized count of the ith word is given by p_i, then the Shannon entropy of the spike train can be estimated using

$$H(T, \delta t) = -\sum_i p_i \log_2(p_i). \tag{1.1}$$

Spike trains can be recorded from a sensory neuron during stimulation, and also for multiple repetitions of stimuli. The spike train response is known for two regimes, one where the stimulus varies, *the signal response* Y, and one where the stimulus is repeated, *the noise response*, Z. In the discrete approach, the distribution of words is calculated in each case and the difference between the entropy of the signal response and noise response is considered as a measure of the information transmitted:

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= H(Y) - H(Z) \end{aligned} \tag{1.2}$$

where X is the input variable and

$$Y = X + Z. \tag{1.3}$$

This approach introduced information theory into the study of spike trains. However, in addition to the theoretical aspects of the metric space that it ignores, it has a number of practical drawbacks. It suffers from a data sampling problem. For typical discretization size and word length, the number of possible words is extremely large and a typical electrophysiology experiment may not yield enough words to accurately estimate the entropy.

The data used in the original paper was obtained from fly recordings, where spike trains can be recorded for minutes to hours. The behavioural response time of the fly's neuron is fast: of the order of ten milliseconds. In contrast, zebra finch conspecific songs are generally one second in length, with similar behavioural response times. Each behavioural response represents a word, and as the number of possible words, N, is given by

$$N = 2^p \tag{1.4}$$

where $p = (\text{ T } / \delta t)$, for longer words the space becomes increasingly difficult to sample from.

The estimated entropy value depends on the discretization scale and on the word length. There is no obvious principle that can be used to fix values for either. Lastly, the discretization approach ignores the structural similarities of the spike trains. Two similar sequences from a spike train, which would produce a similar synaptic response, will map to different words, where there is no notion of some words being similar and others dissimilar.

A metric space approach is able to address these problems. Data sampling can be made more efficient in a metric-based bin-less approach to estimating the word distribution [30]. The estimated entropy no longer depends on the discretization scale and on word length, since the train is regarded as a continuous time interval. A metric-based bin-less approach can lead to the identification of structural similarities between spike trains; they can be clustered according to their spike train distances.

1.6 Thoughts on the metric space approach

Arguably the primary mathematical difficulty associated with the analysis of spike trains is their natural lack of a coordinate system. With only a set of distances rather than coordinates there is no obvious way of developing probabilistic models of spike trains, that is, spike trains have no obvious associated random variables. That is not to say a metric space approach does not provide useful insights into the neuronal system: given a tuned metric, spike trains can be clustered according to stimulus, and distances between spike trains corresponding to same stimulus presentations can be used to investigate noise models of spike trains. However, with respect to the application of information theory, we are left with few choices: either abandon it, rephrase it in terms of metric quantities, or just apply the discrete approach. During this thesis a rephrasing of the information theory channel capacity formula in terms of metric distances is proposed.

An interesting component of channel capacity formula rephrasing is a probability density model for the overall noise behaviour of a neuron. This is derived from the spike trains of the neuron. Hence, the noise associated with spikes induces distributions on the neurons themselves. The distributions of neurons will be seen to embed the neurons in what is termed a *noise space*. These neuronal spaces, which have associated coordinates, would seem the natural starting point for the application of classical information theory.

1.7 A thesis overview

The next four chapters constitute the areas of research this thesis addresses. A short overview of the concepts and findings of these chapters is now presented. This will hopefully both whet the reader's appetite, and serve as a roadmap for the research material that follows.

Information theory is defined in a vector space, with vector coordinates given by random variables. Now any normed vector space is by definition a metric space, so by using a Euclidean norm to measure distances between vectors we are by definition rephrasing information theory in a metric space. This leads to a different way of thinking about information theory; instead of considering random variables we think of the distances between them.

While the normed vector space of information theory can be rephrased as a metric space, the primary question is how similar this metric space is to the metric space of spike trains. Additive Gaussian noise in information theory has a number of additional properties, the main being that distances measured between noise vectors are χ-distributed. For consistency between spaces, distances measured between spike trains should also follow a χ-distribution. This is found to be the case, allowing metric quantities calculated in the space of spike trains to be substituted for metric quantities that feature in the rephrased channel capacity formula. An example application is given, where the method is applied to spike trains from the auditory forebrain of zebra finch.

Spike trains are often variable, with the same stimulus producing different responses from presentation to presentation. These variations can be thought of as being composed of two different types of noise; variations in the spike times and variations in the spike count. The Victor-Purpora metric can be used to separate these two types of noise. The Victor-Purpura distance metric is an example of an edit-based metric. To measure the distance between two spike trains it performs a set of operations that transform one spike train into another. Each operation has an assigned cost, and the operations consist of adding, deleting, or shifting spikes. The shifting of spikes reflects the temporal variation, termed jitter, that is typically found in spike trains.

The Victor-Purpora metric can be used to extract *only* the temporal differences between spike trains in a given dataset. This allows the distribution in spike time variations to be calculated. This distribution is calculated for a collection of example datasets. For these data, the distributions are not Gaussian but, in most cases, they can be accurately modelled by a hyper-Laplace distribution.

Having uncovered the hyper-Laplacian model of spike train jitter, the question of how to use this result arises. One idea, and the approach adopted here, is to use a combination of the jitter distribution and variations in spike rate to generate artificial spike trains. Again, the Victor-Purpora metric can be used to extract the number of insertions and deletions that take place during a distance computation. From this the average number of insertions and deletions can be found separately, and hence so can the probability of insertion or deletion.

A spike train representative of the dataset is extracted, for instance a template or mean spike train, and its spikes are deleted with the probability of deletion previously found. Next, its spikes are jittered using values drawn from the neurons hyper-Laplace distribution. Spikes are then inserted using a Poisson process, with rate consistent with that of the average spike insertion. Lastly, each artificial spike train is tested, using rejection sampling, to ensure its noise properties could have been drawn from the original neurons χ-distribution with high probability. This entire process can be repeated until the required number of artificial spike trains is generated. Hence, combining the above noise processes provides a generative model for artificial spike trains.

When a metric is used to measure the distance between spike trains corresponding to the same stimulus, the distribution of these distances associates a distribution, termed here as a *noise distribution*, with each neuron. This embeds each neuron in a distribution space known as a *statistical model*, which can be treated as a manifold. Each neuron is a point on the manifolds surface, with coordinates given by the free parameter values of that particular neurons noise distribution. Low dimensional embedding methods can be applied to the set of noise distributions to estimate both the dimension and structure of this neuronal noise manifold. Here, the isomap algorithm is used as we do not know whether this manifold is flat or not, which depends on the curvature tensor of the space. Isomap, applied to the noise space of neurons, provides a technique for exploring models of neuronal noise. Specifically, both the χ-distribution and the variation of noise throughout a network of neurons are visualized.

Appropriate sets of coordinates for the space are investigated. Firstly, the space is found to be strongly two dimensional, meaning that two coordinates are needed to specify each neuron. The average distance between same stimulus spike trains, which reflects the noise associated with a neuron, is found to be a reasonable choice, as is the standard deviation of the same stimulus distances. Lastly the distribution of firing rates throughout the space is plotted. Here, approximately eight percent of the neurons, those with elevated firing rates, are also seen to have the greatest level of noise, and are generally confined to a particular arm of the graph.

Data used here consists of extracellular spike trains from the large extracellular zebra finch dataset made available on the Collaborative Research in Computational

Neuroscience database [40]. Data is obtained from 455 neurons throughout eight regions: L, L1, L2a, L2b, L3, Mld, CM and OV, of zebra finch auditory forebrain during playback of conspecific song. Neuronal noise can also be visualized throughout the above networks of neurons. Strong variations between networks are not observed, with the exception of OV which is in general confined to the same bottom right arm which displays increased levels of noise and firing rates.

Chapter 2

The channel capacity of a neuron

2.1 Introduction

The space of spike trains is a metric space [18, 31]. While distances representing dissimilarity can be defined between spike trains, it is not known how a mathematical integration measure can be defined on the space. As a consequence, there is no obvious way of calculating probability distributions on the space of spike trains.

As has been discussed, spike trains can be discretized, converted into series of zeros and ones. These sequences can then be divided into groups and the probability of one such group appearing estimated. However in doing this we lose the notion of similarity between trains, slightly different sequences will map to completely different numbers. Also the spikes themselves are part of a continuous time process in which, respecting the basic constraints of refractory periods, spikes can occur at any time and not in specific time bins. We are left with no obvious way of associating probabilities with spike trains.

Information theory can be applied to processes described by probability distributions. It is phrased in terms of random variables, that is, a vector space. Therein lies the difficulty in applying information theory to spike trains, we have no such random variables. However, we can make use of the relationship between a vector space and a metric space: any normed vector space is by definition a metric space with a Euclidean distance metric. It should be noted the converse does not hold, a metric space does not imply a vector space exists, that is, it does not imply coordinates can be associated with each point in the metric space.

Conventionally, artificial coordinates such as binary sequences are associated with spike trains and hence probabilities can be associated with trains, allowing for the application of information theory [20]. Here the opposite approach is taken, the vector space of information theory is rephrased in terms of its underlying metric quantities. In information theory, L^2 distances between Gaussian vectors follow a χ-distribution. These Gaussian vectors exist in the vector space of information theory, and the distances in the metric space. The χ-distribution then forms an observable link between the spaces. The idea is to test if L^2 metric distances between same-stimulus spike trains also follow a χ-distribution. If so, then these distances can also be seen as measuring the length of hypothetical Gaussian vectors, and substituted for their corresponding metric quantities in information theory.

2.2 The van Rossum metric

We now introduce the first of two metrics used throughout this thesis: the van Rossum metric. This metric is both intuitive and easy to apply. It firstly maps a spike train to a function, then computes the L^2 distance between two such functions. As will be seen, the L^2 distance property of the metric will introduce an Euclidean structure to the space of spike trains, which can then be compared with the Euclidean metric space of information theory.

A spike train consists of a set of spike times $\mathbf{u} = \{u_1, u_2, .., u_n\}$. In order to calculate the van Rossum metric, the spike train is firstly filtered: it is mapped to

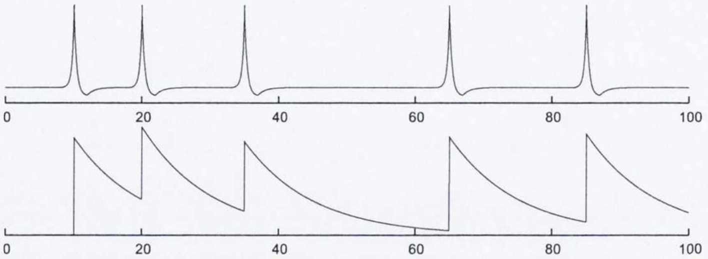

Figure 2.1: The mapping of a spike train to a set of exponential functions.

a function of time, $f(t;\mathbf{u})$:

$$\mathbf{u} \mapsto f(t;\mathbf{u}) = \left[\sum_{i=1}^{n} h(t-u_i)\right] . \tag{2.1}$$

where $h(t)$ is a kernel often taken as [25, 26]

$$h(t) = \begin{cases} 0 & t < 0 \\ e^{-t/\tau} & t \geq 0 \end{cases} \tag{2.2}$$

and τ is a timescale parametrizing the metric.

The L^2 metric on the space of real functions then induces a metric on the space of spike trains, specifically, if $\mathbf{u}$ and $\mathbf{v}$ are two spike trains, then the distance between them is

$$d(\mathbf{u},\mathbf{v}) = \sqrt{\frac{1}{\tau}\int [f(t;\mathbf{u}) - f(t;\mathbf{v})]^2\, dt}. \tag{2.3}$$

As in kernel density estimation, [22], the precise shape of the kernel is not thought to be significant. However, the timescale is important. A standard method has been developed for calculating the optimal timescale [15] and is discussed in the next section.

2.3 The timescale of the van Rossum metric

The van Rossum metric contains a free parameter, τ. This free parameter controls how quickly the exponential tail attached to each spike decays, and has important implications for the encoding of information in spike trains: by determining the width of the kernel assigned to each spike, different values of τ interpolate the metric between rate and temporal coding schemes. At one extreme, as τ tends to zero, the metric reduces to coincidence detection, counting only non-coincident spikes, as seen through

$$\lim_{\tau \to 0} d^2(\mathbf{u}, \mathbf{v}) = \frac{1}{\tau} \int_0^\infty (f^2(t; \mathbf{u})(t) - f^2(t; \mathbf{v})(t))\, dt = \frac{M+N}{2} \tag{2.4}$$

where here it is assumed that $f(t; \mathbf{u})$ contains M spikes and $f(t; \mathbf{v})$ contains N spikes.

In the other extreme, as τ tends towards infinity, the metric approximately measures the difference in the total spike count:

$$\lim_{\tau \to \infty} d^2(\mathbf{u}, \mathbf{v}) = \frac{1}{\tau} \int_0^\infty (Me^{-\frac{t}{\tau}} - Ne^{-\frac{t}{\tau}})^2\, dt = \frac{(M-N)^2}{2}. \tag{2.5}$$

Thus the value of τ can provide insight into the coding scheme of the neuron under examination: rate or temporal, or some combination. This often underestimated property allows for an agnostic approach to the problem of choosing a neurons coding scheme. In theory metric models of neuronal activity make no assumption as to whether single spike trains encode information on a rate or temporal basis; it is only through the estimation of τ that the coding scheme is decided upon. This can also be generalized to a network of neurons: a value of τ can be found for each neuron, allowing the distribution of coding schemes throughout the network to be viewed through the kernel density estimation of τ for the network.

2.3.1 Estimating the timescale

It is crucial that τ is chosen so that distances between spike trains corresponding to the same stimulus are smaller than distances measured between the spike trains of different stimuli. An algorithm for this exists for the Victor-Purpora metric, and can be adapted to the van Rossum metric [15]. The method for finding τ is now outlined.

Spike train data consisting of groups of spike trains corresponding to the same stimulus must be used. For instance, a particular auditory neuron may be stimulated to fire a spike through presentation of a conspecific song stimulus. If this song is played ten times, then the resulting ten spike trains form a same-stimulus group. In typical zebra finch datasets twenty different conspecific songs are played ten times each, resulting in twenty groups with ten spike trains in each group.

Let N be a *confusion* matrix. N_{ij} denotes the number of responses from stimulus i which are closest, on average, to responses from stimulus j. For an ideal dataset we should have $N_{ij} = 0$ unless if $i = j$. The following algorithm is used to find optimal τ.

N, a $n \times n$ matrix, is initially filled with zeros. Remove a spike train from its stimulus group. Compute its average distance from both its own, and every other group. The distance from spike train i to the kth cluster C_k is given by:

$$d_k = \left(\frac{1}{|C_k|} \sum_{s \in C_k} d(i,s)^z \right)^{1/z}. \tag{2.6}$$

Here z is used as a robustness exponent, generally chosen as -2, which helps give a smaller weighting to dataset outliers. Store the list of distances associated with the ith spike train. This list of distances must now be searched for its minimum value, say the pth entry for example, then increment the N_{ip} entry of the matrix by +1.

Remove another spike train and repeat the same process. Do this for every spike train in the set, updating the matrix. Compute h, where h is termed the transmitted information defined by:

$$h = \frac{1}{N_{tot}} \sum_{ij} N_{ij}(\log_2(N_{ij}) - \log_2(\sum_i N_{ij}) - \log_2(\sum_j N_{ij}) + \log_2(n)). \tag{2.7}$$

and N_{tot} defines the total number of spike trains.

Lastly we must maximize h with respect to τ. The method used here is grid search: increment the value of τ and repeat all the above steps to form a new matrix, each time storing the value of τ and h. When all values of h for τ within a certain range have been found, select the value of τ for which h is a maximum. This is the optimal value of τ for the neuron.

For perfect clustering the value of h equals $\log_2(C)$, [], where C denotes the number of stimuli response classes. For instance in this case it would be $\log_2(20)$ since there are 20 song stimuli. It can be helpful to normalize h and compute $\tilde{h}$, where $\tilde{h} = h/\log_2(n)$, yielding maximum h values of one. Neurons that show good clustering typically have values of h in the region of 0.7. It should be noted that h, termed the transmitted information, is a measure of how well the spike trains are clustered according to stimulus.

It is found that a simple grid search procedure is a practical way of finding an optimal τ. While for large timesteps of τ, that is of order 0.01, the value of $\tilde{h}$ tends to increase or decrease monotonically, when more precise increments are used $\tilde{h}$ is found to vary noisily, making a global value of τ difficult to find. As a consequence, faster search procedures are discarded for the dependable grid search.

2.4 Information theory

Many events, such as the outcomes of a race or rolling a dice, have random outcomes that are best described using probability theory. For instance in a horse race each horse has a certain probability of winning. If each horse has the same odds of winning then predicting the winning horse is at its most difficult, that is, the race is in a state of maximum uncertainty. Information theory addresses the idea of quantifying the uncertainty associated with a probabilistic event [9].

The uncertainty associated with a horse race, where each horse x_i has probability

$p(x_i)$ of winning is given:

$$H(X) = -\sum_{i=0}^{n} p(x_i) \log_2 p(x_i). \tag{2.8}$$

Here the random variable X is clearly discrete and hence so is the entropy, which is measuring the uncertainty associated with X.

The concept of entropy can be extended to that of conditional entropy. If we have two random variables X and Y then we may ask 'what is the uncertainty in X given that we know the value of Y'. This is expressed as

$$H(X|Y) = -\sum_{i,j} p(x_i, y_j) \log_2 p(x_i|y_j). \tag{2.9}$$

Another closely related concept is that of mutual information. This measures the mutual dependence of one random variable on another in terms of entropy:

$$I(X,Y) = H(X) - H(X|Y) \tag{2.10}$$

or equivalently:

$$I(X,Y) = \sum_{i,j} p(x_i, y_j) \log_2 \frac{p(x_i, y_i)}{p_1(x_i)p_2(y_j)} \tag{2.11}$$

where p_1 and p_2 are the marginal distributions of X and Y respectively. From the entropy definition of mutual information we can see that $I(X,Y)$ measures the decrease in uncertainty in X through knowing the outcome of Y, which can be interpreted as the information 'passed' between X and Y.

Should the random variables be continuous rather than discrete then the above equations can be altered accordingly, replacing probability distribution functions with probability density functions and integrating over the random variables. For instance, the continuous version of entropy, also called differential entropy, is

$$h(X) = -\int_A f(x) \log_2 f(x)\, dx \tag{2.12}$$

where X has probability density function $f(X)$ with support A. Measure theory can be used as a means of relating discrete and continuous entropy, [?].

A number of important distinctions between discrete and differential entropy arise. Unlike discrete entropy, continuous entropy can take negative values. Also it is not invariant under change of variables, meaning it is not absolute. As entropy is used to quantify uncertainty, the lack of invariance under change of variables makes comparing differential entropy values difficult, and similarly it is hard to understand the meaning of negative entropy.

Again we can see that information theory is a theory based on probability distributions or densities. But what is the distribution of a spike, or spike train? Without artificially deconstructing the train through binning, [4], no such distributions have been found to exist. While sets of probability distributions, probability vectors, live in a vector space, there exists the opportunity to phrase these vectors in terms of an Euclidean metric space by just introducing an Euclidean metric. The possibility exists then to rephrase this normed vector space in terms of metric space quantities. In practical terms this means rephrasing probability vectors in terms of the distances measured between such random vectors.

2.5 Gaussian channel capacity theory

Gaussian channel capacity theory quantifies the maximum mutual information that can be communicated with minimal error by a continuous time-discrete channel with Gaussian noise. Typically many of the theorems in information theory are based on taking the limits of mathematical functions, [9], hence since these limits in practice are not achievable the error can be minimized, but not eliminated. The signals themselves are continuous, but input to the channel is discrete. The channel at time i is modelled by a signal X_i, additive noise Z_i and output Y_i, so

$$Y_i = X_i + Z_i \tag{2.13}$$

where the noise are independent identically distributed (iid) Gaussian, and $f_{Z_i}(z) = f_N(z, \sigma)$. The time at which input to the channel takes place is discrete, but X_i itself is a continuous random variable [21].

A code consists of a set of codewords, sufficiently well separated to make them distinguishable despite the noise. Of course, since the noise is additive, the capacity of the channel is infinite unless there is some limit on X_i. In the traditional application of this theory, which can be typified by radio communications, the most convenient constraint is a power constraint: for any codeword $(x_1, x_2, \ldots, x_n)$ it is required that

$$\frac{1}{n}\sum_{i=1}^{n} x_i^2 \leq \nu^2. \tag{2.14}$$

The central result of channel capacity theory is that the capacity of a Gaussian channel is:

$$C = \frac{1}{2}\log_2\left(1 + \frac{\nu^2}{\sigma^2}\right) \text{ bits per time unit.} \tag{2.15}$$

This is a bound on the maximum mutual information: information flow though the channel cannot exceed C with minimal error. This bound, C, is calculated using the distribution which leads to the largest information flow rate: a Gaussian distribution with zero mean and with variance ν^2. A more detailed proof of this result can be found here: [?]. Calculating the actual information flow rate would require knowing the distribution of X_i, which we will see is not straight forward.

2.6 From a vector to a metric space

Say we have a vector of k iid Gaussian random variables with zero mean:

$$\mathbf{Z} = (Z_1, Z_2, \ldots, Z_k) \tag{2.16}$$

where

$$f_{Z_i}(z_i) = f_N(z_i; \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-z_i^2/2\sigma^2}. \tag{2.17}$$

The length of such a vector is

$$|\mathbf{Z}| = \sqrt{Z_1^2 + Z_2^2 + \ldots + Z_k^2} \tag{2.18}$$

where $|\mathbf{Z}|$ follows what is called a non-standard χ-distribution. In statistics significantly more attention is paid to the χ-squared distribution due to its role in hypothesis testing. The χ-squared distribution is the distribution followed by a random variable which is itself the sum of squared random variables, each following a standard normal distribution with mean zero and variance one. The standard χ variable is distributed according to the square root of this summation of random variables. The non-standard χ is just a slight alteration, where the underlying variables are drawn from a normal distribution with zero mean, rather than a standard normal distribution. The non-standard χ-distribution has the following probability density function:

$$f_{|\mathbf{Z}|}(|\mathbf{Z}| = x) = f_\chi(x; k, \sigma) = \frac{1}{\sigma^k 2^{k/2-1}\Gamma(\frac{k}{2})} x^{k-1} e^{-x^2/2\sigma^2}. \tag{2.19}$$

One derivation for this density can be found in the appendix: [?].

