Measuring representational distances – the spike-train metrics approach

Conor Houghton[1,*]and Jonathan Victor[2,†]

January 1, 2010

[1]School of Mathematics, Trinity College Dublin, Dublin 2, Ireland.

[2]Department of Neurology & Neuroscience, Weill Cornell Medical College,
1300 York Avenue, New York, New York 10021, USA

Abstract

A fundamental problem in studying population codes is how to compare population activity patterns. Population activity patterns are not just spatial, but spatiotemporal. Thus, a principled approach to the problem of the comparison of population activity patterns begins with the comparison of the temporal activity patterns of a single neuron, and then, to the extension of the scope of this comparison to populations spread across space.

Since 1926 when Adrian and Zotterman reported that the firing rates of somatosensory receptor cells depend on stimulus strength, it has become apparent that a significant amount of the information propagating through the sensory pathways is encoded in neuronal firing rates. However, while it is easy to define the average firing rate for a cell over the lengthy presentation of a time-invariant stimulus, it is more difficult to quantify the temporal features of spike trains. With an experimental data set extracting a time-dependent rate function is model dependent since calculating it requires a choice of a binning or smoothing procedure.

[*] houghton@maths.tcd.ie
[†] jdvicto@med.cornell.edu

The spike train metric approach is a framework that distills and addresses these problems. One family of metrics are “edit distances” that quantify the changes required to match one spike train to another; another family of metrics first maps spike trains into vector spaces of functions. Both these metrics appear successful in that the distances calculated between spike trains capture the differences between the stimuli that elicit them. Studying the properties of these metrics illuminates the temporal coding properties of spike trains.

The approach can be extended to multineuronal activity patterns, with the anticipation that it will prove similarly useful in understanding aspects of population coding. The multineuronal metric approach forms a conceptual bridge between metrics applicable to time series and metrics applicable to spatial activity patterns.

Finally, since the metrics presented here are unlikely to exhaust the possible ways to usefully quantify distances between spike trains, the chapter concludes with some comments on how the properties of neuronal computation could be used to derive other spike train metrics.

Index Words: metric, multineuronal metric, Victor-Purpura metric, van Rossum metric, transmitted information, edit length, earth mover distance, spike train, spike code, spike rate, population code, synapse, zebra finch, linear filter, kernel.

1 Introduction

Spike trains are the means by which information propagates through the brain. Despite the apparently variable nature of an individual neuron’s firing patterns, our mental abilities – including our ability to rapidly and reliably perceive external stimuli – rely on information being communicated and processed as spike trains. Nonetheless, it is not clear how spike trains can best be described from a mathematical perspective; but any description should support analysis of the coding properties of individual neurons and of neural populations, in terms of both average behavior and variability.

Clearly the firing rate is one important feature of a spike train. In the visual system for example, examining how the firing rate is modulated by a stimulus has led to considerable progress in understanding how visual information is processed. However, even though the firing rate seems a straightforward concept, it becomes less straightforward when it is used as a measure of behavior on fine temporal scales: a spike train is composed of individual spikes and the precise pattern of spike times is typically highly variable across multiple presentations of the same stimulus. The obvious way to calculate a time-varying spike rate requires a binned or smoothed spike train, often averaged across trials. However, even this simple procedure is model-dependent and it is difficult to determine whether a protocol for calculating the rate preserves the relevant temporal patterning of the spike times.

The metric approach is an attempt to construct a principled mathematical framework for analyzing spike train data. The basic idea is to define a measure of the distance between spike trains. For the analysis of multineuronal responses, this strategy can be extended to give a notion of distances between sets of spike trains, or, equivalently, composite spike trains in which the spikes are labeled as to neuron of origin.

A useful analogy is provided by distances in a city; the most immediate measure of the distance between two points is the "as the crow flies" distance, the actual physical displacement of the two locations. However for many applications a more relevant measure of displacement is provided by the *route distance*; that is, the distance along the ground. The route distance takes into account the built geography of the city. Not only is the route distance more useful for calculating journey times, it is also descriptive of the nature of urban geography. The relationship between the route distance and the "as the crow flies" distance gives a local quantification of the urban grain: the way steets are arranged into a street system. Similarly, in the case of the brain, finding a geometric framework for describing spike trains in terms of a physiologically motivated distance, rather than a generic one, should allow for a more meaningful description of the relationship between stimuli and responses in local neuronal circuits.