Consider now the hypothetical situation where there are k random variables, that is k coordinates, for a certain length L of spike train. Then a spike train would be seen as

$$\mathbf{Y} = (Y_1, Y_2, \ldots, Y_k) \tag{2.20}$$

where $\mathbf{X} = (X_1, X_2, \ldots, X_k)$ is the input vector to the channel representing the information about the stimulus, $\mathbf{Z} = (Z_1, Z_2, \ldots, Z_k)$ is the additive Gaussian noise vector in the channel and the variables follow the relationship $Y_i = X_i + Z_i$.

Consider the experimental set-up where the response to a repeated stimulus is recorded and the variation of the responses is taken to be noise. It is, of course, a very strong assumption, but assuming the variables in $\mathbf{Z}$ were additive iid, then by taking the distance between two $\mathbf{Y}$ vectors corresponding to the same stimulus we are computing

$$d(\mathbf{Y}, \mathbf{Y}') = \sqrt{(Y_1 - Y_1')^2 + (Y_2 - Y_2')^2 + \ldots + (Y_k - Y_k')^2} \tag{2.21}$$

that is:

$$d(\mathbf{Y}, \mathbf{Y}') = \sqrt{(X_1 - X_1' + Z_1 - Z_1')^2 + \ldots + (X_k - X_k' + Z_k - Z_k')^2}. \tag{2.22}$$

Now, since only spike trains corresponding to the same stimulus are being examined:

$$\mathbf{X} = \mathbf{X}' \tag{2.23}$$

allowing the distance to simplify to

$$d(\mathbf{Y}, \mathbf{Y}') = \sqrt{(Z_1 - Z_1')^2 + \ldots + (Z_k - Z_k')^2} \tag{2.24}$$

hypothetical or equivalently:

$$d(\mathbf{Y}, \mathbf{Y}') = d(\mathbf{Z}, \mathbf{Z}'). \tag{2.25}$$

So assuming the coordinate representation of a spike train, by taking the distance between two same-stimulus spike trains we are effectively finding the length of a Gaussian vector. This distance should in turn follow a χ-distribution.

It is easy to show that if Z and Z' are iid Gaussian variables with variance σ^2 their difference is Gaussian with variance $\sigma_d^2 = 2\sigma^2$ and, hence:

$$f_{Z-Z'}(z - z' = \delta) = f_N(\delta; \sigma_d). \tag{2.26}$$

Similarly for two Gaussian vectors $\mathbf{Z}$ and $\mathbf{Z}'$:

$$f_{|\mathbf{Z}-\mathbf{Z}'|}(|\mathbf{z} - \mathbf{z}'| = \zeta) = f_\chi(\zeta; k, \sigma_d). \tag{2.27}$$

It is proposed here to turn this on its head and work back from an assumption that the distances have the χ-distribution $f_\chi(\zeta; k, \sigma_d)$. More precisely, while coordinates for the space of spike trains have not been found, it is possible to calculate distances. In the analogy, it is possible to compute ζ, but $\mathbf{Z}$ is not defined. Of course, k in general is not an integer. While the derivation of the χ-distribution involves the summation of k Gaussian variables, once derived the density function places no integer constraint on k, it need only be a positive real number. Therefore k is not really the dimension of a spike train, but it is more like an average dimension, the average number of coordinates that would be required to describe length L fragments of spike train for that cell; as estimated by examining the noise.

2.7 Gaussian channel capacity theory and spike trains

Channel capacity theory hinges on the calculation of quantities which can be viewed as distances. It is possible to rewrite the formula for C in terms of distance quantities. It is noted above that $\sigma_d^2 = 2\sigma^2$, the other quantity which appears in the formula for C is the power constraint ν^2. In the classical Gaussian channel capacity theory a limit is imposed on the variance of a code word. The variance of the signal is in general not known, hence a power constraint must be used instead, such as an amplitude constraint or a frequency constraint. Here, while a number of possible power constraints could be applied to spike trains, we don't need to use these as we can calculate the variance of the signal itself, notated ξ_d^2, the variance in the distribution of distances between different fragments, all same length, of spike trains corresponding to different stimuli. This is related to the variance in Y_i, rather than the X_i we need. Of course, since Z_i and X_i are independent, using the additive property of variance the corresponding X_i-related quantity is

$$\nu_d^2 = \xi_d^2 - \sigma_d^2. \tag{2.28}$$

It remains to relate ν_d^2 to ν^2, the variable that actually features in the original capacity formula. This is not difficult but two differences between them need to be addressed. Firstly, the distance variance is related in the usual way to the variance of the random variable by

$$\xi_d^2 = 2\xi^2. \tag{2.29}$$

The second difference is that ν_d^2 relates to the variance of X_i whereas ν^2 is a constraint on the variance over an individual codeword. However, it can be seen from the derivation of the channel capacity in, for example [9], that this distinction does not matter.

2.8 A metric capacity

In summary, the distances between the output and noise vectors in the Gaussian channel define an associated metric space. Relations have been derived to relate the variance of the distances to the variance of the implied k coordinate variables. If the noise coordinates are additive Gaussian then the noise vectors in the metric space are χ-distributed. Specifically the noise variance σ^2 and the output variance $\sigma^2 + \nu^2$ can be calculated from the corresponding inter-fragment distance variances, σ_d^2 and $\xi_d^2 = \sigma_d^2 + \nu_d^2$.

The equivalent channel capacity in the related metric space is then

$$\begin{aligned} C &= \frac{1}{2}\log_2\left(1 + \frac{\nu_d^2}{\sigma_d^2}\right) \text{ bits per dimension} \\ &= \frac{1}{2}\log_2\left(\frac{\xi_d^2}{\sigma_d^2}\right) \text{ bits per dimension.} \end{aligned} \tag{2.30}$$

The channel capacity in bits per second is

$$C = \frac{k}{2}\log_2\left(\frac{\xi_d^2}{\sigma_d^2}\right) \text{ bits per second.} \tag{2.31}$$

where k is the average number of variables per length L of spike train.

2.9 The capacity of a neuron?

While a metric space consistent with the vector space of information theory has been obtained, there is no reason to assume that the properties of this metric space are consistent with the metric space of spike trains. Indeed, additive Gaussian channel capacity theory predicts the overall behaviour of distances in the metric space of information theory. As a consequence, the metric space of spike trains can also be searched for this behaviour, to check if channel capacity theory can be applied to the space of spike trains. First, the distances between spike trains corresponding to the same stimulus must behave as if they were χ-distributed. Second, as a spike train is treated as equivalent to a length L of k variables, a decrease in L should lead to a linear decrease in k. Lastly, as each spike train is treated as the sum of k normal distributions with mean zero and variance σ^2, the value of the variance should be roughly constant.

2.10 Dataset used

Zebra finch spike trains are used to test for common properties of the two metric spaces. The dataset consists of sets of spike trains from 24 neurons. Each set of spike trains is recorded from a site in field L of the auditory fore brain of anesthetized zebra finch during playback of 20 conspecific songs. Each song is repeated ten times, to give a total of 200 spike trains. These spike trains, and the experimental conditions used to produce them, are described in [19, 32]. Data were collected from sites which showed enhanced activity during song playback. Of the 24 sites considered here, six are classified as single-unit sites and the rest as consisting of between two and five units [19]. The average spike rate during song playback is 15.1 Hz with a range across sites of 10.5-33 Hz. For the purposes of this data an average metric timescale value of $\tau = 12.8$ ms is chosen, following [15].

2.11 Three samples from the dataset

The primary assumption is that the distances between noise responses will have a χ-distribution. For the data here, 900 distances are calculated for each cell; that is ten choose two, or 45, fragment pairs for each of the twenty songs. It will be seen below that the quantities required for calculating the capacity are estimated by considering the distribution of inter-fragment distances for different fragment lengths. Here, though, for the purpose of discussing how well the noise responses are modelled by a χ-distribution, the fragments are chosen to be one second long. For each cell the distribution is approximated using kernel density estimation with a Epanechnikov kernel whose bandwidth is determined by least-squares cross-validation [22]. The actual process of kernel density estimation is outlined in the appendix: [?]. The L^2-error between this curve and the χ-distribution is then calculated; the appropriate χ-distribution is chosen by using the moments to estimate k and σ_d, as described below. The error of the 24 sites lies in the range $[0.022, 0.102]$, with average 0.054

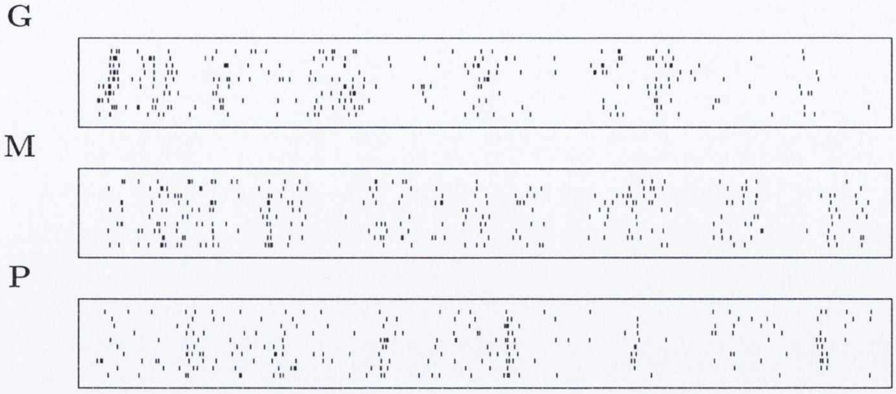

Figure 2.2: Example raster plots. These are example raster plots for the **G**, **M** and **P** sites. Each shows the ten responses to the same song, one of the twenty used to make the whole dataset. The spike trains are one second long and start at song onset.

and standard deviation 0.0196.

Of the 24 sites examined, three have been selected, based on their L^2-errors, to demonstrate the detailed application of the theory. These are labelled **G**, **M** and **P**; abbreviating good, middling and poor: raster plots for these three sites are shown in Fig. 2.2 and a comparison between the noise response distribution and the χ-distribution is given in Fig. 2.3. Additional processing was required for one of the 24 sites. This site appeared to give an unusual result, further inspection revealed that three spike trains were anomalous, showing a non-biological firing pattern, that is, were the result of electrophysiological error. These specific spike trains were removed and the site became unremarkable; this site in not one of the three featured sites.

2.12 Evaluating the channel capacity formula

The formula for the channel capacity C requires three quantities, the two variances ξ_d^2 and σ_d^2, and k, the average number of dimensions per length L of spike train. This means that estimators are required for these quantities. Here, they are calculated from the moments of the distribution of distances between fragments of spike train.

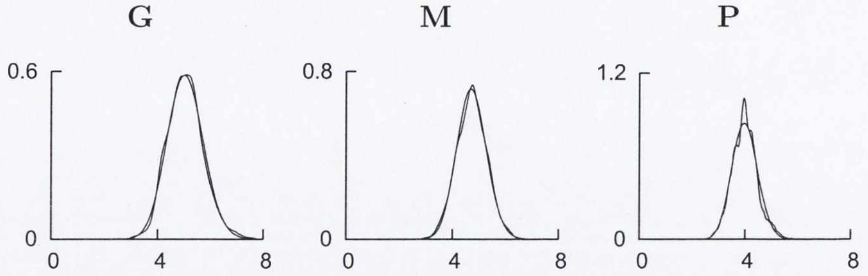

Figure 2.3: A comparison between the distance distributions for the noise responses and the corresponding χ-distributions. Kernel density estimation is used to generate the probability density distribution of the distances, for the sites **G**, **M** and **P**. In each case, this is compared to the χ-distribution, where the parameters of the distribution are estimated from the moments.

Rather than choosing a particular length of spike train, the moments are calculated for a range of lengths and curve fitting is used to extract a robust estimate.

2.12.1 Calculating k

Calculating k requires that the distribution of noise distances is fitted to a χ-distribution. To do this the second and fourth non-central moments of a χ-distribution are used. For the distribution $f_\chi(x; k, \sigma)$

$$\begin{aligned} \langle x^2 \rangle &= \sigma^2 k \\ \langle x^4 \rangle &= \sigma^4 k(k+2). \end{aligned} \tag{2.32}$$

These equations are then solved for k:

$$k = \frac{2\langle x^2 \rangle^2}{\langle x^4 \rangle - \langle x^2 \rangle^2}. \tag{2.33}$$

Let $k(L)$ be k calculated, in this way, for fragments of length L.

2.12.2 Calculating ξ_d^2 and σ_d^2

To calculate ξ_d^2 and σ_d^2 two types of variation need to be considered, the variance of the noise for σ_d^2 and of the signal for ξ_d^2. For a given fragment length L, a set of inter-fragment distances is calculated. This set is composed of two parts: the distances between the noise responses, the fragments responding to the same stimulus and the distances between the signal responses, that is the responses to different stimuli.

By linearity, the variation of the signal distances should be

$$\xi_d^2(L) = k\xi_d^2 \tag{2.34}$$

and the variation of the noise distances:

$$\sigma_d^2(L) = k\sigma_d^2. \tag{2.35}$$

Here, $\xi_d(L)^2$ and $\sigma_d(L)^2$ notate the variances of the distributions of signal and noise distances respectively, measured between length L fragments of spike train. There are k coordinates per length of spike train. Note also that if only one coordinate were needed to represent the train, then the variances of the distributions of signal and noise distances would reduce to ξ_d^2 and σ_d^2 respectively, simply variances and not functions of length.

As both $\xi_d(L)^2$ and $\sigma_d(L)^2$ should vary linearly with L, the slope of the line formed using least squares estimation of $\sigma_d(L)^2$ against L will yield an average variance per unit length. The calculation of average values of $\xi_d(L)$ and $\sigma_d(L)$ is illustrated for three example sites in Fig. 2.4 and the average value of $k(L)$ for the same three sites in Fig. 2.5.

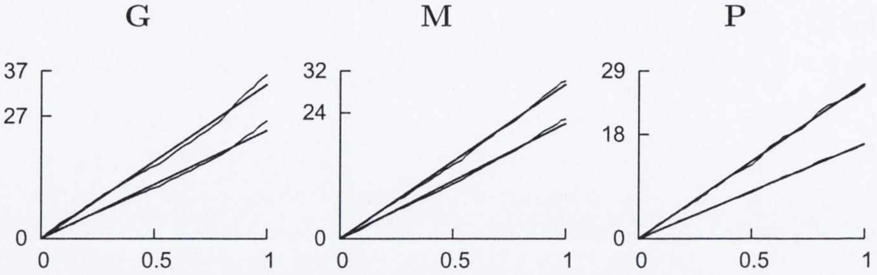

Figure 2.4: The variance for signal and noise plotted against fragment length. In each case the upper curve represents $\xi_d(L)$, the variance of the signal distances against the fragment length L, for sites **G**,**M** and **P** respectively. The lower curve represents $\sigma_d(L)$, the variance of the noise distances against the fragment length L, for the same sites. In each case the behavior is well approximated by a fitted line with zero intercept.

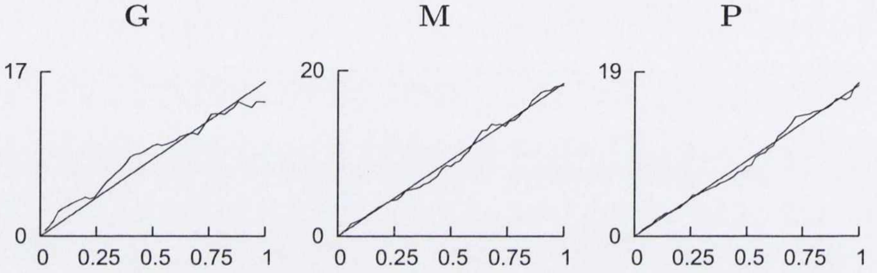

Figure 2.5: Plots of $k(L)/2$. For **G**, **M** and **P** the moments of the distance distributions have been used to calculate $k(L)/2$ against fragment lengths, L, up to one second. The straight line represents the predicted least squares fit of the data.

2.13 Testing the noise model

2.13.1 The Anderson-Darling test of the χ-distribution

The primary test for consistency between metric spaces centers on testing whether or not distances between spike trains corresponding to the same stimulus follow a χ-distribution. It is important to test this model since it relies on the original assumption of additive Gaussian noise. Actually, it can be seen from the comparison of the kernel density estimated distribution and the χ-distribution, Fig. 2.5, that it seems to work very well. It is, however, always difficult to make a more direct, quantitative, evaluation of the appropriateness of a statistical model. Usually the best that can be done is to attempt to significantly rule out the distribution using a statistical test and to show that this attempt fails. This is what is done here.

The Anderson-Darling goodness-of-fit test was chosen to hypothesis test the model [42]. The Anderson-Darling test relies on comparing a test statistic to a table of critical p-values and such tables are only available for a few specific distributions. The χ-distribution used in this thesis is not one of these and p-values are estimated here using a simple simulation.

The Anderson-Darling test defines a statistic calculated on a set of outcomes $\{X_1, X_2, \ldots, X_n\}$ ordered with $X_1 \leq X_2 \leq \ldots \leq X_n$. To test whether this data significantly differs from a distribution with cumulative $F(x)$ the statistic

$$A^2 = -n - S \tag{2.36}$$

is calculated where

$$S = \sum_{k=1}^{n} \frac{2k-1}{n} \left[\ln(F(X_k)) + \ln(1 - F(X_{n+1-k})\right]. \tag{2.37}$$

In the case being considered here, the distribution is a χ-distribution with k and σ fitted to the dataset. To test that this is a good model, the statistic A^2 is calculated for the experimental data, the distribution of distances. This is then compared to the distribution of the statistic itself for data drawn according to the hypothesized cumulative distribution F. If the experimental value of the test statistic is greater than 0.95 of the values of the test statistic for data drawn according to F then the experimental data is said to differ significantly from F. In fact the result is usually phrased in terms of the fraction of simulated values larger than the experimental value, this is called the p-value and so a distribution is significantly ruled out if the corresponding p-value is less than 0.05.

Of course, a positive result in this context would be the failure to significantly rule out the distribution. This does not show that some other distribution would be still more suitable, however statements of that sort would be difficult to make since they rely on some overall distribution of possible statistical models. Nonetheless, the Anderson-Darling test is considered a useful and sensitive test of hypothesised statistical models [14].

Since there is no theoretical calculation of the distribution of the test statistics available and so the experimental value of the statistic is compared to a collection of values calculated for simulated data. For each site, a fragment length is chosen to make the k-value close to being an integer, the granularity of the data makes this an imperfect process but, using an integer makes it easy to calculate simulated data. If $[k]$ is the integer which is nearly equal k, then $|[k] - k|$ has a range of $[0.0003, 0.1661]$ with average 0.045428. The Box-Muller transform is used to generate a $[k]$-vector of normal random variables with standard deviation σ_d, where σ_d is the standard deviation of the distances. The length of this vector has a χ-distribution. Applying this 900 times generates a sample of points $\{X_1, ..., X_{900}\}$ from the χ-distributed random variable. The Anderson-Darling method is then used to compute a theoretical test statistic A^2 from this set. This process is carried out 1000 times giving a distribution of values of the test statistic. The p-value for the experimental result is the fraction of these simulated values of the test statistic which are greater than the experimental value.

Using the Anderson-Darling method, the real test statistic for each cell was found and the corresponding p-values computed. The null hypothesis is that the distances follow a χ-distribution $f_\chi(x; k, \sigma_d)$, where k and σ_d are calculated from the moments of the distribution. At the significance level of 0.05 there was insufficient evidence to reject the null hypothesis for 23 out of the 24 sites. This means that the noise model presented here passes the Anderson-Darling test for all but one site. Out of the 23 cells that pass, the p-values are in the range of $[0.055, 0.988]$, with average p-value of 0.523 and standard deviation 0.28. One site does fail, its p-value is actually indistinguishable from zero, but this is not surprising for experimental electrophysiological data.

To test the sensitivity of the test, it was applied using the wrong distribution. For a site with value k the hypotheses that the data is modelled by a χ-distribution with $[k] \pm 1$ and $[k] \pm 2$ has been tested. It was possible to rule out the distributions with $[k] + 1$ for all sites and with $[k] - 1$ for all but one site and in that case the p-value was low, 0.069. For $[k] \pm 2$ the p-value was zero in every case, in other words, in each case, the value of the statistic for the experimental data was larger than all 1000 values of the simulated data.

2.13.2 Variation of k and σ with spike train fragment length

Both the standard deviation of the Gaussian distributions underlying the χ-distribution, σ, and the dimension of the noise vector, k, can be plotted as a function of the spike train length. Again, the analogy here being that as the fragment length is reduced so is the dimension of the fictitious vector of Gaussian coordinates. However, while k, the vector dimension should decrease linearly with decreasing fragment length, the standard deviation of the Gaussians should not, as it is independent of fragment length. This is of course not entirely true as the there exists some correlation with the stimulus intensity and neural activity, but as can be seen for the three cells the standard deviation remains roughly constant, Fig. 2.6. As can also be seen, though not perfectly, k decreases linearly with decreasing fragment length, as was seen in Fig. 2.5. Least squares fitting can be used to find an average value of k for each cell.

2.14 Information channel capacities

Kernel density estimation was used to plot the probability distribution of the capacities of the 24 sites, see Fig. 2.7. The average channel capacity was found to be 7.36 bits per second, with a standard deviation of 4.96 bits per second. The values for the **G**, **M** and **P** sites are given in Table 2.1.

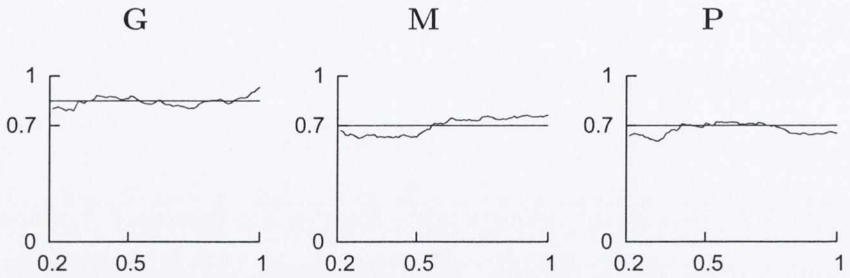

Figure 2.6: Plots of the variation of σ for the sites **G**, **M** and **P**. Each is fitted with a line of zero slope. As the length of spike train is decreased, the variance of the noise is seen to fluctuate around a line of zero slope. This is consistent with the hypothesis of spike train noise being represented by k Gaussian variables of identical variance; as variables are removed from the L of train being examined the underlying variance of the remaining variables remains the same.

Cell	L^2 error	$\left\langle \frac{\xi_d^2(L)}{L} \right\rangle$	$\left\langle \frac{\sigma_d^2(L)}{L} \right\rangle$	k	C	Cs^{-1}	p-value
G	0.037	34	23.83	31.948	0.2564	8.189	0.324
M	0.041	29.4	21.94	36.9	0.2111	7.791	0.841
P	0.09	26.7	16.35	34.84	0.3538	12.333	0.174

Table 2.1: Numerical values for the featured sites. The column marked C gives the capacity for each time unit, that is, for a time interval of length $1/k$; the capacity per second is given in the column marked Cs^{-1}. The designation of good, middling and poor were made based on the L^2 error in the χ distribution; this does not appear to affect the capacities in the same way and the Anderson-Darling p-values show that **M** cell is better modelled by the χ-distribution than the **G** cell, even if the L^2 error is greater.

It is interesting to compare the channel capacity to the transmitted information. This was done for 24 cells, with a demonstratively linear relationship, seen in Fig. 2.8, with coefficient of determination $R^2 = 0.68$.

2.15 Discussion

A novel method for calculating the channel capacity of spike trains has been presented. This is motivated by the idea that the space of spike trains can be most naturally thought of as a metric space. The new method appears to work well for

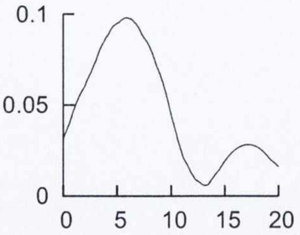

Figure 2.7: A plot of the probability density against measured capacity for the 24 sites. Kernel density estimation has been used to estimate a smooth distribution for the channel capacity. Interestingly, there is a bimodal split between four high capacity sites and 20 with lower capacity. Another notable feature is that the curve is not close to approaching zero at zero capacity. Kde is known to overspill boundaries and introduce biases at edges, hence the significance of the result of non zero capacity at the edge remains unclear.

the example dataset. The approach taken was to proceed as if there is a coordinate space and to then translate the calculation into distance-based quantities, giving a formula for the channel capacity on the metric space. Obviously, this approach could be more fully realized by giving a version of information channel capacity theory on a metric space which makes no mention of coordinates.

One difficulty with applying information theory to neuroscience is that the paradigm underpinning information theory is quite different from the situation which holds for electrophysiological data. For example, channel capacity describes a discrete set of stimuli which are encoded in a discrete set of signals, signals which are in turn embedded with noise in a continuous space of outputs. This is hardly the situation that holds in the sensory pathways. Moreover, the theorems which are proved for information theory often become principles or techniques when applied in this less well-defined context.

These issues are nicely summarized in, for example, [11], where rate-distortion theory is suggested as the correct information theory framework for the sort of data discussed here. However, applying rate distortion theory to sensory electrophysiological data is quite a challenge, it requires a well-defined stimulus space and a relevant rate distortion function. It seems likely, though, that the technique suggested in this thesis, the redefining of coordinate-based information quantities as metric space ob-

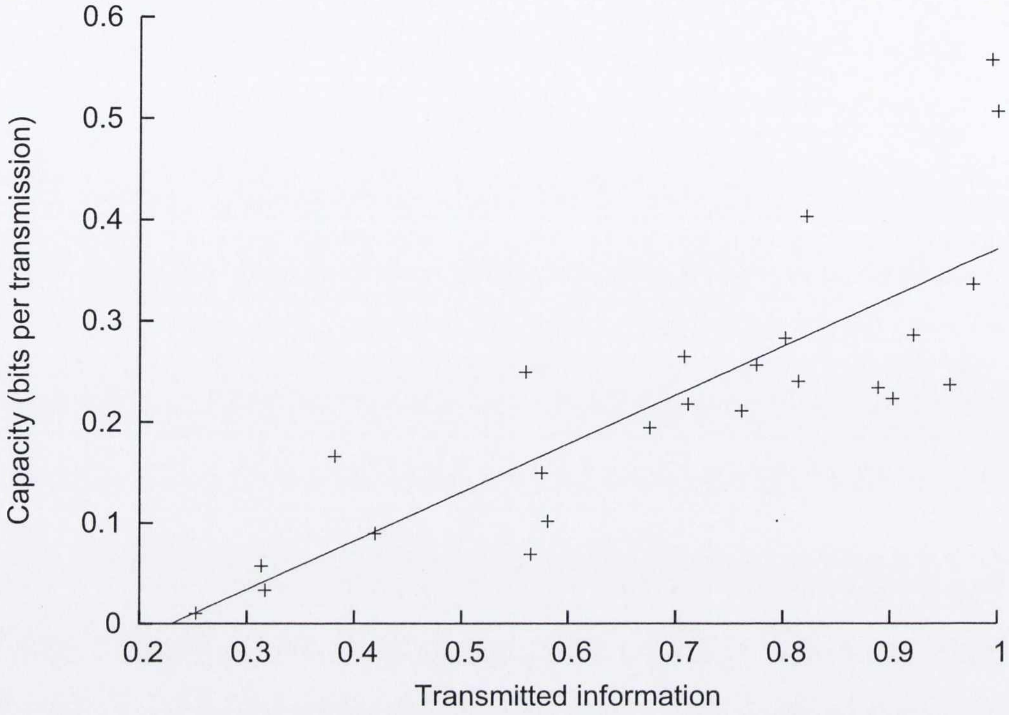

Figure 2.8: A plot of channel capacity per transmission against transmitted information for the 24 sites. Data was then fitted using least squares, with resulted in an $R^2 = 0.68$ linear relationship.

jects and a model of noise based on an analogy with a coordinate-based noise model, will be useful in developing this rate distortion theory.

The van Rossum metric is used. This is because it has an L^2 Pythagorean structure in the interval length. There is also an L^2 Victor-Purpura metric [10] but the L^2 structure in that case refers to the individual spikes. Of course, this raises the question of what is the most suitable metric structure for spike trains. This has been studied for the data used here by evaluating the accuracy of distance based clustering [15, 16, 17]. However, these comparisons show that the metrics all have a similar performance. Furthermore, the clustering-technique is really only a good probe for the local metric structure. The question of the most appropriate metric structure for spike trains has not been answered; the L^2 van Rossum metric is certainly the metric which fits easiest into the metric space approach to information that is proposed here.

Neural spike trains can be considered as point processes and developing a distance measure on spike trains from the perspective of point processes is a challenge to future approaches. Rate-distortion theory of point processes has suggested distance measures on spike trains [12, 13]. Although they may not seem natural in a neuroscientific context, a more comprehensive theory of information theory and the geometry of spike trains should make it possible to relate these metrics to spike train analysis. Conversely, it would be good to have a fuller account of what properties of spike trains distinguish them within the general family of point processes.

Both the gamma and log-normal distributions were also tested using the Anderson-Darling method, and both were found to pass. This should not, however, be seen to confuse the issue of the appropriateness of the χ-distribution. The χ-distribution arises from the distribution of distances measured between Gaussian vectors in information theory. Other distributions do not. The χ-distribution is used as a link between information theory and the metric space of spike trains. The other distributions have not been shown to provide this necessary link. Hence, although it is interesting to note that insufficient evidence exists to rule out other distributions, this result yields no additional information on the capacity theory itself.

This method has been applied to a single example dataset; it will be interesting to establish how well it performs on other sets of spike trains. In particular, it will be interesting to see if the basic assumption that the inter-fragment distances for noise responses satisfy a χ-distribution, is accurate. Conversely, it would be useful to generalize the discussion presented here to allow for other noise processes. Additional work would include an investigation of the signal distances to check if they too follow a χ-distribution, indicating that the signal distribution has been optimized.

While the current method is specific to a single neural channel, a number of possibilities exist to extend it to a network model. Using a multi-neuron metric, a series of parallel channels can be treated as a single channel, and an approach analogous to the single neuron method could be used. There already exist a number of multi-neuron metrics [35], hence it remains to identify the most useful, with the possibility of altering it, and apply it to the prospective data.

Channel capacity theory also addresses the problem of computing the capacity of Gaussian channels in parallel [9]. If the neurons in question are not interconnected then we can treat them as parallel Gaussian channels. However, assumptions about the distribution of the input and the independence of the noise in each channel could make this difficult to implement. Indeed it is likely that the multi-neuron metric approach will be used; it is more general than that of parallel Gaussian channels and is also easier to apply.

2.16 Additional note

The methods and results described were published in the Journal of Computational Neuroscience (Volume 30, Number 1, 201-209).

Chapter 3

The distribution of spike time jitter

3.1 Introduction

The Victor-Purpura metric is an example of an edit distance metric. A set of small edits is defined, comprising of moving spikes and adding or deleting spikes, along with an associated cost for each edit type. The distance between two spike trains is the minimum of all cost sequences of edits that transform one spike train into the other. One benefit of this metric type is that it pairs up spikes, one from one train and one from another, identifying which spikes should be considered as being related by noise-driven variations in timing.

The difficulty with analysing noise in spike trains is that there are two ways that a pair of spike trains resulting from the same stimulus can differ, [29]. The spike count can vary from train to train, with spikes present in one train absent in another. This is sometimes called *unreliability.* In addition, the time of specific spikes can vary between spike trains. This is called *jitter.* In the last chapter, the difficulty that this presents was avoided by studying noise on the metric space of spike trains [45]. Here, in contrast, the jitter component of the noise is studied on its own, using the Victor-Purpura metric to distinguish jitter from unreliability.

This new jitter algorithm is then applied to another zebra finch dataset and the distribution of jitter is studied. This distribution appears to be well-modelled by a hyper-Laplace distribution; the hyper-Laplace distribution is a sum of two Laplace distributions and so the jitter appears to be associated with two timescales. The lesser of these timescales is surprisingly small, having an average value of 3.9 ms for the data examined here, a timescale closer to that of the neurons refractory period. This indicates an unexpected precision in spike timing.

3.2 The Victor-Purpora metric

In the Victor-Purpura metric the distance between two spike trains is calculated by editing one so that it is the same as the other [27]. Each edit has an associated cost, and the distance between two spike trains is defined to be the minimum of the sum of the costs to turn one spike train into another. The set of operations is:

1. Insertion of a spike with a cost of one.

2. Deletion of a spike with a cost of one.

3. Moving a spike a distance Δt costs $q||\Delta t||$.

Here q is the free parameter of the metric and, like the timescale of the van Rossum metric, is chosen so two spike trains corresponding to same stimulus will typically be closer than spike trains corresponding to different stimuli. Again, an optimal value of q should be chosen using the information-based measure of the accuracy of metric-based clustering [27] previously described for the van Rossum metric timescale. As previously discussed, see section 2.3, the need to optimize q is one strength of the metric approach: as q interpolates between rate and temporal coding schemes, each neuron should by necessity have its own q value, or an alternative parameter reflecting the intrinsic coding scheme of the neuron. The dependence of the metric

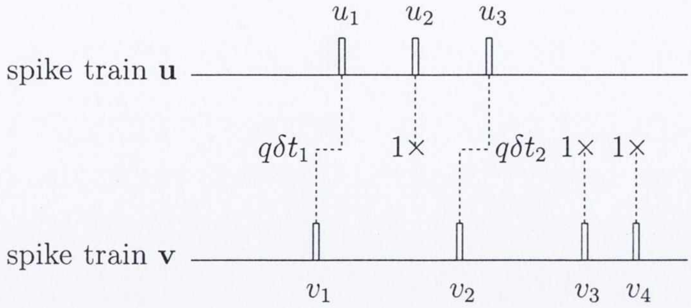

Figure 3.1: A diagramatic representation of the Victor-Purpura metric calculation. This illustrates the edits used to change the spike train $\mathbf{u} = \{u_1, u_2, u_3\}$ to $\mathbf{v} = \{v_1, v_2, v_3, v_4\}$. u_1 is moved to v_1 at a cost of $q\delta t_1 = q\|u_1 - v_1\|$, u_3 to v_2 at a cost of $q\delta t_2$; u_3 is paired to v_2 because it is closer than u_2. u_2, v_3 and v_4 are all deleted at a cost of three. The total cost is $3 + q\|\delta t_1 + \delta t_2\|$. The above calculation is obviously dependent on q, and for high q values no spike shifting takes place, just insertion and deletion. The lowest possible cost is defined as the metric distance.

on the q parameter represents the natural dependence of the spike trains on coding scheme, and hence should not be excluded from the modelling process.

If the distance between two spikes is greater than $2/q$ then it is cheaper to delete one spike and insert another. Thus, q is an inverse timescale that limits the time range over which it is useful to move a spike. Of course, this does not mean any two spikes less than that distance apart are related by jitter, as there may be a choice of which spikes to pair together.

There is a convenient algorithm for finding the minimum cost required for the Victor-Purpura metric; this algorithm is adapted here to give the jitter distribution. The Victor-Purpura algorithm can be easily described using a spreadsheet analogy. Consider two spike trains

$$\begin{aligned} \mathbf{u} &= \{u_1, u_2, \ldots, u_n\} \\ \mathbf{v} &= \{v_1, v_2, \ldots, v_m\}. \end{aligned} \tag{3.1}$$

Let $G_{i,j}$ denote the distance between the truncated trains formed by the first i spikes of train **u**, which will be called $\mathbf{u}|_i$ and the first j spikes of train **v**: $\mathbf{v}|_j$. Now, since the cost of inserting or deleting a spike is one,

$$G_{i,0} = i \text{ and } G_{0,j} = j. \tag{3.2}$$

which can be seen in the initial rows and columns of the spreadsheet samples in panels **C** and **D** of Fig. 3.2. From the properties of the minimum-cost path, it can be shown that the other entries of G can be found recursively starting at $G_{1,1}$ using

$$G_{i,j} = \min\left\{G_{i-1,j-1} + q|u_i - v_j|, G_{i-1,j} + 1, G_{i,j-1} + 1\right\}. \tag{3.3}$$

These three possibilities correspond to the three edit types: jitter or inserting a spike or deleting a spike. The rows and columns are filled with each new position (i, j) chosen so that the entries $G_{i-1,j}$, $G_{i,j-1}$ and $G_{i-1,j-1}$ have already been calculated [27, 47]. The distance between the trains $\mathbf{u}$ and $\mathbf{v}$ is then the bottom right entry in the completed spreadsheet, $G_{n,m}$.

Here, jitter is defined as the temporal distance between pairs of spikes, where one is moved to match the other in the minimum cost Victor-Purpura edit. The minimum cost edit can be reconstructed from the spreadsheet used to calculate the Victor-Purpura distance. Imagine keeping a second spreadsheet alongside the spreadsheet of G_{ij} values: a *directions* spreadsheet. The entries in the directions spreadsheet are arrows pointing either upwards, $\uparrow$, to the left, $\leftarrow$, or diagonally upwards and to the left, $\nwarrow$. To maintain a spreadsheet analogy, a spreadsheet rather than matrix index order is used: $G_{i-1,j}$ and $G_{i,j-1}$ are described as being respectively to the left and above $G_{i,j}$.

Each time an entry is added to the G_{ij}-spreadsheet, it is chosen to be the minimum of three possibilities. If

$$G_{ij} = G_{i-1,j-1} + q|u_i - v_j| \tag{3.4}$$

is the minimum then the spikes u_i and v_j are related by jitter and the minimum cost edit between the truncated spike trains $\mathbf{u}|_i$ and $\mathbf{v}|_j$ is given by adding $q|u_i - v_i|$ to the minimum cost edit between $\mathbf{u}|_{i-1}$ and $\mathbf{v}|_{j-1}$. In this case, a diagonal arrow, $\nwarrow$, pointing from (i, j) to $(i-1, j-1)$ is added to the (i, j) entry in the directions spreadsheet. Similarly, if

$$G_{ij} = G_{i-1,j} + 1 \tag{3.5}$$

then, in the minimum cost edit relating $\mathbf{u}|_i$ to $\mathbf{v}|_j$ the spike u_i is deleted and the total cost is the sum of the cost of that deletion, one, added to the minimum cost edit relating $\mathbf{u}|_{i-1}$ to $\mathbf{v}|_j$. In this case, a left arrow, $\leftarrow$ pointing from (i, j) to $(i-1, j)$ is added to the (i, j) entry in the direction spreadsheet. Finally, if $G_{ij} = G_{i,j-1} + 1$ an up arrow, $\uparrow$, pointing to $(i, j-1)$ is added. The minimum path can now be deduced by following the arrows in the directions spreadsheet back from the (n, m) entry. The jitter values, $u_i - v_j$, correspond to the (i, j) entries with a diagonal arrow in the direction spreadsheet. This jitter may be positive or negative. An example of this calculation is given in Fig. 3.2.

3.3 The jitter algorithm

Any entry $G_{i,j}$ in the Victor-Purpora spreadsheet gives the minimum cost required to transform the positions of the first i spikes of $\mathbf{u}$ into the positions of the first j spikes of $\mathbf{v}$.

Each entry $G_{i,j}$ has value of either $G_{i,j-1} + 1$, $G_{i-1,j} + 1$ or $G_{i-1,j-1} + q||\Delta t||$, where the minimum of the three possibilities is chosen. For each entry a record can be made of which of the three entries preceding it incurred the minimum cost. Say we use labels a, b and c respectively to denote whether the path of minimum cost to $G_{i,j}$ is obtained from $G_{i,j-1}$, $G_{i-1,j}$ or $G_{i-1,j-1}$ respectively. Then each spreadsheet entry has associated with it either a, b or c.

Obviously the edge entries form special cases where only two letters are possi-

A

u		u_1	u_2		u_3		
times	0.515	0.55	0.65	0.71	0.75	0.88	0.95
v	v_1			v_2		v_3	v_4

B

	0	1	2	3
0	0	1	2	3
1	1	$q\delta t_1$	$1 + q\delta t_1$	$2 + q\delta t_1$
2	2	$1 + q\delta t_1$	$q(\delta t_1 + \delta t_3)$	$1 + q(\delta t_1 + \delta t_2)$
3	3	$2 + q\delta t_1$	$1 + q(\delta t_1 + \delta t_3)$	$2 + q(\delta t_1 + \delta t_2)$
4	4	$3 + q\delta t_1$	$2 + q(\delta t_1 + \delta t_3)$	$3 + q(\delta t_1 + \delta t_2)$

C

	0	1	2	3
0	0	1	2	3
1	1	$\nwarrow$	$\leftarrow$	$\leftarrow$
2	2	$\uparrow$	$\nwarrow$	$\nwarrow$
3	3	$\uparrow$	$\uparrow$	$\uparrow$
4	4	$\uparrow$	$\uparrow$	$\uparrow$

D

	0	1	2	3
0	0	1	2	3
1	1	$\nwarrow$	$\leftarrow$	
2	2			$\nwarrow$
3	3			$\uparrow$
4	4			$\uparrow$

Figure 3.2: An example calculation of the jitter. Here, two example spike trains are considered, with spike times given in the table **A**. These times were also used for the illustration of the Victor-Purpura edit in Fig. 3.1. $\delta t_1 = u_1 - v_1 = 0.035$, $\delta t_2 = u_3 - v_2 = 0.04$, $\delta t_3 = v_2 - u_2 = 0.06$. For this example, $q = 15$ so that $q\delta t_1 = 0.525$, $q\delta t_2 = 0.6$ and $q\delta t_3 = 0.9$, all the other intervals give costs greater than one. The spreadsheet for the metric calculation is given in **B**, the actual distance is given by the the bottom right hand entry $3 + q\|\delta t_1 + \delta t_2\| = 4.125$. **C** is the directions spreadsheet, a map of how the spreadsheet is calculated. A left arrow is shown when the new entry is calculated by adding one to the entry on the left, an up arrow when it is calculated by adding one to the entry above and a diagonal arrow when it is calculated by adding the cost of jitter to the entry diagonally up from it. The actual route, working back from the bottom-right entry, gives the actual sequence of edits. This route is given in **D**. The two diagonal arrows, $\nwarrow$, along this route give the two jitter values, δt_1 and δt_2.

bilities, and the entry $G_{0,0}$ has no letters associated with it. This allows a path of minimal cost to be traced out from the entry $G_{m,n}$. For example, if $G_{m,n}$ has c associated with it, then a record is made of the jitter $||\Delta t||$ added to the entry $G_{m-1,n-1}$. Next the letter associated with $G_{m-1,n-1}$ is checked. If, for example, it has letter a, then a record is made that a deletion has occurred and $G_{m-2,n-1}$ is considered next. The process continues until the $G_{0,0}$ entry is reached. At this stage a set of three lists has been compiled, corresponding to the history of insertions, deletions and jitters that took place along the path of minimal cost taken to reach $G_{m,n}$.

Every time the distance between two spike trains is computed a corresponding history of operations along the minimal cost path can be recorded. Again this history is divided into three categories, insertions, deletions and jitter. For a set of 200 spike trains, divided into 20 groups of ten spike trains corresponding to 20 stimuli, 900 same-stimulus distances can be calculated. Addressing for the moment only the jitter aspect of the calculation and ignoring the history of insertions and deletions, a complete record of all fluctations in temporal structure, the jitter, is now available. Kernel density estimation can be used to estimate the probability density of the jitter distribution.