These considerations lead us to start with a very general mathematical description of the space of spike trains as a space in which the distance between points can be calculated, namely, a metric space. By taking a general mathematical structure as a starting point we avoid fitting spike trains to the Procrustean bed of a particular mathematical framework that may have proved successful for other data types but may not be suitable for spike trains. After a brief introduction to metric spaces, two types of spike train metric are described below: the first type – kernel-based metrics, typified by the van-Rossum metrics – work by first mapping spike trains into the vector space of functions; the other metrics – typified by the Victor-Purpura metric – are *edit distance* metrics, somewhat similar to the Earth Mover Distance used in image analysis or the edit distance used for genome comparison.

In broad terms our goal is to find geometries for the space of spike trains which are relevant to the manner in which information is coded and processed in the brain. This, of course, requires a method for evaluating candidate metrics against this goal. One such method is described below. It evaluates a metric by computing how well the distance based clustering of spike trains matches the "true clustering" which groups the spike trains according to which stimulus caused them. A related, but distinct, question asks about the genesis of the metric structure of the space of spike trains. For this there are at least three approaches that can be taken. The first is a statistical approach: to think of similarity as a measure of how likely two spike trains are to be the result of the same stimulus. The second is based on effect, how similar is the effect of two spike trains on other neurons. The third is based on principle; asking what metrics have the properties that a spike metric might be required to have. None of the metrics discussed here fall precisely into any of these three viewpoints. However, we think that it is worthwhile to mention these considerations because ultimately, a satisfying theory will need to provide this understanding.

Of course, along with the particular properties of spike trains, the most striking feature of brain activity is the size and complexity of the neuronal network; understanding the coding and processing of information requires the analysis of multineuronal data. Each of the two example metrics described here can be extended to a metric on the space of multineuronal spike trains in a natural way: the metrics which are based on an embedding into a vector space, are extended by extending the vector space; the edit-distance metrics are extended by adding an extra type of "edit".

Below we present the ideas behind the metric space approach, and some examples of their application to neural data. Our examples are drawn from sensory systems, because in sensory systems, the structure of stimuli can be manipulated directly, and there are behavioral measures of "similarity" to complement the analysis of neuronal activity. For a brief overview of the metrics discussed, see Table 1.

We recognize that our approach leaves unanswered numerous mathematical questions related to the above considerations, such as a first-principles taxonomy of biologically reasonable metrics, and whether a framework even more general than metric spaces is appropriate. We defer a discussion of these and other open issues to the Conclusions.

Metric	Characteristics	Applicability
van Rossum	kernel-based, linear, Euclidean	single spike trains
multineuronal van Rossum	kernel-based, linear, Euclidean	multineuronal spike trains
synapse	kernel-based, non-linear, Euclidean	single and multineuronal spike trains
Victor-Purpura	edit distance, non-Euclidean	single and multineuronal spike trains
earth mover distance	edit distance, non-Euclidean	spatial patterns

Table 1: An overview of the metrics discussed.

2 Metrics

The definition of a metric abstracts the intuitive notion of a distance. Formally, a metric is a map from pairs of points in a space, a and b, to a real number $d(a,b)$ such that $d(a,b)$ is positive

$$d(a,b) \geq 0, \tag{1}$$

with equality if and only if $a = b$:

$$d(a,b) = 0 \Longleftrightarrow a = b, \tag{2}$$

symmetric

$$d(a,b) = d(b,a), \tag{3}$$

and satisfies the triangle inequality

$$d(a,b) \leq d(a,c) + d(c,b). \tag{4}$$

The meaning of the triangle inequality is that the distance between two points cannot be made shorter by going by way of a third point.