3.3.1 Data

A different dataset is used here than in the previous chapter. Data is from the large extracellular zebra finch dataset made available on the Collaborative Research in Computational Neuroscience database by the Frederic Theunissen laboratory at UC Berkeley [40].

The song corpus for the dataset generally includes 20 songs. The songs have variable length but all are at least one second long. Here, the first second is used for all songs. A small number of cells are excluded because they contain trials with no spikes. This leaves a total of 449 cells. The number of trials for each song varies, but is most commonly ten. The optimal value of q for each cell is calculated

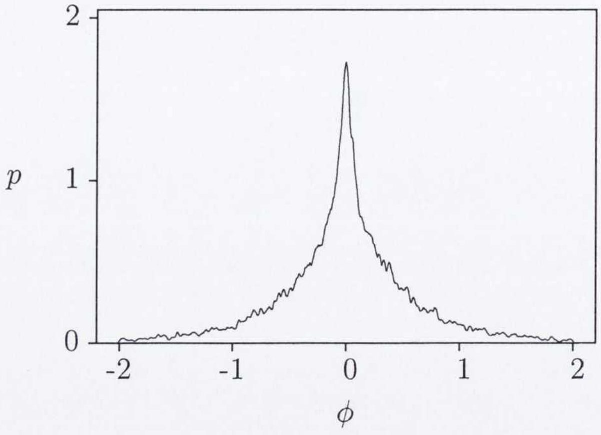

Figure 3.3: An example jitter distribution. An example cell has been chosen, which has a reasonably high number of jitter values, 18735. This means that the continuous jitter distribution can be estimated from the calculated values using kernel density estimation with a small band width kernel. The scaled jitter is used for the horizontal axis, with $\phi = q\delta t$. Here, an Epanechnikov kernel is used with bandwidth 0.02. This cell has $q = 54.05$ and is labelled `blabla0713_2_B` in the dataset.

in the usual way, by evaluating the accuracy of metric clustering [27, 17]. These spike trains and the experimental conditions used to produce them are described in [49, 44, 51, 50, 41].

3.4 Applying the jitter algorithm

For each pair, the order they were evaluated in is chosen randomly. This was to eliminate a very small bias in the sign of the jitter that would result from always evaluating a pair in the same order as the trials were measured. The bias is thought to arise from a slight slowing of the neurons spiking response to repeated stimulus. Typically a cell will have ten responses to twenty songs and, therefore, 900 such spike train pairs. The jitter values for all pairs are aggregated. This is a very large sample from the jitter distribution, for the 449 cells studied the average number of jitter values is 7075. There is considerable variation, the cell with the most values has 117923 and the one with the least has 266.

A typical jitter distribution is shown in Fig. 3.3. This distribution is clearly not Gaussian. Infact, it is possible to rule out the Gaussian distribution for almost all of the jitter distributions. Of course, the jitter distribution has a compact support. For simplicity, the scaled jitter

$$\phi = q\delta t \tag{3.6}$$

is used and

$$-2 \leq \phi \leq 2 \tag{3.7}$$

since spikes are only related by jitter if moving a spike is cheaper than deleting one and adding the other. This means that the Gaussian distribution that is used is actually a compact version restricted so

$$p(\phi) = \begin{cases} \frac{1}{Z}\exp\left(-\frac{\phi^2}{2\sigma^2}\right) & |\phi| \leq 2 \\ 0 & \text{otherwise} \end{cases} \tag{3.8}$$

where

$$Z = \sqrt{2\pi\sigma^2}\mathrm{erf}\left(\frac{2}{\sqrt{2\sigma^2}}\right) \tag{3.9}$$

and $\mathrm{erf}(x)$ is the error function. Z is needed to ensure that $p(\phi)$ integrates to one despite being cut off at $\phi = \pm 2$. For each cell, the closest Gaussian distribution was found by maximizing the log-likelihood of the calculated jitter values using the Brent algorithm [46]. This Gaussian best-fit is plotted for the example cell in Fig. 3.4A. It is clearly a poor fit.

Here again the Anderson-Darling test is used to evaluate goodness-of-fit quantitatively [42]. As no table of critical values exists for the hyper distributions, the Anderson-Darling statistic calculated from the real data will be compared to the distribution of values that the same statistic will have for artificial data drawn from the hypothesised distribution. This will allow for a hypothesis test of the data.

The proposition that the jitter distribution is Gaussian can be ruled out with

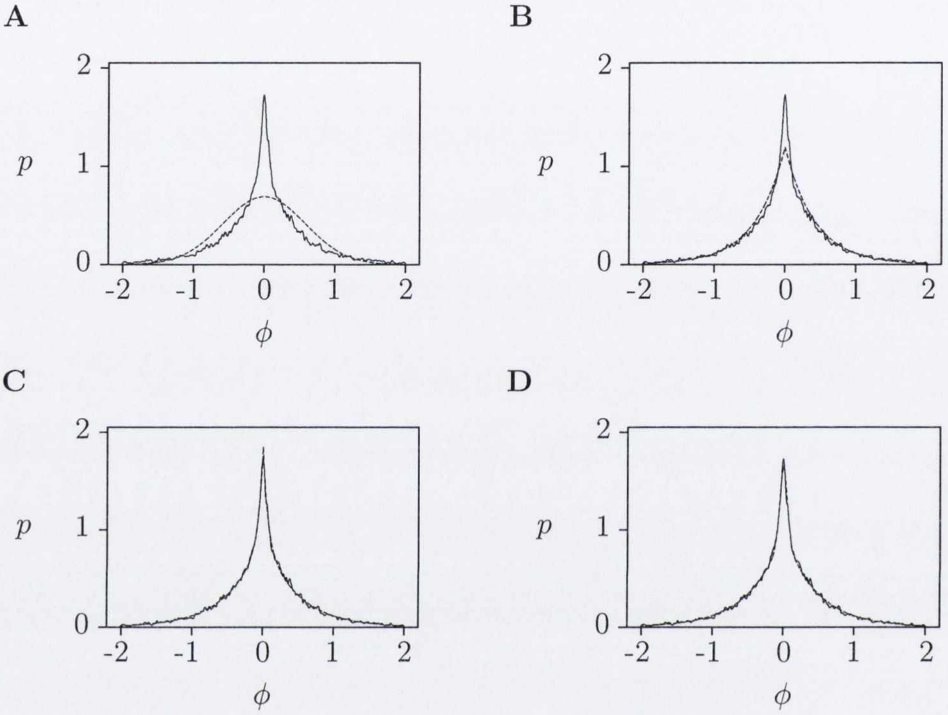

Figure 3.4: Fitting probability densities to the example jitter distribution. In each case, a probability density has been fitted by maximizing the log-likelihood. The four densities are (**A**) Gaussian with $\sigma^2 = 0.334$, (**B**) Laplace with $b = 0.423$, (**C**) hyper-Laplace with $p_1 = 0.12$, $b_1 = 0.062$ and $b_2 = 0.48$, and (**D**) hyper-Gauss with $p_1 = 0.9$, $\sigma^2 = 0.003$ and $b = 0.47$. In each case, the kernel density estimate of the distribution is a solid line and the distribution fitted to it a dashed line. In the Anderson Darling test, the hyper-Laplace distribution (**C**) scores 0.171 and the hyper-Gaussian (**D**) scores 0.098, the other two score zero. Obviously the hyper-Laplace and hyper-Gaussian distributions match extremely well. Zooming in to the two graphs shows that the peak of the hyper-Gaussian falls slightly short of the jitter distribution.

significance greater than 99% percent for all the cells. From an inspection of the corresponding graphs, it is clear that the problem with the Gaussian distribution is that it does not have a heavy enough tail to match the data. A more leptokurtotic distribution is needed. One possibility is the Laplace distribution, the symmetrical version of the exponential distribution,

$$p(\phi) = \frac{1}{Z}\exp\left(-\frac{|\phi|}{b}\right) \tag{3.10}$$

where b is a scale parameter and Z the normalization factor. A Laplace distribution is an attractive possibility because of the relationship between it and Poisson processes; waiting times for spikes generated by a Poisson process follow an exponential distribution, while intervals between such iid exponential variables follow a Laplace distribution. However, even this is not sufficiently leptokurtotic. To make the distribution compact, $p(\phi)$ is set to zero for $|\phi| \geq 2$ and the density is normalized by setting

$$Z = 2b\left[1 - \exp\left(-2/b\right)\right]. \tag{3.11}$$

Best fits were calculated by optimizing the scale parameter b using log-likelihood and the Brent algorithm, as for the Gaussian. The best fit for the distribution is shown in Fig. 3.4B and, although the fit looks quite good, the Laplace distribution fails the Anderson-Darling test with significance 95% for all but five of the 449 cells.

A still more leptokurtotic possibility is the hyper-Laplace distribution

$$p(\phi) = \begin{cases} \frac{1}{Z}\left[\frac{p_1}{2b_1}\exp\left(-\frac{|\phi|}{b_1}\right) + \frac{p_2}{2b_2}\exp\left(-\frac{|\phi|}{b_2}\right)\right] & |\phi| \leq 2 \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

with

$$Z = \frac{p_1}{2}\left[1 - \exp\left(-2/b_1\right)\right] + \frac{p_2}{2}\left[1 - \exp\left(-2/b_2\right)\right] \tag{3.13}$$

and $p_1 + p_2 = 1$. This distribution corresponds to a statistical process with a

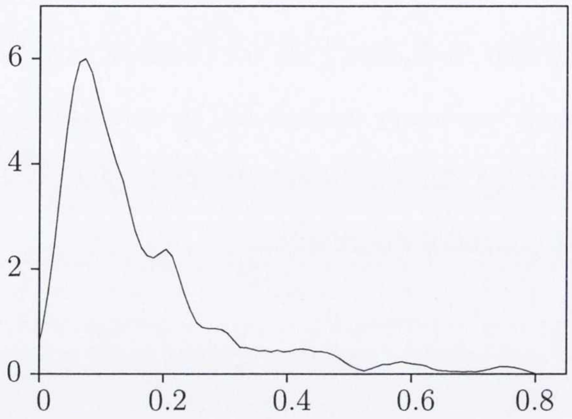

Figure 3.5: The values of b_1. Kernel density estimation has been used to give the distribution of b_1 values for the 427 cells with $b_1 < 0.8$. The remaining 22 cells are sporadically distribution with isolated values going up to 20.52.

probability p_1 of behaving like a Laplace distribution with scale parameter b_1 and probability p_2 of behaving like one with scale parameter b_2. The parameters p_1, b_1 and b_2 were fitted by maximizing the log-likelihood as before. As there are multiple parameters, the amoeba algorithm is used [46]. The fit for the example cell is extremely good, see Fig. 3.4C. Infact, only for 65 of the 449 cells can the hypothesis be rejected with significance 95%. For simplicity, it is assumed that $b_1 \leq b_2$. Over all 449 cells, the average values are $p_1 = 0.37$ and $b_1 = 0.41$. However, this value of b_1 is misleading since it is dominated by outliers. If the 22 values greater than 0.8 are removed, the average becomes 0.16. The distribution of b_1 values is graphed in Fig. 3.5. The value of $2/q$ is often interpreted as an indicative timescale for precision of spikes. However, the measurement of jitter seems to indicate that spike times are more precise than this: since $\phi = q\delta t$, an average b_1 value of 0.16 shows considerably finer temporal percision than $2/q$ would suggest. Dividing the average value of the smaller timescale by q, which is calculated over all 449 cells, gives 3.95 ms; this is considerably smaller than 49.2 ms, which is the average value of $2/q$.

As a final example, a hyper-Gaussian distribution is considered. This is similar to the hyper-Laplace distribution in that it is composed of two different distributions, it has one Gaussian term and one Laplace term. Although the Anderson-Darling is extremely sensitive for large datasets, the hyper-Laplace and hyper-Gaussian distributions are very similar for suitable choices of parameters. The test only rejects the best-fit hyper-Gauss distribution 84 times at significance 95%. Visually, the hyper-Gauss appears to be a good fit. It is plotted for the example cell as Fig. 3.4D. As such it is hard to choose between the hyper-Laplace and hyper-Gaussian distributions by counting how many times each is rejected by the Anderson-Darling test.

The Anderson-Darling statistic can be considered as a measure of goodness-of-fit by examining the number of artificial values that are greater than the real value. The higher this number, the further the distribution is from being rejected. The average number of artificial values greater than the true value is 0.43 for the hyper-Laplace, larger than the 0.34 for the hyper-Gaussian. The score for hyper-Laplace distribution is greater than that for the hyper-Gaussian for 307 cells. Thus, the hyper-Laplace and hyper-Gaussian distributions both appear to explain the calculated jitter distributions but the hyper-Laplace distribution appears to give the better fit.

3.5 Discussion

Using additional book keeping, the usual spreadsheet method for calculating the Victor-Purpura metric is extended to give values for spike jitter. This method has been applied to an example dataset. The resulting jitter appears to be well described by a hyper-Laplace function.

The result is unusual in the sense that the jitter is non-Gaussian, but is instead drawn from a distribution which can be considered as bimodal, indicating that two timescales are needed to describe the jitter. In considering why jitter could have a hyper-Laplace distribution, let us first consider why it could have a simple Laplace distribution. This can be seen through examining a basic firing rate model.

One interesting issue concerns the jitter in individual spikes: are there some spikes that have a large amount of jitter and others that have small amounts, or do most spikes have both small and large jitter values? As a rough test of this the average size of the jitter for each spike was calculated. For each cell a histogram was constructed with 20 bins, with the bin width chosen so that the largest average value falls into the 20th bin. These bins where then tested for bimodality: the bins with values less than half the maximum value were set to zero to reduce noise and the discrete derivative formed by taking the difference of a bin and its succeeding bin was calculated. 232 cells showed evidence of bimodality under this definition. However, visual inspection shows that for many of these cells the second local maximum is a small fluctuation near the global maximum and many of the cells exhibiting this bimodality have small numbers of jitter values. The cells with bimodality have, on average 513.44 jitter values, compared to 1757.0 for the cells without. Thus, it seems likely that more sophisticated analysis, for example Hartigan's dip test, will show that individual spikes tend to have both small and large jitter values, but this would require further analysis.

In a basic firing rate model the number of spikes a neuron generates during a certain time interval is typically modelled using a Poisson distribution [3]. The Poisson distribution itself is derived from the concept that, were one to wait for the instance of a spike, the distribution of waiting times follows an exponential distribution. The number of spikes occurring during an interval, assuming an exponential distribution of waiting times, is given by a Poisson distribution. Now a Laplace distribution governs the difference between two iid exponentially distributed random variables. Here we have used the Victor-Purpora metric to measure the difference between two spikes, which themselves, under a Poisson firing rate distribution, can be hypothesised to have exponentially distributed waiting times. Hence differences between spikes from different trains would in theory have a Laplace distribution.

The presence of a hyper-Laplace could be modelled using a 'two state' neuron: the neurons firing rate is modelled by two Poisson processes. The resulting spike train is not a superposition of the two processes, but rather the neuron fires in one Poisson state for some time interval, then switches to its second state to fire for another time interval. By using the Victor-Purpora metric to examine differences between paired spikes we are implicitly extracting two sets of differences, each corresponding to one of the two Poisson processes. The resulting distribution would then be hyper-Laplace.

Chapter 4

A method of generating artificial spike trains

4.1 Introduction

The variation in the temporal structure of spike trains can be modelled using a hyper-Laplace distribution. Now assuming the variation in same-stimulus spike count, termed unreliability, can be measured, a mathematical description of the two separate regimes of spike train noise can be obtained.

Here, the idea is to develop an overall generative model for spike train noise through the combination of the hyper-Laplace model of jitter and a model of spike count unreliability. This model can then be applied to a spike train, allowing for a set of artificial trains to be generated; trains generated from the original spike train through alterations of its spike count and temporal structure in accordance with the jitter and unreliability models. These artificial trains should typically be found in the response space of a neuron for that particular stimulus.

4.2 The noise distribution

If distances between spike trains corresponding to the same stimulus are computed, then the resulting collection of distances reflects the noise associated with that neuron. This approach can be generalized to a set of stimuli and neurons: the distances between spike trains corresponding to both the same stimulus and neuron can be computed.

For a particular neuron, kde can be used to plot the estimated distribution of these distances, Fig. 4.1. It should be noted that other than smoothness, kde makes no assumption as to the specific type of density function the distances should follow. Instead, it provides an empirical estimation akin to a smoothed histogram.

The empirical distribution that results from kde is termed here as a *noise distribution.* Each neuron has its own noise distribution, although it is reasonable to assume that this distribution will change over time. For datasets involving short one to two second presentations of conspecific song to zebra finch, stationarity of the noise distribution is assumed.

In this way a noise distribution, which represents an overall characterization of noise, can be associated with each neuron. If a spike train from a dataset is altered both temporally and in its spike count using jitter and unreliability models, then distances between the resulting artificial trains should behave as if they were drawn from the neurons noise distribution with high probability. In this sense, the noise distribution is the final part of a generative model linking various noise models to each other.

4.3 Rejection sampling

When only an estimate of an unknown probability distribution exists, as is the case with kde, the technique of rejection sampling can be used to draw random variables from the estimated distribution [46]. Rejection sampling is based on repeatedly

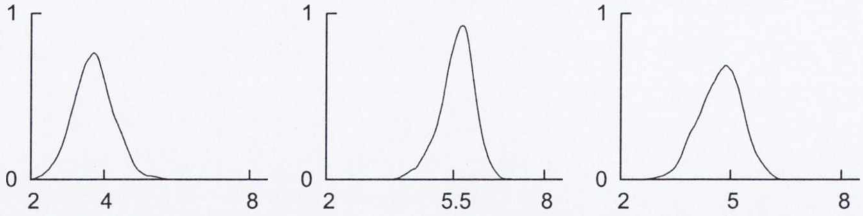

Figure 4.1: Kernel density estimation of noise distributions of three cells. The x-axis represents the van Rossum distance between spike trains. Neuronal data here is taken from the zebra finch auditory forebrain, with $\tau = 12.8$ ms

sampling random variables from an envelope distribution, then rejecting all but those that could have been drawn from the underlying distribution with high probability.

Let f denote the estimated density function, that is, the function we want to sample from. A suitable envelope function, g, can be chosen such that

$$f(x) \leq cg(x) \tag{4.1}$$

where c is a constant that ensures the inequality holds. As will be seen, c must be large enough for the inequality to hold, yet the larger the L^2 distance between f and cg the slower the sampling process.

A sample x, drawn from $g(x)$, is accepted as being drawn from $f(x)$ if

$$u \leq \frac{f(x)}{cg(x)} \tag{4.2}$$

with

$$u \sim U(0, 1). \tag{4.3}$$

Here $U(0, 1)$ is simply the uniform distribution on interval $[0, 1]$, and u is drawn randomly from $U(0, 1)$ at the same time that x is drawn from g. In a practical way this constrains g somewhat: while g should have a shape close to that of the estimated distribution, suitable algorithms must also be available to allow easy sampling from g. This excludes more obscure but perhaps better fitting distributions from the process.

The more accurate the envelope function the less time it will take to sample from f as the ratio of estimated density to envelope will generally be close to one, and most samples will be accepted. Rejection sampling will be used to compare artificial trains to those associated with a neurons noise distribution.

4.4 Unreliability and the Victor-Purpora metric

As has been previously described, the Victor-Purpora metric can be used to keep track of jitter associated with spike train responses to same stimulus presentations. However, this neglects another use of the metric: the metric can be used to document the unreliability associated with a neuron.

The number of spike deletion and insertion operations carried out during the comparison of two spike trains can be recorded in a process identical to the recording of spike jitter.

4.5 Families of spike trains

Both the jitter and unreliability distributions can be sampled from, in such a way as to allow for the generation of spike train noise. For instance, spikes of a particular spike train could be shifted according to the jitter distribution, and additional spikes could be inserted or deleted using a model consistent with spike train unreliability results. This generates a spike train which differs from the original through noise typically found in the dataset. If this process is repeated multiple times then a family of artificial spike trains would be associated with the original spike train.

The concept of a mean or template spike train is not new, [34]. One method of finding a mean spike train is through using a medoid clustering approach: if ten spike trains correspond to the same stimulus, then the spike train with the minimum average least squares distance to all the other nine trains is termed the mean spike train. The aim is to generate sets of artificial trains from this mean. These artificial spike trains will also be associated with that particular stimulus.

4.6 Spike deletion

Spike deletion is one of the three operations the Victor-Purpora metric can use to transform one spike train into another. Let D_{ij} denote the number of spikes deletions implemented during the transformation of the ith train into the jth train, and let N_i and N_j denote the numbers of spikes in the ith and jth trains respectively. Then we can define a_k as

$$a_k = \frac{2D_{ij}}{N_i + N_j}. \tag{4.4}$$

The index k represents the kth instance of a metric distance computation, and a_k is the average number of deletions needed per train for the computation. Note that if no deletions take place the result is not stored as only jittering is needed for the transformation.

The probability of overall spike deletion, notated p_d, is just the mean probability of a_k:

$$p_d = \sum_{k=0}^{n} \frac{a_k}{n} \tag{4.5}$$

where n is the number of metric distance computations in which deletions have taken place.

Now lets say we have a template or mean spike train, denoted T. Each spike in T is firstly selected, then deleted if u passes the following test:

$$u < p_d \tag{4.6}$$

where again $u \sim U(0,1)$. The resulting train is labelled T_d.

4.7 Spike jitter

The remaining spikes in T_d must now be jittered, that is, shifted by time periods commonly associated with neuronal noise. To do this we must draw from the hyper-Laplace distribution of the neurons jitter. Doing this is a simple procedure, to draw from a Laplace distribution generate $x \sim X$ where

$$x = \mu - bs(u)\ln(1 - 2|u|). \tag{4.7}$$

Here μ and b are the free parameters of the Laplace, $u \sim U(-0.5, 0.5)$, and

$$s(Y) = \begin{cases} -1 & \text{if } Y \leq 0 \\ +1 & \text{if } Y > 0 \end{cases} \tag{4.8}$$

Since the hyper-Laplace is a combination of two Laplacians,

$$H = f_1 L(\mu_1, b_1) + f_2 L(\mu_2, b_2) \tag{4.9}$$

random variables can be drawn from H by drawing from either $L(\mu_1, b_1)$ with probability f_1 or $L(\mu_2, b_2)$ with probability f_2.

Every time a variable is drawn from H it is added to a spike in T_d until all of T_d has been jittered. The jittered train, T_j, is then

$$T_j = \{t_1 + s_1, ..., t_l + s_l\}. \tag{4.10}$$

The resulting train T_j is a template train that has been subjected to both spike deletion and spike jitter.

4.8 Spike insertion

Just as the average number of deletions per spike train can be calculated, so can the average number of insertions. This leaves the task of actually inserting spikes. Here a Poisson process is used to do this. The process rate λ is equal to the average number of insertions, and I_a is the average number of insertions per spike train comparison:

$$\lambda = \frac{I_a}{2}. \tag{4.11}$$

The denominator of two is used because two spike trains are compared, hence the number of insertions will be on average half of the total number of insertions used in the spike train comparison.

A set of insertion spikes $I = i_1, .., i_m$ can then be generated using the standard Poisson process generator:

$$i_{q+1} = i_q - \frac{1}{\lambda} \ln(u) \tag{4.12}$$

where $u \sim U(0, 1)$ and spikes are inserted until a spike occurs outside the interval λ was measured over, in this case one second. The process ends once a spike occurs with a value greater than one second.

The set of insertion spikes I are then combined with the list of spike times in T_j, and the artificial train T_a results. This represents a template spike train that has been subjected to all the operations of the Victor-Purpora metric, deletion, jittering and insertion.

4.9 The rejection sampling of artificial trains

Multiple artificial trains can be created using the above generative model. However, these trains must be tested further. This is to ensure that metric distances measured between the artificial trains and the template train resemble distances drawn from the original noise distribution with high probability.

Firstly, the van Rossum distance between each artificial train and the template train must be computed. While it is not necessary to use the van Rossum metric, and indeed the Victor-Purpora metric could be used instead, the metric is used to link this method with methods described in chapter five, where noise distributions generated using the van Rossum metric are used extensively.

Again kde can be used to estimate the density function underpinning the set of distances measured between the artificial and template spike trains, and this density is denoted $g(x)$. The variable this density depends on, x, is a distance. The density g will effectively form our envelope function since we are sampling directly from it.

If the noise distribution is denoted $h(x)$, then each artificial distance x_i can be tested using

$$u < \frac{h(x_i)}{cg(x_i)} \tag{4.13}$$

where $u \sim U(0,1)$ and c are chosen as outlined before.

Each x_i corresponds to a distance between a particular artificial spike train and the template train. So if x_i fails the rejection test then the corresponding artificial train is discarded from the set of artificial trains.

If the cardinality of the resulting set of artificial spike trains is smaller then required, more trains can be generated and rejection sampled until the desired quantity is achieved. This allows a family of artificial trains to be generated from a template spike train, with noise consistent with that typically found in the neurons dataset.

4.10 Discussion

The generative model of spike train noise was programmed in C++, and was successfully used to generate spike trains with noise consistent with that obtained from the data itself. A homogeneous Poisson process was used to insert spikes into the artificial spike train. One might question why a 'two state' neuron was not used to generate spikes. While a two state neuron could have been used, the number of spikes that need to be generated, typically of the order of about two, is low enough that it was thought a two state would add unnecessary complications to the process. Also the rejection sampling of the spike trains is used to filter out noise uncharacteristic of the original dataset, implicitly filtering uncharacteristic jitter. Should a whole spike train need to be generated, of length one second say for auditory forebrain data, then the use of a two state neuron is advocated.

The generative model depends on the timescales of the jitter distribution, and the timescale of the van Rossum metric. This should be seen in a positive light; a timescale can be seen to 'tune' a metric to reflect the level of temporal and rate coding present in the spike trains of that neuron. This varies from neuron to neuron, hence each neuron should have its own set of timescales. This dependence of the generative model on these timescales allows for a more accurate model of that particular neurons spike activity.

An alternative approach would be to use a peristimulus time histogram, [33], constructing a histogram and drawing samples from it. The first drawback of this method is that it involves binning the spike trains, a process that, as has been previously discussed, is unnatural and difficult to apply to spike trains. Secondly, by focusing on binwidth only, this process ignores the other timescales found present in the data through the use of a metric space approach. Its advantage would be found in its simplicity of approach, but at a cost of ignoring the hyper-Laplace and noise distributions of the neurons themselves.

The method presented is a means of adding noise to spike trains. There are two obvious situations where this could be utilized: if we need to add noise to a spike train generator, or we need to generate spike trains that are close but not identical to a template spike train. Here the concept of template spike train is slightly ambiguous; it can refer to a spike train that is on average closest to every other spike train corresponding to a specific stimulus, so can be seen as a medoid spike train. It could also be generated through summing function mapped same-stimulus spike trains, then thresholding to retain dominant spikes. The selection of a template spike train is not the focus of this work, instead an implementation of combined noise models was presented.

Chapter 5

The noise space of neurons

5.1 Introduction

Accurately associating probability distributions with spikes and spike trains is certainly an elusive process. The same cannot be said of neuronal coordinates however; parameters such as firing rate can be used to describe the state of a neuron, or phrased geometrically, embed the neuron on a manifold with local coordinates given by the chosen neuronal parameters.

An interesting development of information geometry [36] is that distributions are themselves elements of a structure analogous to a manifold know as a *statistical model.* Families of distributions typically have a number of unspecified parameters, termed here as *free parameters.* For example, the free parameters of a Gaussian are given by its first and second moments. The free parameters of a family of distributions form a set of coordinates for each distribution in this space. A simple example is that of the space of Gaussian distributions. Elements of the set are the Gaussian distributions themselves, while the coordinates of each Gaussian are given by particular values of its mean and variance. As only two parameters are needed to specify the coordinates of each element, this statistical model has a dimension of two.

Like parametrizing the surface of a sphere, there are a number of sets of coordinates available to parametrize the distribution space, and there is generally no reason to believe the free parameters form the most useful set. Indeed for the space of neurons different coordinate sets are used, with some proving more informative than others.

It has previously been asserted that the distances between same-stimulus spike trains follow a χ-distribution. In this way, a χ-distribution can be associated with each neuron from the dataset. This is equivalent to embedding the set of neurons on a statistical manifold, termed here as the *noise space of neurons.* As the χ-distribution has two free parameters, the dimension of the related manifold is two.

Multidimensional scaling and principal component analysis are among a range of methods that can be used to isolate the geometric structure of a dataset. They do not assume that a parametric model for the dataset exists. Their application can, however, provide insights into which parametric models are appropriate for the dataset [28]. Hence, a parametric model can be derived from their results. For instance, if data is embedded on the surface of a relatively flat manifold, multidimensional scaling provides a means of visualizing the surface of this manifold. Application of such a method will also estimate the dimension of the structure, that is, the number of parameters needed. This assumes of course that a structure exists in the first place.

One such method, a refinement of multidimensional scaling, is isomap. The advantage of using isomap over other methods is in its use of geodesic distances: isomap is able to estimate non-Euclidean structures, that is, isomap is useful when data is embedded on a non-flat manifold. Hence, choosing isomap makes sense when the nature of the space remains uncertain. Here it is proposed to use isomap to estimate the dimensionality of the neuronal space. As a backdrop, this will again be testing the χ assertion, as the dimension of the space should turn out to be approximately two.

While isomap generates an Euclidean embedding of the space, the method is used with the knowledge that instability can result from the effects of noise on aspects of the algorithm. In certain cases [39], the introduction of noise to a system is found to produce disproportionately large changes in the Euclidean embedding. Here, the algorithm is first tested on noisy datasets with a known theoretical outcome. Distortions found in this application can be taken into account when applied to unknown datasets.

The dataset used here comprises of spike trains from 455 neurons, that is, 455 points will be used to map out the geometry of the space. To further explore the spaces geometry it is useful to generate artificial data points, and scatter these throughout the space to highlight structure. Here a method is proposed to do this, which at the same time does not alter the structure or relative positions of the original points in the space.

5.2 Statistical model

A set of distributions form a space termed a statistical model [36]. This is analogous to a manifold, where each element in the space is itself a distribution, and the coordinates of each element are given by the specific free parameters of its distribution.

If a distribution is given as

$$p : \chi \to \mathbb{R} \tag{5.1}$$

then its corresponding statistical model is given:

$$S = \{p_\xi = p(x;\xi) | \xi = (\xi^1, .., \xi^n) \in \Xi\} \tag{5.2}$$

where $\Xi \subset \mathbf{R^n}$ Again, analogous to a manifold, the dimension of the space is given by the number of free parameters:

$$\dim(S) = n \tag{5.3}$$

Distances between elements are computed using the Fisher metric, a Riemannian metric:

$$g_{ij}(\xi) = \int \partial_i(\log p(x;\xi))\partial_j(\log p(x;\xi))p(x;\xi)\,dx \tag{5.4}$$

where partial derivatives are taken with respect to the components of the ξ vector.

The Fisher metric allows the statistical model to be seen as a Riemannian manifold, where the metric is invariant under coordinate transformations. Relating this to our dataset, if 455 noise distributions corresponding to 455 neurons are generated, the noise distributions are implicitly embedding the neurons on a Riemannian manifold. The Fisher metric cannot be computed without assuming specific types of density function for the noise distributions. This is because it is based on taking the partial derivatives of known distributions with respect to known free parameters, meaning that other metrics must be used to visualize the arrangement of neurons in the space.

5.3 Multidimensional scaling

The question of finding coordinates for a set of elements arises when only the distances between elements of the set are available. A simple example is having only the distances between, say, twelve cities, and wanting to reconstruct a two dimensional map of their positions; an Euclidean embedding of the higher dimensional space the cities live in. Multidimensional scaling (MDS) is an algorithm used to address this question [37]. Its application to the twelve cities example results in a two-dimensional map which approximates the actual positions of the cities. The approximate nature of the method is due to the fact that we generally do not have access to the exact geodesics between the cities; we have approximate distances, from which we can construct approximate coordinates.

MDS finds coordinates for a set of elements, such that Euclidean distances measured between these elements approximate the geodesic distances measured in the elements metric space. This process is now described.

5.3.1 A matrix of distances

MDS takes as input a square matrix $d_X(i,j)$ of L^2 Euclidean distances between all elements in a dataset. In the case of distributions, each element is represented by some function f_i, so

$$d_{ij}^2 = \int [f_i(x) - f_j(x)]^2 \, dx \tag{5.5}$$

where f_i and f_j are kde estimates of the density functions associated with the noise distributions of the ith and the jth neurons. Here the L^2 distance is used for two reasons: first, taking L^2 distances achieves a level of consistency with the standard application of isomap. Second, differences between L^2 distances and Anderson-Darling distances, another suitable choice of distance, computed while hypothesis testing the χ-distribution, were not significant.

The matrix D_X will then be given

$$D_X = \begin{pmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,m} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m,1} & d_{m,2} & \cdots & d_{m,m} \end{pmatrix}$$

MDS aims to find a set of m coordinate vectors $v_1, .., v_m$, elements of a Euclidean space V, such that

$$||v_i - v_j|| \approx d_{i,j} \tag{5.6}$$

for all i, j in $1, .., m$.

5.3.2 Decomposing the matrix

The cosine law can be used to decompose the matrix D_X. This law simply relates the distances between the vertices of any triangle, and a contained angle, with each other. For a triangle with vertices i, j and k, with sides d_{ij}, d_{ik} and d_{jk}, and angle θ_{jik} between sides d_{ij} and d_{ik}, the cosine law gives

$$d_{ij}d_{ik}\cos\theta_{jik} = \frac{1}{2}(d_{ij}^2 + d_{ik}^2 - d_{jk}^2) \tag{5.7}$$

where we denote

$$b_{jik} = d_{ij}d_{ik}\cos\theta_{jik} \tag{5.8}$$

so that

$$b_{jik} = \frac{1}{2}(d_{ij}^2 + d_{ik}^2 - d_{jk}^2) \tag{5.9}$$

If vectors are used instead of distances, then the cosine law can be seen as a method of writing any vector in terms of the scalar dot product of two other vectors, which all together form a triangle, with angle θ_{jik}.

The cosine law can then be used to rewrite the matrix D_X as a matrix of scalar dot products, B. As B comprises of dot products it can be written in the form

$$B = XX^T \tag{5.10}$$

where X is a matrix and each X_i is a vector representing the coordinates of the ith cell, with the jth component of the ith cell given $(X_i)_j$.

5.3.3 Torgerson double centering

The quantity b_{jik} is centered on point i of the triangle. To allow for a more convenient representation, point i can be translated to the origin of the data, taken to be the center of mass of the data.

Take the matrix B_i to be the matrix of scalar products of elements of b_{jik}. The matrix B_i has origin at point i. This point i can be translated to the centroid of all points, termed double centering the matrix, turning the matrix B_i into the centered matrix B^* using

$$b^*_{ij} = b_{ij} - \frac{1}{n}\sum_k^n b_{ik} - \frac{1}{n}\sum_k^n b_{kj} + \frac{1}{n^2}\sum_g^n\sum_h^n b_{gh} \tag{5.11}$$

The restatement of the cosine law, Eq. 5.9, is used for substitution leading to

$$b^*_{ij} = -\frac{1}{2}\left(d^2_{ij} - \frac{1}{n}\sum_k^n d^2_{ik} - \frac{1}{n}\sum_k^n d^2_{kj} + \frac{1}{n^2}\sum_g^n\sum_h^n d^2_{gh}\right) \tag{5.12}$$

This is the centered matrix of dot products, centered using the column and row means of the original B_i matrix. This matrix is eigendecomposed:

$$B^* = U^* V^* U^{*T} \tag{5.13}$$

where the coordinates of the centered points are given

$$X^* = U^* V^{*\frac{1}{2}} \tag{5.14}$$

5.3.4 Coordinates

The X matrix is written in terms of its eigenvalues and eigenvectors, where for example if we had three points:

$$\begin{pmatrix} (X_1)_1 & (X_2)_1 & (X_3)_1 \\ (X_1)_2 & (X_2)_2 & (X_3)_2 \\ (X_1)_3 & (X_2)_3 & (X_3)_3 \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ u_{21} & u_{22} & u_{23} \\ u_{31} & u_{32} & u_{33} \end{pmatrix}$$

or in general:

$$(X_c)_l = \lambda_l u_{lc} \tag{5.15}$$

where $(X_c)_l$ is the lth coordinate component of the cth point. Through decomposing the B matrix we can find the eigenvectors and eigenvalues of the matrix, and hence can find a set of coordinates for each point in the set.

5.4 Isomap

A method such as MDS is needed to estimate the intrinsic geometry of the dataset. Isomap [38], an extension of MDS, can be used for this purpose. Isomap differentiates itself from MDS in that it calculates geodesics rather than Euclidean distances between elements of the dataset, thus estimating the intrinsic geometry of the space. Should the space be non-Euclidean, isomap is better suited to finding appropriate coordinates, rather than MDS which assumes an Euclidean space.

5.4.1 The nearest neighbour algorithm

Like MDS, isomap takes as input the square matrix $D_X = d_X(i, j)$ of L^2 Euclidean distances between all noise distributions from the dataset. It then edits this matrix to construct a second matrix $D_G = d_G(i, j)$ according to a nearest neighbour set of rules.

First, choose the value of a global variable, either K or ϵ, corresponding to the number of nearest neighbours or Euclidean distance that, if within this nearest neighbour number or distance, two points in the dataset are classed as nearest neighbours, Fig. 5.1. Set $d_G(i, j) = d_X(i, j)$ if i and j are linked by an edge, and $d_G(i, j) = \infty$ otherwise. This connects each neuron to its nearest neighbours using a Euclidean distance, but leaves out any large Euclidean distances, which presumably would not represent geodesics of the space.

Second, re-edit the matrix D_G, replacing all entries $d_G(i, j)$ by the minimum of $d_G(i, j)$ and $d_G(i, p) + d_G(p, j)$, where $p \in 1, .., n$ in turn, and n is the number of points in the dataset. D_G now contains the shortest path distances between all pairs

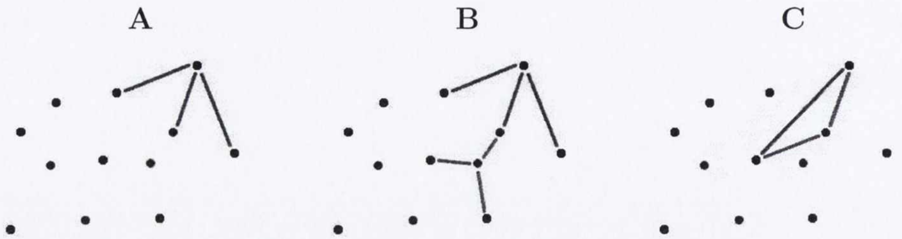

Figure 5.1: Evaluating the distance between two cells based on a nearest neighbour method. Here $K = 3$, so a cell is connected to its three nearest neighbours, see **A**. This process is applied to every cell, and only a sampling of this is shown here, see **B**. The distance between two cells is defined as the minimum of the sum of Euclidean distances between connecting cells, rather than just the direct Euclidean distance between the two cells, see **C**.

of points. Any large Euclidean distances have been replaced with distances that go through the network, thus approximating geodesics.

This matrix D_G can be seen as an unwrapped version of the original D_X. The standard example application of isomap is to a space where all points lie on the surface of a roll, see Fig. 5.2: in this case Euclidean distances between points would cut through the surface of the roll, and the geometry of the structure is lost. In contrast, a nearest neighbour approach would trace paths that are approximately along the surface of the roll, reflecting the true geodesics of the space.

The accuracy of isomap increases asymptotically with increases in the number of data points. The denser the sets of data points, the smaller the Euclidean distances measured between data points and the more accurate the structure estimation becomes.

5.4.2 Optimal dimension

The approximate dimension of the data is determined by the number of eigenvectors used to construct the Euclidean space embedding. To achieve this we firstly eigendecompose D_G. We are looking for a subset of coordinate vectors v_i for points in a Euclidean space V, such that, with minimal error, the subset minimize the cost

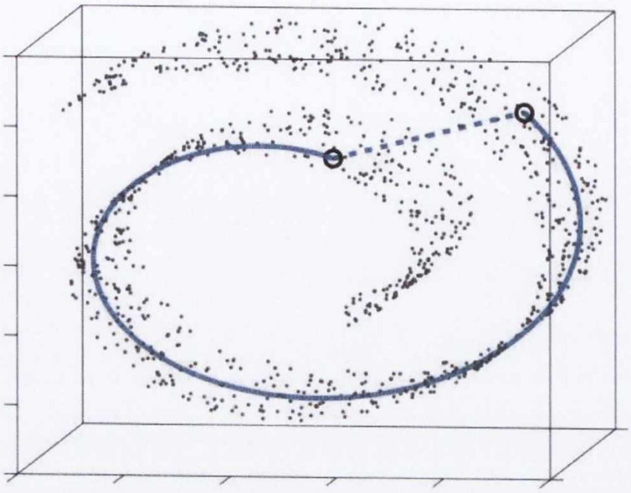

Figure 5.2: The standard example application of isomap to a rolled space. For the two points shown here, instead of treating the geodesic between them as being their Euclidean distance, which clearly ignores the structure of the space, the geodesic is taken to be the minimum distance through other connected points along the surface of the space. This image is taken from [38].

function

$$E = \|\eta(D_G) - \eta(D_V)\|_{L^2} \tag{5.16}$$

where $D_V = \{d_V(i,j) = \|v_i - v_j\|\}$ represents the Euclidean embedding, $\|A\|_{L^2} = \sqrt{\sum_{i,j} A_{i,j}^2}$, and η performs Torgerson double centering on the matrix.

If the embedding is constrained to p dimensions· it can be shown that the top p eigenvectors of $\eta(D_G)$ achieve a global minimum for the cost function. Obviously E decreases with increasing p, where p represents the dimension of the Euclidean space reconstruction.

The residual variance, defined as

$$r = 1 - R^2(\eta(D_G), \eta(D_V)) \tag{5.17}$$

where $R(X, Y)$ determines the linear correlation coefficient between entries X and

Y, is used as a measure of the error between $\eta(D_G)$ and reconstructed $\eta(D_V)$. The residual variance r originally proved useful as a way of evaluating the difference in performance between isomap and other MDS methods, and has subsequently been adopted as the primary measure of error.

Interpreting the residual error is relatively straightforward. If the data is two-dimensional then trying to embed the data in a one-dimensional Euclidean space will produce a high residual error. On the other hand, if the two-dimensional data is embedded in a three-dimensional Euclidean space then there should be little change in residual error values between results from the two and three dimensional residual error values.

Unfortunately isomap does not yield an exact method of estimating the dimension of a dataset, but rather provides a graph of residual error against dimension. This leaves it to the analyst to decide upon the value of dimension where increases in dimension fail to produce sufficient decreases in residual error. The residual error rarely drops to near zero once the actual dimension has been found, but rather tails off in slow decline. Generally it is the 'elbow' of this curve that is taken as the datasets dimension, where the residual error ceases to decrease notably with increasing dimension.

5.4.3 Nearest neighbour choice

To decide which points should be regarded as connected to each other one must decide on either a classification according to nearest connected neighbours, K, or nearest neighbour radius, ϵ. For example, if we set $K = 3$ then each data point will be connected to its three nearest data points under the L^2 Euclidean distance. If instead $\epsilon = 1.4$ is used then each point is connected to all data points within a L^2 radius of 1.4 units.