An instructive and familiar example of a metric is three-dimensional real space, $\mathbf{R}^3$. This is also a vector space, and, as we describe below, the vector space structure implies a metric structure as well. As with any other vector space, the vectors that describe the points in $\mathbf{R}^3$ can be added and multiplied by scalars to give other points in the space. Like most familiar vector spaces, it is also an inner-product space; there is a *dot product* between any two vectors $\mathbf{x} = (x_1, x_2, x_3)$ and $\mathbf{y} = (y_1, y_2, y_3)$ given by

$$\mathbf{x} \cdot \mathbf{y} = x_1 y_1 + x_2 y_2 + x_3 y_3 \tag{5}$$

To construct a metric, we first observe that the dot-product of a vector with itself is *a norm*, a measure of the length of the vector

$$|\mathbf{x}| = \sqrt{\mathbf{x} \cdot \mathbf{x}}. \tag{6}$$

This norm induces a metric, called the l^2*-metric*:

$$d(\mathbf{x}, \mathbf{y}) = |\mathbf{x} - \mathbf{y}| = \sqrt{\sum_{i=1}^{3} (x_i - y_i)^2}. \tag{7}$$

This metric is the familiar real-world notion of distance. The process of inducing a metric from a norm generalizes to any finite number of dimensions: these are *Euclidean spaces*. Provided some care is taken, this can be further generalized beyond finite-dimensional spaces to infinite-dimensional spaces, such as function spaces. For example, if $f(t)$ and $g(t)$ are both real functions on the same interval, $[0, T]$ say, then the L^2*-metric*, the function-space analog of the l^2-metric, is

$$d(f, g) = \sqrt{\int_0^T dt (f - g)^2}. \tag{8}$$

While, as we just saw, all Euclidean spaces are metric spaces, the converse is not true. In fact, Euclidean spaces are very special and have characteristic properties that are not shared by other geometries. One such property is that in a Euclidean space, the ratio of the circumference of a circle to its radius is 2π. An example of a non-Euclidean space is the surface of a sphere, along with a metric in which distances between points is the length of the great-circle arc that joins them. This metric is non-Euclidean; the ratio between the circumference and radius of a circle will depend on the radius and will always be less than 2π. This example is typical of non-Euclidean spaces in that the deviation from Euclidean geometry can be thought of as an intrinsic curvature of the space itself.

A more complex example of a non-Euclidean space is provided by the perceptual space of color vision. Human color perception is a three-parameter space, there are three different types of cone cells, each with a different spectral sensitivity. A color percept has coordinates corresponding to the intensity of activation of each cone type. In fact, this is an *affine space*, meaning that there is an additive structure: composing two colors gives a third color whose location in color space is the sum of the vectors describing the two original colors. In this way, color space can be embedded in $\mathbf{R}^3$ and this induces a Euclidean metric on it. So it is tempting to think of color space as a three-dimensional Euclidean space. However, this Euclidean structure does not correspond to perception. The perceptually relevant metric is the one that captures perceived color differences: the distance between two colors is determined by how different they are perceived to be by a human observer. Psychophysical measurements of this perceptual distance show that the metric cannot be Euclidean [27].

Like colors, spike trains cannot be assumed to have an intrinsically Euclidean geometry. In fact, they do not appear to form a vector space. While one might posit that "adding" spike trains corresponds to superposition, there is no natural way to subtract spike trains, and no first-principles reason to assume that adding the same quantity to two spike trains preserves the distance between them. It is even difficult to give a useful definition of the dimension of a spike train. Since a spike train is parametrized by its spike times, spike trains with different numbers of spikes would have to be considered to have different dimensions – even though it might make sense to regard spike trains with a similar but different number of spikes as being close to each other. In fact, for the two main types of spike train metrics discussed here, the edit-length metrics are typically non-Euclidean [5], while the metrics derived from embeddings into vector spaces are not.

2.1 Kernel-based metrics: spike train metrics derived from vector spaces

In this section, we describe a large and important family of spike train metrics exemplified by the van Rossum metrics. These metrics are defined by first mapping spike trains into a vector space, and then using the Euclidean distance in the vector space to calculate the distance between the original spike trains. To make the intuitions behind these metrics more evident, though, we begin with some special limiting cases before presenting the general definition. A schematic summary of the kernel-based metrics is given in

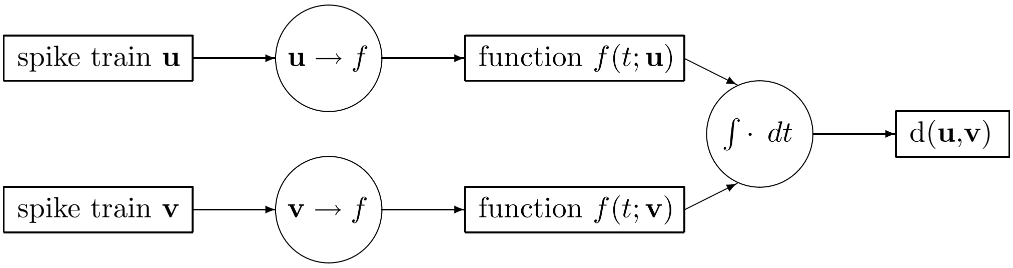
Fig. 1.