Like the selection of the optimal dimension, the choice of whether to use K or ϵ is empirical, as is the ideal value that should be selected. In the limit of maximum

K, if every point is linked to all others then isomap has been reduced purely to MDS and local geometry, should it have been non-Euclidean, is lost. In the low K limit, if each point is only connected to one or two of its nearest neighbours, then unless a dense, large dataset is used, the data will be viewed as islands of points separated by discontinuities, again breaking the geometric structure. The choice of whether to use K or ϵ, or indeed their specific values, is less and less significant as the data size and density is increased.

5.5 Metric normalization

The free parameter τ, used in the van Rossum metric, reflects the combination of rate and temporal coding detected in a neurons spike trains. As this quantity varies from neuron to neuron, the value of τ must be estimated for each neuron in the dataset, rather than taking an overall average value.

Motivation for this lies in the fact that the data used is from a number of different brain regions and cell types, so there is no reason to assume τ should be fixed. Different values of τ scale the metric distances, which would make comparisons between different noise distributions difficult due to different τ values.

To solve this problem the metric is normalized so that the L^2 distance measured between a spike train function containing a single spike and another containing no spikes is one unit:

$$A \int_0^\infty f^2(x)\, dx = 1 \tag{5.18}$$

where

$$f(t) = \sum_i e^{\frac{-(t-t_i)}{\tau}} \tag{5.19}$$

Through basic integration we find

$$A = \frac{2}{\tau} \tag{5.20}$$

Hence under normalization the van Rossum distance becomes

$$D^2(f,g) = \frac{2}{\tau}\int [f(t) - g(t)]^2 \, dt \tag{5.21}$$

It is this normalized form of the metric that will be applied to the dataset.

5.6 The space of Gaussian distributions

To learn more about the algorithm itself, isomap was first applied to a known statistical model. 1440 datasets were generated, where all 200 values in each set were drawn from a Gaussian distribution. Again the Box-Muller transform was used for this, and the elements of each dataset all had the same mean and standard deviation. Gaussian means were chosen in the range $[0.1, 5.0]$ with increments of 0.1, and standard deviations were in range $[0.1, 3.0]$ with increments 0.1.

Kde was used to estimate the distribution of each dataset. A matrix, D_V, of L^2 distances was then constructed. For example, $d_V(i,j)$ denotes the L^2 distance between kde estimate of the ith cell and the kde estimate of the jth cell. Isomap takes as input this matrix of inter-distribution distances.

Key to this idea is evaluating the performance of isomap under noisy conditions where the true result is already known. Isomap itself has proved a controversial tool for tackling noisy data [39], and there are no obvious methods of selecting K and the dimension of the space without a priori information of the space itself. Small changes in the value of K or ϵ can lead to large changes in the geometry of the space. This is because isomap is vulnerable to short circuit errors, where the nearest neighbours have been incorrectly identified, which can lead to many entries in the matrix of geodesic distances being incorrect, generating a different estimated geometry.

It was found that varying the nearest neighbour parameters had negligible effect on determining the optimum dimension, see Fig. 5.3. However, visualization of the

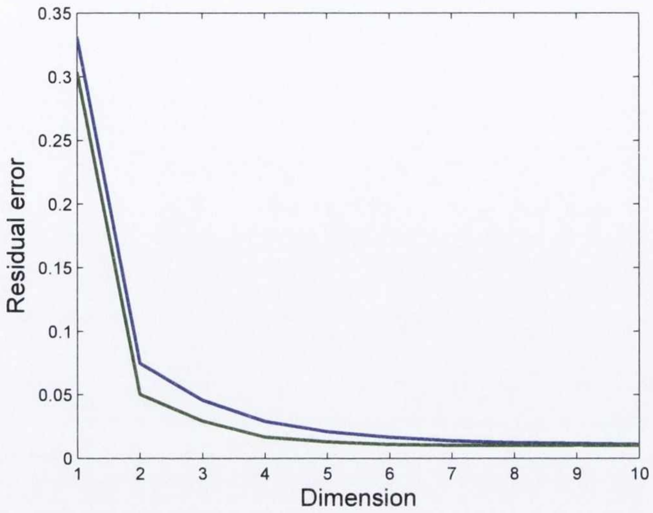

Figure 5.3: A residual variance graph for $K = 60$ in blue, and $K = 6$ in green. Here, isomap has been applied to the space of Gaussian distributions. The x-axis denotes the number of eigenvectors used to construct the space, that is, the dimension of the space. The y-axis denotes residual variance. In this case we expect the space to be two dimensional, and the similarity between the two graphs demonstrates the lack of sensitivity of the Gaussian space dimensionality to changes in K. The dimension of the space is clearly identified as two.

Euclidean embedding was found to be distorted from its true embedding, see Fig. 5.4. Given enough data, the points representing Gaussian distributions should line up in a band, where the position of each point along the bands axes is determined by its mean and standard deviation values. This distinguishes the bands axes as a linear scaling of the mean and standard deviation. In reality this band is found to be both bunched and curved.

5.7 Data

Data used here is again from the large extracellular zebra finch dataset made available on the Collaborative Research in Computational Neuroscience database by the Frederic Theunissen laboratory at UC Berkeley [40]. Data is obtained from 455 neurons throughout eight regions: L, L1, L2a, L2b, L3, Mld, CM and OV, of zebra

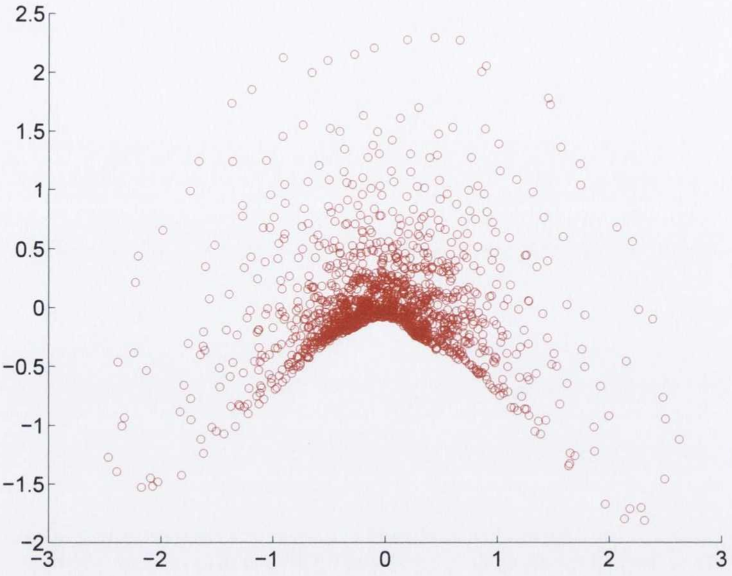

Figure 5.4: A two dimensional embedding of the statistical model of Gaussian distributions, with $K = 6$. We can see isomap fails to unwrap the space fully, demonstrated in the curvature of the lower section of the graph. In theory the graph should should have an uncurved, band like structure. The x-axis should distinguish the free parameter μ of the Gaussian distribution, and the mapping between the x-axis and μ should be linear, instead of the quadratic form demonstrated here. Varying the K parameter produces only very minor changes in the space. This information can be used when considering the noise space of neurons: less precedence should be placed on the choice of K, and we should expect the axes of the free parameters to be curved relative to the x and y axes that isomap identifies.

finch auditory forebrain during playback of conspecific song.

5.8 Construction and dimensionality of noise space

By computing only same stimulus spike trains for the dataset a total of 900 noise distances can be associated with each neuron. The probability density function of this distribution of distances can then be estimated using kde. Thus, each neuron has an associated empirical noise distribution with some unknown number of free parameters, and the set of noise distributions represent elements of the underlying statistical model.

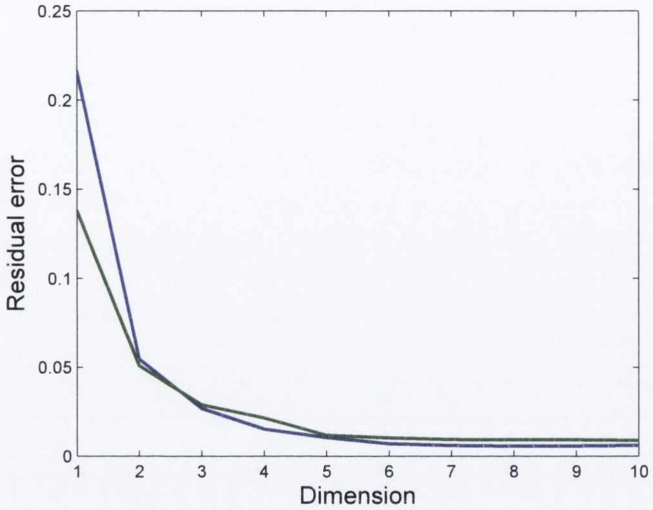

Figure 5.5: Plots of residual variance against dimensionality for $K = 6$ in green and $K = 60$ in blue. Data comprises noise distributions generated from the neuronal dataset. As can be seen, the largest decrease in error occurs when two rather than one eigenvectors are used. Introducing more eigenvectors just leads to a tailing off of error. Hence the data is seen to be highly two dimensional. For $K = 60$ the decrease in error is strongest when the dimension of the coordinate space in increased from one to two. Also for $K = 60$ the elbow of the graph, where this tailing off takes place, is easier to see as having an x-axis value of two.

Again square roots of the L^2 distances between all noise distributions from the dataset are computed, generating a matrix of inter-noise distribution distances. It is important to note the distinction that these distances are not between spike trains, but between distributions of spike train distances.

The collection of 455 cells is found to change little with respect to K, with values of $K \in \{6, .., 455\}$, see Fig. 5.5. The graphs do however indicate a dimensionality of two for the space, as can be seen from the 'elbow' of the graph. When K is decreased below a value of six the density of neurons is no longer sufficient to accurately estimate the local structure, causing the manifold to fragment into disconnected clusters.

The isomap algorithm allows for our first graph of the noise space of neurons, see Fig. 5.6. Strong similarities exist between the graph of the noise space, and the

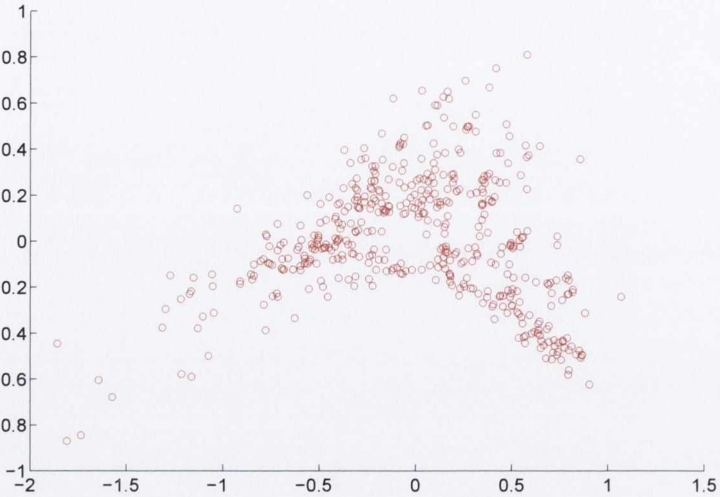

Figure 5.6: A two dimensional plot of the space of neurons, with $K = 6$. It can be seen, from the networks of neurons used here, that neuronal noise is typically limited to exist within certain bands. The meaning of these bands will be explored in the next section. It should be noted that little variation in the structure of the space presents itself as K, the number of connected nearest neighbours, is varied in the isomap algorithm.

graph of the space of Gaussian distributions. Both demonstrate a strong curvature in their graphs, indicating the isomap embedding is approximate rather than a direct mapping from free parameter to x and y axis. It is interesting to note both regions of outliers, and regions of increased density. The meaning of this map will now be explored.

5.9 Visualization of noise space

Isomap is used to visualize the noise space, but furthermore the distribution of neuronal parameters can also be examined in this space. For example, k and σ, the free parameters of the χ-distribution, can be calculated from the moments of each noise distribution:

$$k = \frac{2\langle x^2\rangle^2}{\langle x^4\rangle - \langle x^2\rangle^2}. \tag{5.22}$$

and

$$\sigma^2 = \frac{\langle x^2 \rangle^2}{2\langle x^4 \rangle} - \frac{\langle x^2 \rangle^2}{2} \tag{5.23}$$

where $\langle x \rangle$ denotes the expected value of x.

Each data point in the noise space can then be coloured relative to its k value using the standard `matlab` colour scale, with red denoting the highest values of the set and blue the lowest. An obvious advantage of doing this is that it allows one to see how each parameter correlates with the axes isomap has isolated.

If k is indeed one of the coordinates parametrizing the noise distribution, then it should scale linearly accordingly, visualized through the colour scale mapping, along one of the graphs axes. Likewise for σ, which should be distinguished along the other axis.

The following plots demonstrate how isomap singles out the dependence of k and σ on the x and y axes respectively, see Fig. 5.7 and Fig. 5.8. For example, in the first graph each point is coloured according to its k value, and the variation in colour can be seen relative to the y-axis. In this instance, isomap indicates that k can be choosen as a free parameter of the noise distribution. The graphs also demonstrate that, although clear dependencies between the axes and free parameters can be seen, better choices for the coordinate parameters can be made, which will vary linearly with respect to the x and y axes. This should not be taken as a failing arising from fitting the noise space with χ coordinates. Instead it should be seen as, for the purposes of viewing the space, more convenient coordinates should be found.

5.10 μ and k

The average value of a variable drawn from a noise distribution, μ, would seem a natural choice of parameter for the noise space. Its meaning is intuitive: the mean of each noise distribution represents the average distance between two spike trains corresponding to the same stimulus. This reflects the average level of noise

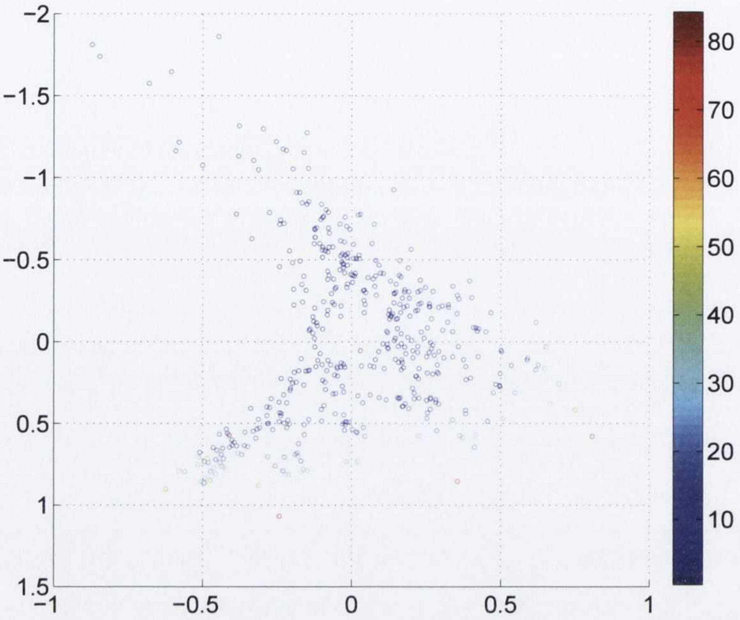

Figure 5.7: Values of k, one of the free parameters of the χ-distribution, are mapped in the form of colour onto the noise space plot. Values of k are given by the colourbar using the standard `matlab` colour scheme. The graph has been rotated to help view the k-axis dependence. Values of k are clearly not sensitive to small changes in the y-axis. While this will be discussed in more detail, one can infer that the choice of k as one of the coordinates of the noise space is neither convenient nor particularly informative, and preference should be shown for another coordinate system.

associated with the neuron.

We can plot the mean of each noise distribution, in colour, on the noise space graph, see Fig. 5.9. As shown in the graph, the same bottom right arm has an increased level of noise. Increases in the level of noise scale linearly with the x-axis, as can be seen in the colour variation relative to the principal axis isomap has distinguished.

A small change in μ leads to large change in k, one of the two free parameters from the χ-distribution. This is shown through integrating the non-standard χ:

$$< x >= \frac{1}{Z} \int x^k e^{\frac{-x^2}{2\sigma^2}}\, dx \qquad (5.24)$$

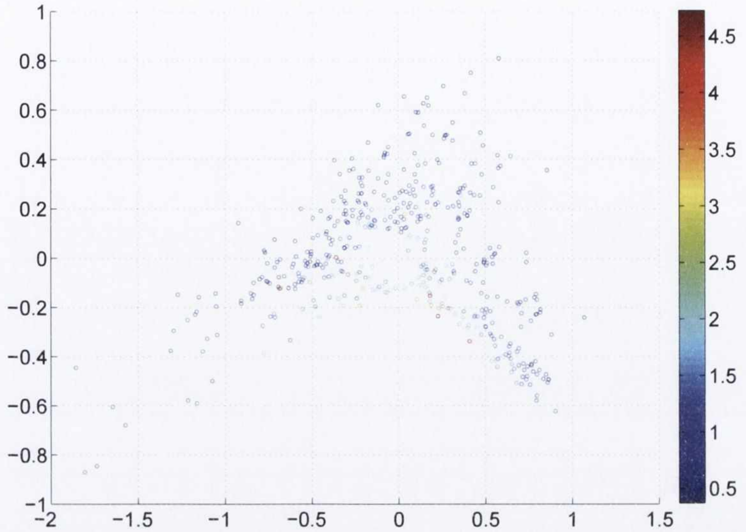

Figure 5.8: Values of σ, the second free parameters of the χ-distribution, are mapped in the form of colour onto the noise space plot. Values of σ are given by the colourbar using the standard `matlab` colour scheme. Like the plot of k, a clear order is found in the colouring of the cells, with cells aligning along the y-axis in accordance to decreasing σ values. However, like k, the change in σ with respect to the y-axis is not particularly strong, implying another parameter would prove a better choice of coordinate.

where

$$Z = \sigma^k 2^{\frac{k}{2}-1}\Gamma(\frac{k}{2}) \tag{5.25}$$

and

$$\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}\,dt \tag{5.26}$$

Substituting $y = x/\sqrt{2}\sigma$ yields

$$< x >= \frac{(\sqrt{2}\sigma)^{k+1}}{Z}\int y^k e^{-y^2}\,dy \tag{5.27}$$

Carrying out the substitution $y^2 = t$ yields

$$< x >= \frac{(\sqrt{2}\sigma)^{k+1}}{2Z}\int t^{\frac{k}{2}-\frac{1}{2}}e^{-t}\,dt \tag{5.28}$$

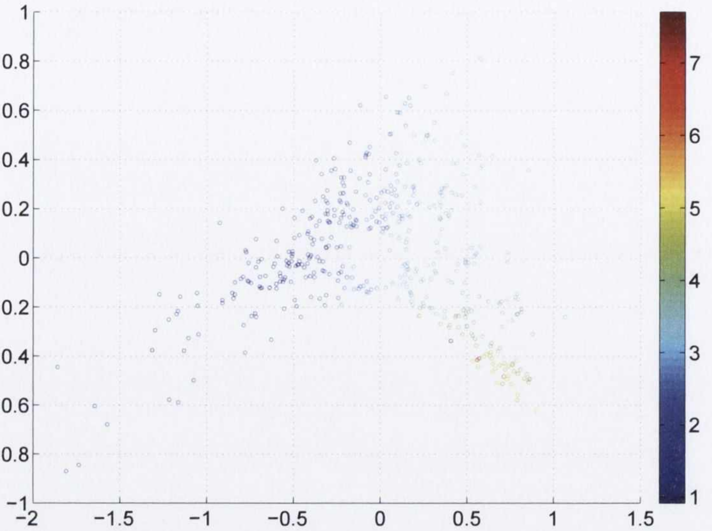

Figure 5.9: A plot of the noise space of neurons, this time coloured by the average level of noise associated with each neuron. The cluster on the bottom right arm, the same cluster with elevated firing rates, are also seen to have the highest levels of noise amongst the neurons in the space.

This is now in a form that can be written in terms of Γ functions, and carrying out the integration gives

$$< x >= \sqrt{2}\sigma\frac{\Gamma(\frac{k+1}{2})}{\Gamma(\frac{k}{2})} \tag{5.29}$$

As can be seen in Eq. 5.29, for fixed values of k the value of μ varies linearly with σ. The dependence of μ on k is more complicated, and a plot for values of k typical of the dataset for $\sigma = 0.5$ is provided, see Fig. 5.10.

This relation demonstrates that a small variation in μ produces a large variation in k. If the dataset is viewed in terms of its μ colouring a more gradual change can be observed, relative to the non-linear jumps in the k colour. This points to the idea of μ being a more natural choice of parameter than k.

For values of k typical of this dataset a change in k will produce a non-linear change in μ. Hence, instead of visualizing the space in terms of k, which for the purposes of this dataset can be uninformative due to the general lack of variation of k, the dataset can be plotted with each neurons μ value coloured. The quantity

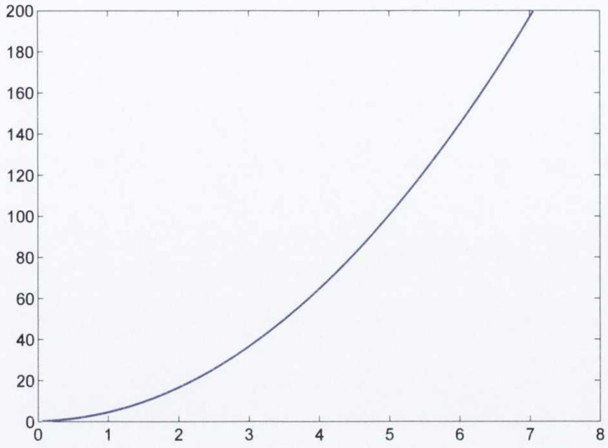

Figure 5.10: A plot of μ against k. Small changes in the value of μ produce large changes in the value of k. As k varies little over the dataset, μ can be seen as a more natural parameter choice as it is better at highlighting changes in the noise levels of neurons.

μ is useful in itself due to its simple and intuitive meaning: it measures the average noise associated with a neurons spike trains since it is the average value of the noise distances.

5.11 Noise throughout networks of neurons

The dataset consists of neuronal output, where each neuron is from one of eight networks of neurons, with some exceptional cases where the neurons network was not identified. We have seen how each neuron can be coloured according to its noise level. As a neurons position in the noise space characterizes the noise associated with it, the distribution of noise amongst a particular network can be visualized.

The above procedure can be taken one step further. Since the dataset comprises networks of neurons, the position of every neuron in the dataset can be plotted and coloured according to its network type. The resulting plot provides a visualization of the distribution of noise *between* networks of neurons. This represents the distribution of noise throughout networks of neurons.

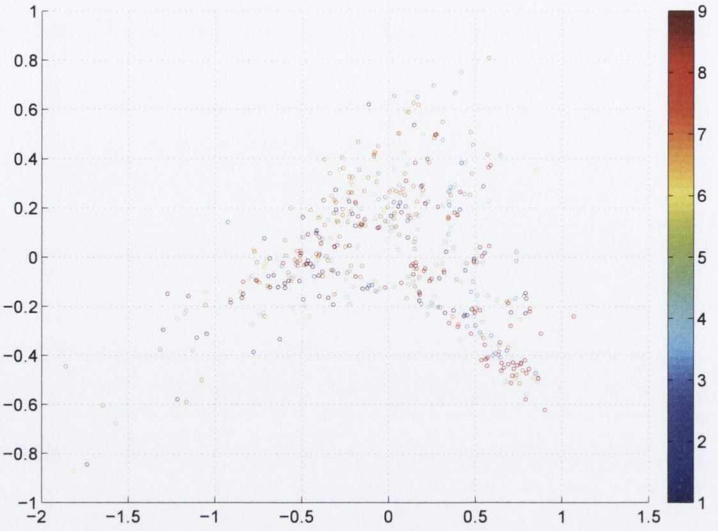

Figure 5.11: A visualization of the noise space of neurons, with each neuron coloured according to the network to which it belongs. The networks are labelled L = 1, L1 = 2, L2a = 3, L2b = 4, L3 = 5, CM = 6, Mld = 7, OV = 8, 'none' = 9. Here we observe that the noise associated with each network is distributed reasonably evenly across the noise space.

This procedure was applied to zebra finch auditory networks identified by experimentalists as L, L1, L2a, L2b, L3, CM, Mld, OV. A ninth type is termed 'None', representing the set of neurons from the dataset that have not been successfully classified during recording and processing. As is seen in Fig. 5.11, at first no obvious clustering according to network type is observed. However, one particular network can be highlighted and the others coloured uniformly. Eight of the types have neurons with noise characteristics that are distributed similarly throughout the noise space. One network of neurons, those of OV, are found to be primarily located in the bottom right arm, see Fig. 5.12.

5.12 Artificial noise space

As the dataset consists of 455 neurons, the noise space consists of 455 points. It is useful to generate more points artificially and scatter them through the space, which

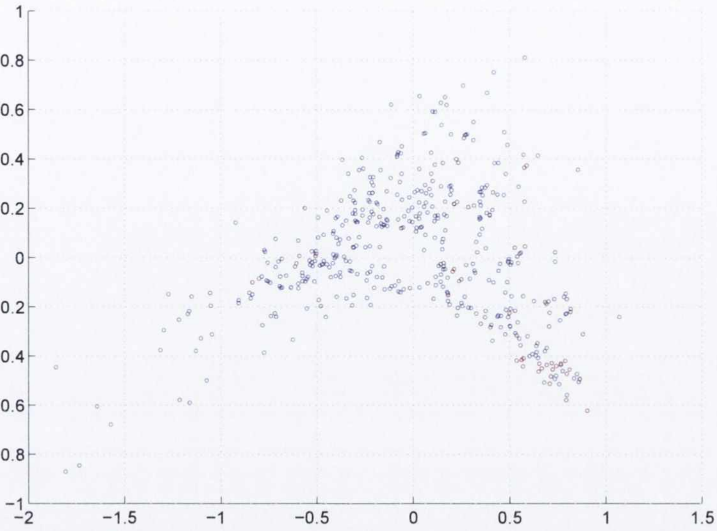

Figure 5.12: The network OV is highlighted in the noise space using colour coding L = L1 = L2a = L2b = L3 = CM = mld = none = 0, OV = 9. It is noted that OV is in general restricted to the right of the x-axis equalling zero mark, and is primarily located in the bottom right arm.

would 'fill out' the space. This must be done without letting the artificial points distort the calculation of the space itself. The idea is to extract the eigenvectors of the original noise space, and distribute artificial points throughout this space. This should, at the very least, give a more complete view of the noise space.

5.12.1 Coordinates of an artificial data point

Decomposition of the B^* matrix allows Euclidean coordinates to be found for the data points between which the distances are measured. If the distance between the artificial points distribution and all the real points distributions were computed and allowed to form part of the D_G matrix, then the generation of the noise space would be distorted by the artificial point. Here instead a method is presented to write the artificial points coordinates in terms of the eigenvectors of the original noise space.

Let b_{ac} denote the dot product of the vector of coordinates of the artificial point

X_a and the real point X_c:

$$b_{ac} = X_a.X_c \tag{5.30}$$

or component wise:

$$b_{ac} = \sum_l X_{al}.\lambda_l u_{lc} \tag{5.31}$$

The eigenvectors of c can be removed from the right hand side:

$$\begin{aligned} \sum_c b_{ac} u_{lc} &= \sum_c \sum_l X_{al}.\lambda_l u_{l'c} u_{lc} \\ &= \sum_c X_{al}.\lambda_l \end{aligned} \tag{5.32}$$

Let $\beta_l = \sum_c b_{ac} u_{lc}$. Again u_{lc} is just the cth component of the lth eigenvector for the noise space, which has already been obtained through the application of isomap to the noise space. This allows for the computation of β_l. To calculate b_{ac} we use the equation used to compute the elements of B^*, Eq. 5.12, where a is substituted for the ith index

$$b^*_{aj} = -\frac{1}{2}\left[d^2_{aj} - \frac{1}{n}\sum_k^n d^2_{ak} - \frac{1}{n}\sum_k^n d^2_{kj} + \frac{1}{n^2}\sum_g^n \sum_h^n d^2_{gh}\right] \tag{5.33}$$

and the distances between the artificial points distribution and all the real points distributions have been calculated, forming d_{aj} for $j \in \{1, .., n\}$.

Noting $\sum_c X_{al}.\lambda_l = cX_{al}.\lambda_l$ and substituting into Eq. 5.32 results in

$$X_{al} = \frac{\beta_l}{c\lambda_l} \tag{5.34}$$

This equation then allows us to compute the lth coordinate of the artificial point X_a, where $l \in \{1, .., n\}$ and there are n real cells. Thus any number of artificial points can be created, with coordinates generated from the β and λ vectors.

5.12.2 Landscaping the noise space

If data is drawn from a specific distribution, and kde used to estimate the distribution, then using Eq. 5.34 this distribution can then be plotted as a point in the noise space. For example, say data is drawn from a Gaussian distribution with free parameters μ_1 and σ_1. The kde of this data is calculated and L^2 distances between this and every real noise distribution computed. Combining this with a knowledge of the eigenvectors and eigenvalues of the original noise space, obtained using isomap, allows for the coordinates of the Gaussian to be plotted.

This process can be repeated many times, and in each case by choosing a different (μ, σ) the space can be filled with points. Here approximately 3000 points are drawn from Gaussian distributions with varying μ and σ. In each case the distance between the artificial distribution and all the real distributions is computed. This process is demonstrated in Fig. 5.13 and Fig. 5.14.

One might question why a Gaussian, rather than a χ-distribution, was used to generate the artificial points. The answer is that, for the purposes of coding, a Gaussian distribution is simple and efficient to implement. Also, for high values of the free parameter k, typically values higher than $k = 4$, the Gaussian distribution acts as an excellent envelope function for the χ-distribution. Hence it can, for certain values of k, be used as an alternate distribution to sample from. Each random variable which contributes to one of the 200 points that make up the artificial kde will be generated by one Gaussian, not the square root of the sum of the squares of many Gaussian random variables.

5.13 Firing rates, the noise space, and OV

The firing rate of a neuron, r, can be defined as the average number of spikes per spike train of length L:

$$r = \frac{1}{mL} \sum_{i=1}^{m} N(s_i(L)) \tag{5.35}$$

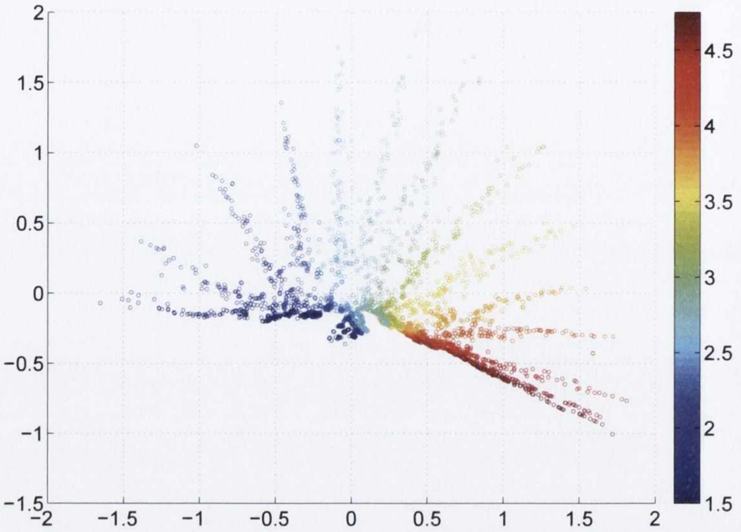

Figure 5.13: A graph of approximately 3000 artificially generated points, scattered throughout the noise space of neurons. Each point is generated using a Gaussian distribution, with values of μ in the range $[1.5, 5]$, and values of σ in the range $[0.2, 3]$. Here each point is coloured according to its μ value, with values given by the colour scale. As can be seen from the colouring, each arm of the graph is given by some fixed value of μ, and values along that arm correspond to variations in the value of σ, as will be seen in Fig. 5.14.

where $N(s_i(L))$ just denotes the number of spikes in the ith spike train s of length L. For the dataset used here $L = 1$ second. Hence a firing rate can be associated with each neuron, and in terms of the space the firing rates can be super-imposed in colour onto each data point, seen in Fig. 5.15. As can be seen in the graph, most neurons have the same firing rate of approximately ten spikes per second, with a second group of neurons, eight percent seen on the bottom right arm, with firing rates above thirty spikes per second.

It is interesting to compare the firing rates with neuronal noise. We know the mean of each noise distribution reflects the average level of noise associated with the neuron. Once again, we can plot the mean of each noise distribution, in colour, on the noise space graph, see Fig. 5.9. As shown in the graph, the same bottom right arm has an increased level of noise. This region is also dominated by neurons

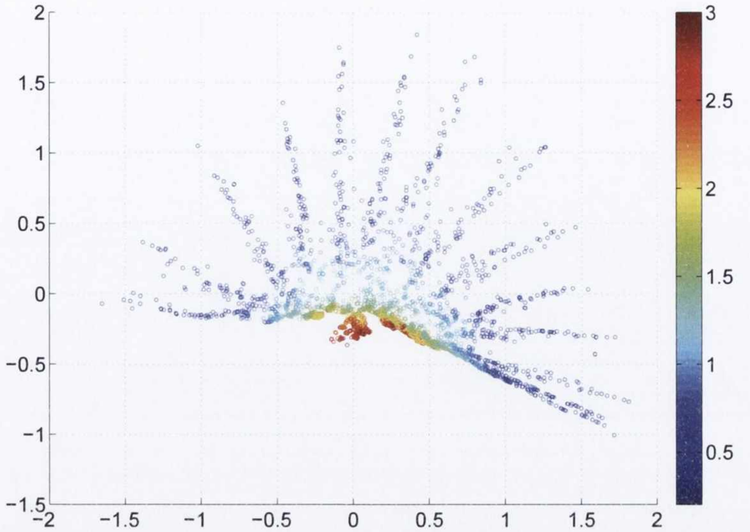

Figure 5.14: Another plot of the scattering of 3000 artificial points into the noise space. Here each arm of the graph represents a fixed value of μ, chosen from the range $[1.5, 5]$. In this case it is σ that is coloured, with values chosen in the range $[0.2, 3]$, given by the colourbar. As can be seen, σ controls the position of the artificial point in a particular arm.

from OV, implying that neurons from this network typically have increased levels of noise and firing rate.

The network OV has been previously studied experimentally, where it was found that neurons from the OV network have higher firing rates and higher neuronal variability than either Mld or Field L neurons, [52]. This is consistent with the findings of the noise space of neurons, which provides an additional mathematical backdrop to these experimental results.

5.14 Discussion

The embedding of neurons on a manifold allows for manifold based mathematics to be applied to neuronal data. Information theory is an example of an area where manifolds are needed, and it is natural to apply information theory directly to these neuronal distributions. Here the difficulty of finding a natural language to describe

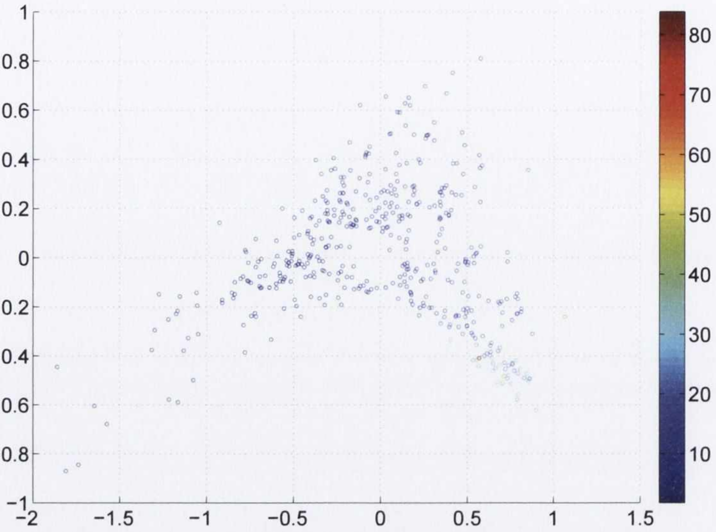

Figure 5.15: A plot of the noise space of neurons, coloured by the firing rate of each neuron. The bottom right arm highlights a group of neurons, approximately eight percent of the population, with increased firing rates, given by the colour scale. The cluster on the bottom right arm, as was seen previously, also have the highest levels of noise amongst the neurons in the space.

spike trains vanishes, as instead the properties of spike trains are used to generate a distribution that describes the neuron itself. The entropy of a noise distribution can be computed, as well as the mutual information between two noise distributions. This is combined with the Fisher metric, which defines the geometry of the space.

It is unrealistic to assume the noise distributions are stationary. The free parameters of the noise distributions clearly change over time. The evolution of the free parameters can be seen as the tracing out of paths along the noise distribution manifold. It would be interesting to observe the paths traced by the distributions. Obviously these paths are bounded by the limits of the free parameters of the distributions, however the opportunity exists to look for periodicities or limit cycles in the motion of the distributions.

When isomap was applied to the space of Gaussian distributions, curvature was found in the resulting band. Three factors could contribute to this curvature. First, noise is certainly present, which as previously discussed has a distorting effect on isomap. Second, the number of points and the density of points in particular regions will influence isomap. Third, the use of L^2 inter-distribution distances, rather than some other measure of distance, may effect isomap. It would be interesting to experiment with different distance measures to see how this effects curvature. While the Fisher metric may seem an obvious choice, this requires knowing the explicit form of the noise distribution, something we don't have. An alternative would be the KL divergence, which while not considered here, could prove more suitable than the L^2 distance in reducing curvature.

It is important to note the difference between noise distributions and the jitter and unreliability associated with spike trains. This noise distribution is a distribution of distances, not spikes or spike trains. Our basic 'unit' is a distance. This of course is a natural side effect of taking a geometrical approach to the analysis of spike trains. Hence, if we compute the entropy of a noise distribution, we are computing the entropy associated with the noise distances. While these reflect the noise found in spikes, the distinction must be made between the two. Information theory is phrased in terms of distances, not spikes.

Using methods such as isomap we can produce a map of the noise space of neurons. The coordinates of each point are assigned based on how noisy the neuron is. Hence this is already a way of visualizing the noise properties of a network of neurons. It is easy to see the 'noise categories' into which each neuron in the network falls. Groups of networks of neurons have been plotted on the same map, and differences in the noise properties of the neurons examined. Here the differences between networks were found to be rather insignificant for the zebra finch data, with the exception of OV. Approximately ten percent of the neurons displayed both high firing rates and high levels of noise. Neurons with lower firing rates did not display such noise.

Based on the noise, firing rate, and proportion found, it may be reasonable to assume these are inhibitory neurons. However, the noise space highlights this region as being dominated by network OV, indicating that the noise space has isolated a cluster of neurons according to network type. A split in the noise space related to whether a neuron is inhibitory or exhitatory is also possible. To confirm this, at the very least, requires having the waveforms of the spikes.

Chapter 6

Discussion

Throughout this work the natural geometry of spike trains has been used analyse tangible quantities related to information theory and noise. The aim, in virtually every case, was to help build a new mathematics of spike trains. By identifying a geometrical structure consistent with the known properties of spike trains, issues fundamental to spike train analysis can be re-examined through this new framework.

Information theory, which is phrased in terms of random variables, was rephrased in terms of distances between random variables. In doing so we created an instance of the concept that any normed vector space is by definition a metric space. The key issue however was not related to this rephrasing, but rather to investigating how similar the metric space of information theory was to the metric space of spike trains.

The noise properties of spike trains from zebra finch auditory forebrain were found to be well modelled by a χ-distribution. Again, the χ-distribution is the distribution that distances between the Gaussian noise vectors of information theory follow. Hypothesis testing, our only tool for such a task, was used to test for χ-distributed noise distances. The strongest positive statement that can arise out of such a test is that 'there is insufficient evidence to prove the data was not sampled from the hypothesised distribution.' For the data considered, this was found to be the case.

The original channel capacity formula from information theory is based on the assumption of additive Gaussian noise. There is no reason to believe that spike train noise should in general behave in such a way, even though this was found to be the case in the dataset we considered. As a consequence, it is necessary to develop this capacity approach to encapsulate both additive and multiplicative noise, thus producing a more robust model.

The question of whether information theory itself is a suitable framework for the analysis of spike trains must be considered. Here this was not addressed; instead, since information based quantities are commonly computed during the analysis of spike trains, the idea was to show equivalent quantities could be calculated in the metric space of spike trains. However the use of these quantities, in this case capacity, was not addressed. For instance, the bimodal distribution of capacities found during the analysis could point to a split between excitatory and inhibitory neurons, but without further information relating to the dataset this cannot be confirmed. It was the nature in which information theoretic quantities should be calculated, rather than the quantities themselves, that was proposed.

On more familiar territory, a metric space examination of neuronal noise was found to shed new light on the form of the jitter distribution. The Victor-Purpora metric was used to isolate variations in the temporal structure of spike trains corresponding to the same stimulus. The result was unusual in the sense that, instead of being Gaussian as could be expected, it was found to follow a hyper-Laplace distribution. Another unexpected result is the bimodality of the distribution, indicating that two timescales are needed to describe the jitter.

Working from the result that jitter is hyper-Laplace distributed, a spiking model was proposed to provide one explanation for this behaviour. This involved what was termed a 'two state' neuron model. This model consisted of a neuron that operated in one of two firing states, each modelled by a separate Poisson process.

This was justified through firstly considering a Poisson process firing rate model. By definition, waiting times for spikes occurring from such a process are exponentially distributed. The time differences between spikes from two trains generated by the Poisson process are then, by definition, Laplace distributed. In our case we have found that these time differences, the jitter, are hyper-Laplace distributed. One explanation for this is that the neuron generates spikes according to one of two Poisson processes, which it flips between. Time differences between spikes would then obey one of two Laplace distributions, with the overall distribution given by a hyper-Laplace.

The field of neuroscience typically requires direct application of any theoretical models. It was thought that, given the new χ-distribution and hyper-Laplace models of neuronal noise, the models could be used to generate spike trains with noise drawn from these distributions. This was found to be the case: a 'template' spike train can have its spikes deleted with a certain probability, jittered according to the associated hyper-Laplace, and spikes inserted using a Poisson process. The noise found in the resulting spike train is then tested, using rejection sampling, to see if it is typical of noise found in the dataset. If this is found to be the case it is kept, else it is discarded and the process repeated.

Lastly, the geometrical implications of noise distributions were examined. Associating a noise distribution with a neuron means implicitly embedding it on a manifold. Embedding data on a manifold allows for coordinate systems to be associated with the neurons themselves, rather than the spike trains. This allows for coordinate based analysis such as information theory and information geometry to be applied to the neurons, rather than spike trains.

Key to this is the idea that instead of focusing on a specific description of a spike train, it is better to use the metric properties of spike trains to define an overall state of the neuron. For instance, a mathematical framework now exists to calculate quantities such as the entropy of a neurons noise distribution, the mutual information of two neurons noise distributions, and the Fisher information between any two known distributions.

A geometrical approach to spike train analysis means considering neuronal activity in a very different way. Neuronal activity is now thought of in terms of distances and not spikes. Our 'basic unit' is not a spike, but a distance. The noise distribution is by definition a distribution of distances, not of spikes. Quantities such as jitter and unreliability arise from distance calculations.

In a metric space approach similarities are defined in terms of both firing rate and temporal structure, with the metrics free parameter giving a weighting to both, based on the dataset under examination. As a rich backdrop, while spike trains exist in metric spaces their neuronal generators trace out paths on manifolds, with coordinates given by many of the observable parameters we have come to associate with the neurons themselves.

Appendix A

Kernel density estimation

A.1 Kernel density estimation

Kernel density estimation is a method of estimating the probability density function associated with a dataset. The resulting function is essentially a smoothed histogram, and provides a method of examining the density of a dataset without trying to fit a known distribution.