First noted in the somatosensory system, the influence of stimulus strength on a neuron's firing rate is perhaps the most broadly observed principle in the sensory systems. Somatosensory receptor cells fire with a rate that depends on the stimulus strength; V1 cells in the mammalian visual cortex fire with a rate that depends on how well the stimulus matches a receptive field and auditory cells are tuned to show a rate response to particular features in sound. In each of these examples, the degree to which some feature is present in a stimulus is represented by a neuron's firing rate. This immediately suggests a metric: if the information in a spike train is encoded in the number of spikes then the way to distinguish between spike trains is to compare the number of spikes. Thus, if $\mathbf{u} = \{u_1, u_2, \ldots, u_m\}$ and $\mathbf{v} = \{u_1, u_2, \ldots, u_n\}$ are two spike trains recorded during two trials, each of duration T, the spike count distance[1] is

$$d(\mathbf{u}, \mathbf{v}) = |m - n|. \tag{9}$$

Obviously we cannot expect that in general, the meaning of a spike train will be fully accounted for by the number of spikes it contains. Indeed, as will be seen below, the spike count distance performs poorly on the clustering task used to evaluate metrics. The immediate problem is clear: the spike train has temporal structure that is ignored by the spike count. We expect this to be the case when the corresponding stimulus changes rapidly with time. This happens, for example, in zebra-finch song, a stimulus with a rich structure at a scale measured in tens of milliseconds; data recorded from zebra finch will be used later in Sec. 3. However, it is also the case even if the stimulus is not changing during the recording – the nervous system adds its own dynamics to that of the stimulus [14]. To address this, the metric needs to be sensitive to

[1] The careful reader may have noticed that strictly speaking, the above definition does not constitute a metric. If two spike trains $\mathbf{u}$ and $\mathbf{v}$ have the same number of spikes, then $d(\mathbf{u}, \mathbf{v})$ will be zero, even though $\mathbf{u}$ and $\mathbf{v}$ differ, thus violating condition in eq. (2). Because of this, the above definition is properly considered a "pseudometric." However, this formal inconvenience is readily fixed by a slight change in the definition of the "points" in the metric space. Specifically, we consider all spike trains that have the same number of spikes to be represented by the same point. With this definition of the space the spike count distance of eq. (9) is a metric. In other words, we recognize that spike trains that are at a distance zero from each other, according to eq. (9), can be considered equivalent, and a pseudometric such as this one can always be considered as a metric on these equivalence classes. Moreover, if the "meaning" of a spike train is fully captured by this spike count distance, then all spike trains within the same equivalence class would, in fact, have equivalent meaning.



In the kernel-based metrics the spike train is mapped to a function of t:

$$\mathbf{u} \to f(t;\mathbf{u})$$

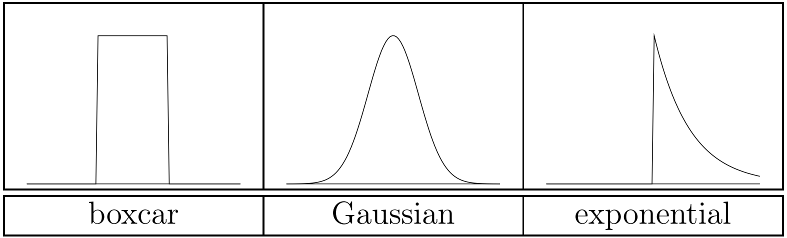
and the distance between two spikes $\mathbf{u}$ and $\mathbf{v}$ is the L^2 distance between the corresponding functions

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

In most versions of the metric the map from spike train to function is a convolution with a filter; if $\mathbf{u}$ has spike times $\{u_1, u_2, \dots, u_m\}$

$$f(t;\mathbf{u}) = \sum_{i=1}^{m} h(t - u_i)$$

where $h(t)$ is a kernel. Three examples are considered here:



For the synapse metric the map from spike train to function is more complicated and is intended to model the dynamics of a synapse:

$$\tau \frac{d}{dt} f = -f \quad \text{with} \quad f \to f + 1 - \mu f \quad \text{when a spike arrives.}$$

Figure 1: A schematic summary of the single neuron kernel-based metrics.

the temporal structure of the spike train, and not just the number of spikes.

One approach is to subdivide the interval of the recording into a number, I, of sub-intervals of length $\delta_T = T/I$ and take the spike count distance in each sub-interval

$$d(\mathbf{u}, \mathbf{v}; T_i, T_{i+1}) = |m(T_i, T_{i+1}) - n(T_i, T_{i+1})| \tag{10}$$

where $m(T_i, T_{i+1})$ is the number of spikes in $\mathbf{u}$ in the sub-interval $(T_i, T_{i+1}]$; $n(T_i, T_{i+1})$ performs the same role for $\mathbf{v}$. The distance between the two spike trains could then be calculated as the Pythagorean sum of all these I sub-interval distances

$$d = \sqrt{\sum_{i=0}^{I-1} d^2(\mathbf{u}, \mathbf{v}; i\delta_T, (i+1)\delta_T)} \tag{11}$$

The Pythagorean sum is used to combine the contributions from each sub-interval because it seems intuitive to do so: these are distances and in ordinary real-space, geometric distances in orthogonal directions are added using the Pythagoras theorem. However, intuition from real-space geometry is not necessarily applicable to spike trains, and this assumption should only be accepted provisionally. Note that when the interval $[0, T]$ is trivially "divided" into a single sub-interval, this reduces to the spike count metric defined above.

An undesirable aspect of this calculation is that it discretizes the spike trains, that is, it bins the spikes into particular sub-intervals. [2] As with any discretization, it introduces an artificial granularity, with arbitary bin boundaries, to the data. Put another way, if two spike trains have just a single spike, then they will be considered the same if the spikes happen to fall into the same sub-interval ("bin"), and different if they do not. However, the bin boundary may well be consequent to an arbitrary start time for data collection. This suggests an alternative: to use a continuously moving window to "look in" at the spike count distance at a time t and then sum over all $t \in (0, T)$. Since t is a continuous variable, the sum is performed as an integral, yielding a windowed spike count metric

$$d = \sqrt{\int_0^T dt d^2(\mathbf{u}, \mathbf{v}; t - \delta T/2, t + \delta T/2)}. \tag{12}$$

[2] Note that this metric also is, formally, a pseudometric: if two spike trains have matching spike counts in each sub-interval, then they will have a distance of zero, and should be considered equivalent.

The above formulation emphasizes the relationship of the metric to spike rates and binning.

There is an alternative form that is both more convenient for calculation, and suggests a path to further generalization. We define a *linear filtering* operation as a mapping from a spike train $\mathbf{u} = \{u_1, u_2, \cdots, u_m\}$ to a real function, $f(t; \mathbf{u})$ using a kernel $h(t)$:

$$\mathbf{u} \mapsto f(t; \mathbf{u}) = \sum_{i=1}^{m} h(t - u_i). \tag{13}$$

Thus, filtering maps spike trains into the vector space of real functions. It therefore immediately induces a metric on spike trains, via the standard L^2 metric on functions:

$$d(\mathbf{u}, \mathbf{v}) = \sqrt{\int dt [f(t; \mathbf{u}) - f(t; \mathbf{v})]^2}. \tag{14}$$

To bring this into approximate correspondence[3] with the metric defined above, we specify that the kernel is the boxcar window

$$h(t) = \left\{ \begin{array}{ll} 1/\delta T & -\delta T/2 < t < \delta T/2 \\ 0 & \text{otherwise} \end{array} \right. . \tag{15}$$

The basic idea here is that spikes are discrete events while a rate function is continuous. Any formulation of the spike rate requires a filtering of the spikes, smearing the discrete spike times into something that can add to give a continuous rate function. This is the role of the kernel $h(t)$, which can be considered to represent the "impact" of a single spike across time. To emphasize a relationship with the rate count distance we have used a boxcar kernel above, but other choices suggest themselves – for example, the exponential kernel [23] which will be discussed in Subsect. 2.2 and the Gaussian kernel

$$h(t) = \frac{1}{\sqrt{2\pi}\sigma} e^{-t^2/2\sigma^2}, \tag{16}$$

which provides a spike-count distance though a Gaussian rather than a boxcar window. Here, σ provides a temporal scale; as δT did before before. In all

[3] The correspondence is only approximate since the boxcar smooths the spike train continuously and without regard to absolute time, while binning is a discrete operation that depends on the (usually arbitrary) convention of when t=0.

of these kernels, there is a free parameter – the width of the kernel – when the kernel is wide, the corresponding metrics merely compare overall firing rate. When the kernel is narrow, they are sensitive to variations in the rate at small time resolutions.