A.1.1 Density estimation

A probability density function, denoted $f(x)$, can be defined relative to a random variable X:

$$f(x) = \lim_{h \to 0} \frac{1}{2h} P(x - h \leq X \leq X + h) \tag{A.1}$$

where $x \in X$. This probability can be approximated by what is termed here as the 'naive estimator' $\hat{f}(x)$, where

$$\hat{f}(x) = \frac{1}{2hn}(\text{number of } (X_1, .., X_n) \text{ falling in } (x - h, x + h)) \tag{A.2}$$

This can be re-expressed using a simple weight function:

$$w(x) = \begin{cases} 0.5 & \mid x \mid \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{A.3}$$

This implies

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} w\left(\frac{x - X_i}{h}\right) \tag{A.4}$$

An immediate difficulty with this choice of weight is that $\hat{f}(x)$ is non-continuous, producing a stepwise graph. This can be avoided by using a smooth kernel K, a normalized density function:

$$\int_{-\infty}^{\infty} K(x)dx = 1 \tag{A.5}$$

which when substituted gives

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{(x - X_i)}{h}) \tag{A.6}$$

Here h is the 'window width', also called 'smoothing parameter'.

The kernel K determines the shape of the bumps associated with each data point, while h will determine their width. As h is reduced more erronious fine structure becomes noticable and we are essentially computing the sum of a set of delta functions. Conversely, as h becomes large detail from the probability distribution becomes obscured.

Now $\hat{f}(x)$ inherits the properties of K, so if K is Gaussian then $\hat{f}(x)$ will be a smooth curve with derivatives of all orders.

A.1.2 Choice of smoothing parameter

A number of methods exist for selecting the smoothing parameter based on an optimization process. Here we describe the method of least squares cross validation, used throughout the course of this work to select the smoothing parameter.

The integrated square error is defined

$$\int (\hat{f}(x) - f)^2 = \int \hat{f}(x)^2 - 2\int \hat{f}(x)f + \int f^2 \tag{A.7}$$

The last term does not depend on $\hat{f}(x)$, so we need to minimize

$$R(\hat{f}(x)) = \int \hat{f}(x)^2 - 2\int \hat{f}(x)f \tag{A.8}$$

Define $\hat{f}(x)_{-i}$ to be the density estimate constructed from every point except X_i:

$$\hat{f}(x)_{-i} = \frac{1}{h(n-1)} \sum_{i \neq j} K(\frac{(x - X_j)}{h}) \tag{A.9}$$

In turn we can define a function $M_0(h)$ through:

$$M_0(h) = \int \hat{f}(x)^2 - \frac{2}{n}\sum_i \hat{f}(X_i)_{-i} \tag{A.10}$$

Taking the expection of the second density estimate we see

$$\begin{aligned} E(\frac{1}{n}\sum_i \hat{f}(X_i)_{-i}) &= E(\hat{f}(X_n)_{-n}) \\ &= E(\int \hat{f}(x_n)_{-n} f(x)dx) \\ &= E(\int \hat{f}(x) f(x)dx) \end{aligned} \tag{A.11}$$

Hence $E(R(\hat{f})) = E(M_0(h))$, which reduces the problem of finding the smoothing parameter to that of an optimization problem of M_0 over h. It can be shown [22] that the mean integrated square error can be best reduced using the Epanechnikov kernel,

$$K_e(t) = \begin{cases} \frac{3}{4}(1 - t^2) & -1 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{A.12}$$

During this work the Epanechnikov kernel was used for all density estimation.

Appendix B

The χ-distribution

B.1 χ-distribution; sample derivation

Here one derivation of the χ-distribution is presented. In principal it is comprised of two components, a derivation of the χ-squared density function, and a change of variables to transform this into the χ-distribution.

B.1.1 The χ-squared density

Let X be a random variable with Gaussian probability density function $f(x)$, where

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{B.1}$$

and σ, μ, are the standard deviation and the mean respectively.

By definition the moment generating function is then

$$\begin{aligned} \Psi_{X^2}(t) &= E(e^{tx^2}) \\ &= \frac{1}{\sigma\sqrt{2\pi}} \int e^{tx^2 - \frac{(x-\mu)^2}{2\sigma^2}} dx \end{aligned} \tag{B.2}$$

Just completing the square and integrating we obtain

$$\Psi_{X^2}(t) = \frac{1}{\sqrt{1-2\sigma^2 t}} e^{\frac{-\mu^2 t}{1-2\sigma^2 t}} \tag{B.3}$$

Now by ensuring X^2 has mean $\mu = 0$ we obtain

$$\Psi_{X^2}(t) = \frac{1}{\sqrt{1-2\sigma^2 t}} \tag{B.4}$$

Define random variable $\chi^2 = X_1^2 + ... + X_k^2$, where the X_i are i.i.d Gaussian distributed with mean zero. Since the X_i are independent:

$$\Psi_{X_i+X_j}(t) = \Psi_{X_i}(t)\Psi_{X_j}(t) \tag{B.5}$$

Therefore χ^2 has moment generating function

$$\Psi_{X^2}(t) = \frac{1}{(1-2\sigma^2 t)^{\frac{k}{2}}} \tag{B.6}$$

It should be noted that a Γ distribution with free parameters α and λ has moment generating function

$$\Psi_X(t) = \frac{1}{(1-\frac{t}{\lambda})^{\alpha}} \tag{B.7}$$

and corresponding density function

$$f(x;\alpha,\lambda) = \frac{\lambda^{\alpha} x^{\alpha-1} e^{-\lambda x}}{\Gamma(\alpha)} \tag{B.8}$$

We can observe that the χ^2 moment generating function is just a special case of that of the Γ moment generating function, with $\lambda = \frac{1}{2\sigma^2}$ and $\alpha = \frac{k}{2}$ The probability density function of the χ^2 is therefore

$$f_{\chi^2(k,\sigma)}(x) = \left(\frac{1}{2\sigma^2}\right)^{\frac{k}{2}} \frac{x^{\frac{k}{2}-1} e^{-\frac{x}{2\sigma^2}}}{\Gamma(\frac{k}{2})} \tag{B.9}$$

B.1.2 χ-square change of variables

The χ distributed random variable, defined $\chi = \sqrt{X_1^2 + ... + X_k^2}$, is obtained from the χ-square distribution through using the transformation

$$\tilde{p}(y) = p(g^{-1}(y)) \left| \frac{dx}{dy} \right| \tag{B.10}$$

where $g : \chi^2 \mapsto \chi$, that is, $g(x) = \sqrt{x}$.

Applying the transformation yields the density function of the non-standard χ-distribution:

$$f_{\chi}(x; k, \sigma) = \frac{2^{1-\frac{k}{2}} x^{k-1} e^{\frac{x^2}{2\sigma^2}}}{\sigma^k \Gamma(\frac{k}{2})} \tag{B.11}$$

with $x > 0$, $\mu = 0$.

Appendix C

Gaussian channel capacity theory

C.1 The capacity of a Gaussian channel

An information channel is a channel with some input given by a random variable X, output given by Y, and noise given by Z. In the case of a channel with additive Gaussian noise this relationship is denoted:

$$Y = X + Z \tag{C.1}$$

The information capacity of a Gaussian channel is defined as the maximum mutual information between input random variable X and output random variable Y, over all input distributions $f(x)$ satisifying constraint P_{av}:

$$E(X^2) \leq P_{av} \tag{C.2}$$

The information capacity is then

$$C = \max_{f(x):E(X^2)\leq P_{av}} I(X;Y) \tag{C.3}$$

We can bound the mutual information:

$$\begin{aligned} I(X;Y) &= h(Y) - h(Y \mid X) \\ &= h(Y) - h(X + Z \mid X) \\ &= h(Y) - h(Z \mid X) \\ &= h(Y) - h(Z) \end{aligned} \tag{C.4}$$

Defining Gaussian noise Z as $Z \sim \phi$ where

$$\phi(z) = \frac{1}{\sqrt{2\pi N}} e^{\frac{-z^2}{2N}} \tag{C.5}$$

and $E(Z^2) - E(Z)^2 = E(Z^2) = N$ we have

$$\begin{aligned} h(Z) &= h(\phi) \\ &= \frac{-1}{\log_2 e} \int \phi(x) \ln \phi(x) dx \\ &= \frac{-1}{\log_2 e} \int \phi(x) \ln \left(\frac{1}{\sqrt{2\pi N}} e^{\frac{-x^2}{2N}} \right) dx \\ &= \frac{1}{2 \log_2 e} \ln(2\pi N) + \frac{1}{2N \log_2 e} \int x^2 e^{\frac{-x^2}{2N}} dx \\ &= \frac{1}{2 \log_2 e} \ln(2\pi N) + \frac{1}{2 \log_2 e} \\ &= \frac{1}{2 \log_2 e} \ln(2\pi N e) \end{aligned} \tag{C.6}$$

Simplifying slightly we obtain

$$h(Z) = \frac{1}{2} \log_2(2\pi e N) \text{ bits} \tag{C.7}$$

Therefore

$$I(X;Y) = h(Y) - \frac{1}{2} \log_2(2\pi e N) \tag{C.8}$$

Now a bound on the entropy $h(Y)$ is introduced:

$$\begin{aligned} E(Y^2) &= E((X+Z)^2) \\ &= E(X^2) + 2E(X)E(Z) + E(Z^2) \\ &= E(X^2) + E(Z^2) \end{aligned} \tag{C.9}$$

since $E(Z) = 0$. As $E(X^2) \leq P_{av}$ and $E(Z^2) - E(Z)^2 = E(Z^2) = N$ we have

$$E(Y^2) \leq P_{av} + N \tag{C.10}$$

A fundamental result of information theory is that a Gaussian distribution maximizes $h(Y)$, where $h(Y) = \frac{1}{2}\ln(2\pi e\sigma^2)$ and σ^2 denotes the variance of Y. We already have a bound on the variance of Y, which yields

$$h(Y) \leq \frac{1}{2}\log_2(2\pi e(P_{av} + N)) \tag{C.11}$$

Combining the results for $h(Y)$ and $h(Z)$ we obtain

$$\begin{aligned} I(X;Y) &\leq \frac{1}{2}\log_2(2\pi e(P_{av}+N)) - \frac{1}{2}\log_2(2\pi eN) \\ &\leq \frac{1}{2}\log_2\left(1 + \frac{P_{av}}{N}\right) \end{aligned} \tag{C.12}$$

Since $max_{f(x):E(X)^2 \leq P_{av}} I(X;Y)$ we must have

$$C = \frac{1}{2}\log_2\left(1 + \frac{P_{av}}{N}\right) \tag{C.13}$$

giving the capacity of a channel with Gaussian additive noise.

Appendix D

Discrete and continuous entropy

D.1 Entropy through measure theory

Probabilities are probability measures with total measure of one. When entropy is phrased in the language of measure theory the distinction between discrete and differential entropy will be seen to be governed by the choice of an appropriate measure. Examination of this approach leads to an intuitive understanding of the non-absoluteness of entropy in both the discrete and continuous context.

D.1.1 Entropy: a measure theory approach

The general form of entropy is given

$$h(X) = -\int_X f(x) \log f(x) d\mu(x) \tag{D.1}$$

where μ is a Lebesgue measure. For example the Lebesgue measure $\mu(A)$ on the interval $A = [a, b]$ is $b - a$, the interval width.

If f is a simple function, then

$$\int f d\mu = \sum_{i=1}^{n} a_i \mu(A_i) \tag{D.2}$$

Now if E is a measurable set:

$$\int_E f d\mu = \sum_{i=1}^{n} a_i \mu(A_i \cap E) \tag{D.3}$$

D.1.2 Discrete entropy

The general form of entropy reduces to the discrete form through use of a specific measure. If ν denotes the counting measure, that is, when applied to a set the number of elements in that set, and if X is discrete, then

$$\begin{aligned} H(X) &= -\int_X f(x) \log f(x) d\nu(x) \\ &= -\sum_{x \in X} f(x) \log f(x) \end{aligned} \tag{D.4}$$

where $d\nu(x) = 1$ for $x \in X$.

D.1.3 The Radon-Nikodym theorem

Say ν is a σ-finite measure on a measurable space (X, Σ), and ν is absolutely continuous with respect to μ, where μ is another σ-finite measure on the same (X, Σ), then there exists f, a measurable function on X, which satisfies

$$\nu(A) = \int_A f d\mu \tag{D.5}$$

Now f is commonly written $d\nu/d\mu$, the *Radon-Nikodym derivative*, and is seen as a probability density function. Note that μ is sometimes known as the *reference measure*.

D.1.4 Entropy in terms of the Radon-Nikodym derivative

Entropy has already been stated as

$$h(X) = -\int_X f(x) \log f(x) d\mu(x) \tag{D.6}$$

or equivalently

$$h(X) = -E(\log f(x)) \tag{D.7}$$

Here $f(x)$ can be written as

$$f(x) = dP/d\mu \tag{D.8}$$

where P is some probability measure. Importantly we see that the entropy depends on two measures, the probability measure P and the reference measure μ. That is:

$$h(X) = -E(\log dP/d\mu) \tag{D.9}$$

We see that measure theory allows for a more general stating of entropy. The measurable function f is a probability density where the 'non-absoluteness'of entropy can be seen: entropy depends crutially on both the choice of probability measure and reference measure.

Bibliography

[1] E.D. Adrian, Y. Zotterman. The impulses produced by sensory nerve endings: Part II: The response of a single end organ. *Journal of Physiology*, **61**:151–171, 1926.

[2] F.E. Theunissen, J.P. Miller. Temporal encoding in nervous systems: A rigorous definition. *Journal of Computational Neuroscience*, **2**:149–162, 1995.

[3] P. Dayan, L.F. Abbott. Theoretical Neuroscience: computational and mathematical modeling of neural systems. *Cambridge, Massachusetts: The MIT Press*, 2001.

[4] W. Bialek, F. Rieke, R.R. de Ruyter van Steveninck, D. Warland. Reading a neural code. *Science*, **252**:1854–1857, 1991.

[5] S. Thorpe, A. Delorme, R. van Rullen. Spike-based strategies for rapid processing. *Neural Networks*, **14**:715–725, 2001.

[6] A.P. Georgopoulos, A.B. Schwartz, R.E. Kettner. Neuronal population coding of movement direction. *Science*, **233**:1416–1419, 1986.

[7] D.M. MacKay, W.S. McCulloch. The limiting information capacity of a neuronal link. *The Bulletin of Mathematical Biophysics*, **14**:127–135, 1952.

[8] A.K. Warzecha, J. Kretzberg,M. Egelhaaf Temporal precision of the encoding of motion information by visual interneurons. *Current Biology*, **8**:359–368, 1998.

[9] T.M. Cover, J.A. Thomas. Elements of information theory. *Wiley, London*,1991.

[10] A.J. Dubbs, B.A. Seiler, M.O. Magnasco. A fast $\mathcal{L}_p$ spike alignment metric. *Neural Computation*, **22**:2785–2808, 2010.

[11] D.H. Johnson. Dialogue concerning neural coding and information theory. (`http://www.ece.rice.edu/~dhj/dialog.pdf`).

[12] I. Rubin. Information rates and data-compression schemes for Poisson processes. *IEEE Transactions on Information Theory*, **22**:200–210, 1974.

[13] I. Rubin. Rate distortion functions for non-homogeneous Poisson processes. *IEEE Transactions on Information Theory*, **20**:669–672, 1974.

[14] M.A. Stephens. EDF Statistics for goodness of fit and some comparisons *Journal of the American Statistical Association*, **69**:730–737, 1974.

[15] C. Houghton. Studying spike trains using a van Rossum metric with a synapses-like filter. *Journal of Computational Neuroscience*, **26**:149–155, 2009.

[16] C. Houghton. A comment on 'a fast l_p spike alignment metric' by A. J. Dubbs, B. A. Seiler and M. O. Magnasco [arxiv:0907.3137]. (2009, http://arxiv.org/abs/0908.1260).

[17] C. Houghton, J.D. Victor. Measuring representational distances - the spike-train metrics approach. (Visual Population Codes: Toward a Common Multivariate Framework for Cell Recording and Functional Imaging, MIT Press, Cambridge, Massachusetts), 2011.

[18] D. Aronov, J.D. Victor. Non-Euclidean properties of spike train metric spaces. *Physical Review E*, **69**:061905, 2004.

[19] R. Narayan, G. Graña, K. Sen. Distinct time scales in cortical discrimination of natural sounds in songbirds. *Journal of Neurophysiology*, **96**:252–258, 2006.

[20] F. Rieke, D. Warland, R.R. de Ruyter van Steveninck, W. Bialek. *Spikes: exploring the neural code. MIT Computational Neuroscience Series,* 1999.

[21] C.E. Shannon. A mathematical theory of communication. *Bell Systems Technical Journal*, **27**:379–423,623–656, 1948.

[22] B.W. Silverman. Density estimation *Chapman and Hall, London* , 1986.

[23] J.L. Hodges, E.L. Lehmann. The efficiency of some nonparametric competitors of the *t*-test. *Annals of Mathematical Statistics*, **272**:324–335, 1956.

[24] R.R. de Ruyter van Steveninck, G.D. Lewen, S.P. Strong, R. Koberle, W. Bialek. Reproducibility and variability in neural spike trains. *Science*, **275**:1805–1808, 1997.

[25] M. van Rossum. A novel spike distance. *Neural Computation*, **13**:751–763, 2001.

[26] J.C. Principe, J. Xu, R. Jenssen, A.R.C. Paiva, I. Park Information theoretic learning: Renyi's entropy and kernel perspectives. *Springer*, 2010.

[27] J.D. Victor, K.P. Purpura. Nature and precision of temporal coding in visual cortex: a metric-space analysis. *Journal of Neurophysiology*, **76**:1310–1326, 1996.

[28] J.D. Victor, K.P. Purpura. Metric-space analysis of spike trains: theory, algorithms, and application. *Network*, **8**:127–164, 1997.

[29] S. Schreiber, J.M. Fellous, J.H. Whitmer, P.H.E. Tiesinga, T.J. Sejnowski. A new correlation-based measure of spike timing reliability. *Neurocomputing*, **52**:925-931, 2003.

[30] J.D. Victor. Binless strategies for estimation of information from neural data. *Physical Review E*, **66**:051903, 2002.

[31] J.D. Victor. Spike train metrics. *Current Opinion in Neurobiology*, **15**:585–592, 2005.

[32] L. Wang, R. Narayan, G.M. Graña, M. Shamir, K. Sen. Cortical discrimination of complex natural stimuli: can single neurons match behavior? *Journal of Neuroscience*, **27**:582–589, 2007.

[33] H. Shimazaki, S. Shinomoto. A method for selecting the bin size of a time histogram. *Neural Computation*, **19**:1503–152, 2007.

[34] W. Wu, A. Srivastava. Towards statistical summaries of spike train data. *Journal of Neuroscience Methods*, **195**:107–110, 2011.

[35] C. Houghton, K. Sen. A new multi-neuron spike-train metric. *Neural Computation*, **20**:1495–1511, 2008.

[36] S. Amari, H. Nagaoka. Methods of information geometry. *American Mathematical Society* , 2007.

[37] T.F. Cox, M.A. Cox. Multidimensional scaling. *Chapman and Hall, London*, 2001.

[38] J.B. Tenenbaum, V. de Silva, J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, **290**:2319–2323, 2000.

[39] M. Balasubramanian, E.L. Schwartz. The isomap algorithm and topological stability. *Science*, **295**:7, 2002.

[40] Frederic Theunissen Laboratory, University of Berkeley. Single-unit recordings from multiple auditory areas in male zebra finches `http://crcns.org/data-sets/aa/aa-2` , 2010.

[41] N. Amin, P. Gill, F.E. Theunissen. Role of the zebra finch auditory thalamus in generating complex representations for natural sounds. *Journal of Neurophysiology* **104**:784–798, 2010.

[42] D.A. Anderson, T.W. Darling. Asymptotic theory of certain 'goodness-of-fit' criteria based on stochastic processes. *Annals of Mathematical Statistics* **23**:193–212, 1952.

[43] G.E.P. Box, M.E. Muller. A note on the generation of random normal deviates. *Annals of Mathematical Statistics*, **29**:610–611, 1958.

[44] P. Gill, J. Zhang, S.M. Woolley, T. Fremouw, F.E. Theunissen Sound representation methods for spectro-temporal receptive field estimation. *Journal of Computational Neuroscience*, **21**:5–20, 2006.

[45] J.B. Gillespie, C. Houghton. A metric space approach to the information capacity of spike trains. *Journal of Computational Neuroscience*, **30**:201–209, 2011.

[46] W.H. Press, S.A. Teukolsky, W.T. Vetterling, P.B. Flannery. Numerical recipes in C++ 3rd Edition: the art of scientific computing. *Cambridge University Press*, 2007.

[47] P.H. Sellers. On the theory and computation of evolutionary distances. *SIAM Journal on Applied Mathematics*, **26**:787–793, 1974.

[48] J.D. Victor, D. Goldberg, D. Gardner. Dynamic programming algorithms for comparing multineuronal spike trains via cost-based metrics and alignments. *Journal of Neuroscience Methods*, **161**:351–360, 2007.

[49] S.M.N. Woolley, T.E. Fremouw, A. Hsu, F.E. Theunissen. Tuning for spectro-temporal modulations as a mechanism for auditory discrimination of natural sounds. *Nature Neuroscience*, **8**:1371–1379, 2005.

[50] S.M.N. Woolley, P.R. Gill, T.E. Fremouw, F.E. Theunissen. Functional groups in the avian auditory system. *Journal of Neuroscience*, **29**:2780–2793, 2009.

[51] S.M.N. Woolley, P.R. Gill, F.E. Theunissen. Stimulus-dependent auditory tuning results in synchronous population coding of vocalizations in the songbird midbrain. *Journal of Neuroscience*, **26**:2499–2512, 2006.

[52] N. Amin, P.R. Gill, F.E. Theunissen. Role of the zebra finch auditory thalamus in generating complex representations for natural sounds. *Journal of Neurophysiology*, **104**:784–798, 2010.