We note that the use of the L^2 metric in the function space is not essential; kernel-based metrics can also be defined by importing any other vector-space metric, such as the L^p metric:

$$d(\mathbf{u},\mathbf{v}) = \left(\int dt |f(t;\mathbf{u}) - f(t;\mathbf{v})|^p \right)^{1/p} . \tag{17}$$

The L^1 metric will be of interest below; it will be seen that the L^1 van Rossum metric with boxcar filter, approximates the Victor-Purpura metric when the spikes are not dense.

2.2 Synapse-like metrics

The Gaussian kernel mentioned above is motivated by statistical considerations: each spike is viewed as an exemplar of a noisy random process whose rate is to be reconstructed using the kernel. However, neurons do not "reconstruct" a firing rate from their inputs; rather, they are biophysical machines that process synaptic inputs. Abstracting this physiology serves to motivate other metrics which include and generalize the kernel-based van Rossum metrics.

The basic idea is that the distance between two spike trains should be related to the difference in their effect on other neurons. Caricaturing the effect of a single spike as a causal exponential yields the kernel that defines the original exponential van Rossum metric [20]:

$$h(t) = \left\{ \begin{array}{ll} 0 & t < 0 \\ \frac{1}{\tau} e^{-t/\tau} & t \geq 0 \end{array} \right. . \tag{18}$$

This choice is motivated by the physiology of neuronal signaling: it can be considered to mimic the dynamics of transmission across a synapse. We go into this in some detail, because these considerations also lead to physiologically-motivated extensions [7] of the van Rossum metric. The central idea is to identify the function $f(t;\mathbf{u})$ with the post-synaptic conductance.

The arrival of a spike at a synapse causes neurotransmitter to diffuse across the synaptic cleft and bind with receptors on the dendritic spine, opening ion channels. This changes the conductance of the post-synaptic membrane, resulting in a change in its membrane potential. Diffusion of transmitter across the synaptic cleft occurs within a millisecond or less. Since this is much faster than other aspects of synaptic dynamics or the membrane time constants, it is reasonable to model the change in the post-synaptic conductance, f, as instantaneous:

$$f \to f + \delta f \tag{19}$$

whenever there is a pre-synaptic spike. The increment, δf, is a constant which can be set to one by rescaling f. Once the transmitter is bound, we assume that it becomes unbound at a constant rate

$$\tau \frac{d}{dt} f = -f \tag{20}$$

where τ is the time constant that governs this release. It is easy to check that, apart from an irrelevant overall amplitude scale, this conductance is identical to the function calculated by filtering the spike train with the exponential kernel. In short, the filtering gives a map from spike trains to functions; this map can be rewritten as an embedding into function space, $\mathbf{u} \to f(t; \mathbf{u})$, where $f(t; \mathbf{u})$ satisfies the decay equation eq. (20), with discontinuities $f \to f+1$ at the spike times $\{u_1, u_2, \dots, u_m\}$. Thus, a simple caricature of synaptic dynamics provides a mapping of spike trains into functions, and this mapping is described by a linear filtering with an exponential kernel.

We can now use this setup to incorporate more complicated aspects of synaptic conductance dynamics ([7]). Many synapses have "depressing" characteristics, [1]: the net effect of an arriving impulse is diminished when the synapse has been recently active. We can model this behavior by allowing the size of the f increment, δf, to depend on the spiking history. A simple way to do this is by including an additional parameter μ, which quantifies the dependence on spike history, by changing the increment to

$$\delta f = 1 - \mu f \tag{21}$$

The parameter μ is in the range zero to one, if $\mu = 0$ the function increases by one at every spike, regardless of the spiking history: this is the original kernel filter map from spike train to function. However, for non-zero values

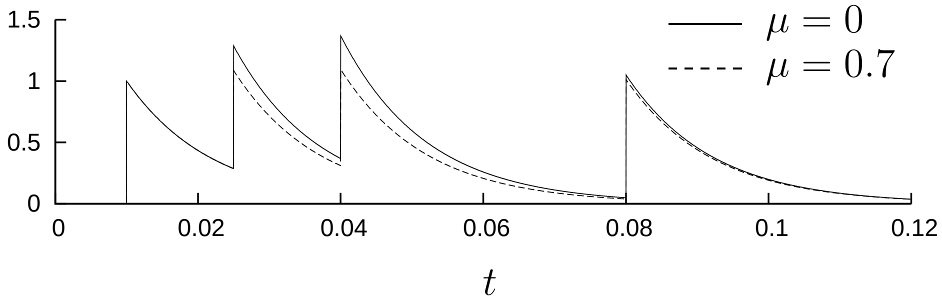


Figure 2: The function f with the synapse-like increment eq. 21calculated with $\mu = 0$ and $\mu = 0.7$ and $\tau = 12$ms. The function has been calculated for an example spike train with spikes at 10ms, 25ms, 40ms and 80ms; the $\mu = 0$ function is plotted with a solid line, the $\mu = 0.7$ with a dashed line. Since the function is zero up to $t = 10$ms both functions rise by the same amount at this first spike. However, the second spike follows soon after and so the two function rise by different amounts at $t = 25$ms; this is even more pronounced at $t = 40$ms. The spike at $t = 80$ms is sufficiently removed from the others so that the function f has nearly decayed to zero, and so the two functions have almost identical increases here.

of μ the amount the function increases depends on the spike history with a smaller increase when there has been a large number of previous spikes. This is illustrated in Fig. 2.

Biologically, this particular map is intended to model the effect of binding site depletion in the synaptic cleft. The increase in conductivity at the dendritic spine is the result of the opening of ligand-gated channels: these channels open when neurotransmitter in the synaptic cleft binds with receptor molecules. This means that the increase in conductivity depends on the number of available binding sites. This number, in turn, depends on how many binding sites are already occupied because of previous spikes; it is this depletion of unoccupied binding sites that is modeled by allowing δf to depend on history. If the number of binding sites is very large compared to the number which is typically occupied because of a singe spike, then the effect of binding-site depletion is small, corresponding to a value of μ near zero. On the other hand, if a spike causes all binding sites to be occupied, then the effect of a spike is to return the conductance to its maximum value: this corresponds to $\mu = 1$. We also mention that because spikes at different times interact, the mapping represented by eq. (21) is no longer a linear map from spike trains to functions, and cannot be expressed as a linear filter, eq. (13). Nevertheless, like the other kernel based metrics, the metric on the space of spike trains is induced from a Euclidean metric in the target space, the space of functions. The map could be represented as a sum of nonlinear kernels of increasing order, that is, as a Volterra expansion [25].

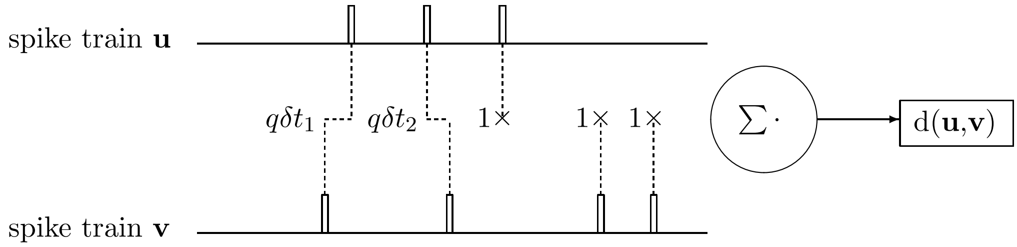
Qualitatively, the effect of incorporating binding site depletion is that some spikes have a larger effect than others on the distance between the spike train and other spike trains. In particular, if a spike follows closely after other spikes, when the binding sites are depleted, it will cause the function to increase by less than a spike that occurs in isolation, when the binding sites have recovered. This means, for example, that the precise timing and number of spikes in a burst is less significant than the timing and number of isolated spike. It will be seen in Sect. 3 that, for the zebra finch data used to compare the different metrics, this results in an improved ability to interpret spike trains.

This extension of the van Rossum metric is just one example of how neurophysiological principles can be used to motivate a mapping from spike times to a continuous function, and thereby define a spike metric. A similar strategy could be used to incorporate pre-synaptic effects, or post-synaptic effects other than binding site depletion.

These metrics – that result from a kernel-based mapping from spike trains into continuous functions, and then applying a metric in the function space – give only one of the two classes of metrics we wish to consider. The other class consists of "edit length" metrics, as typified by the Victor-Purpura metric described below. As we will see, these provide another framework for analyzing spike timing effects, but one that often leads to convergent conclusions.

2.3 Edit-length metrics

The fundamental point of departure between edit-length metrics and kernel-based metrics is that edit-length metrics calculate a measure of dissimilarity directly from a pair of spike trains, rather than going through an intermediate transformation of the individual spike trains into a function space. The broad consequence is that the resulting geometry of spike trains, when viewed through edit-length metrics, need not resemble that of a function space. In particular, edit-length distances can be expected to yield non-Euclidean geometries [23, 24]. Figure 3 gives a schematic summary for the Victor-Purpura metric.



The Victor-Purpura metric is an edit-distance metric; the distance between two spike trains is the total cost of changing one into the other using elementary steps where an individual cost has been assigned to each permitted elementary step:

- deleting and adding a spike costs one
- moving a spike a distance δt costs $q\delta t$.

Obviously there is no unique way of changing one spike train to the other using elementary steps; the metric distance is the minimum possible. There are other varients of the Victor-Purpura metric based on spike intervals rather than spike times, or with a larger collection of elementary moves.

Figure 3: A schematic summary of the single neuron Victor-Purpura metric.

The first component of an edit-length distance is a specification of a set of "elementary steps". Each step is a simple transformation of a spike train that can help mold it into another. The second component of an edit-length distance is an assignment of a non-negative cost to each step. Based on this assignment, any sequence γ of steps has, in turn, a cost $c(\gamma)$ calculated by adding the costs of its individual elementary steps.[4] The distance between two spike trains $\mathbf{u}$ and $\mathbf{v}$ is then defined as the minimum cost of transforming one spike train in to the other using a sequence of elementary steps:

$$d(\mathbf{u},\mathbf{v}) = \min_{\gamma} c(\gamma) \tag{22}$$

where the minimum is is taken over all sequences γ which transform $\mathbf{u}$ to $\mathbf{v}$. Provided that the cost of a step is the same as the cost of inverting the step, the above quantity is also guaranteed to be symmetric. The general problem of finding this minimum-cost sequence can be daunting, but, as we will see below, it can be done in an efficient manner for important special cases.

Typically, one kind of elementary step consists of inserting or deleting a spike, and this step is assigned a cost of one. This has two technical advantages. First, it sets a scale on the metric. Second, including this elementary step guarantees that the distance between two spike trains is finite, since at worst, one can remove all of the spikes from one train, and insert all of the spikes into the other. That is, given two spike trains with, respectively, m and n spikes, the distance between them is no greater than $m + n$. Adding other kinds of elementary steps reduces this distance and changes the characteristics of the metric.

As a simple example, we consider the "spike time" metric. In the spike time metric, we add a second elementary step, consisting of moving a spike. This move is assigned a cost of $q\delta t$, where δt is the distance moved. The behavior of the metric is determined by the parameter q, the cost per distance of moving a spike. Specifically, the parameter q gives a time scale $2/q$ which determines the relative sensitivity of the metric to spike count and spike time. The reason is that in any minimal-cost sequence of steps, no spike is moved

[4] In principle, it is possible to use other rules for calculating the total cost based on the cost of the constituent elementary steps, for example, an "L^2-like" edit-distance could be defined with

$$c(\gamma) = \sqrt{\sum [c(\gamma_i)]^2}$$

where the γ_i are the individual elementary steps making up γ.

by more than $2/q$: the cost to do so would exceed the cost of deleting the spike from one spike train and inserting it into the other. That is, from the point of view of the elementary steps, spikes in the two trains that have times that differ by more than $2/q$ are considered unrelated; it costs two units to bring them into correspondence, while spikes within a narrower window can be related at a lower cost, via a timing shift. In the limit of $q = 0$, the metric becomes the spike count distance, since there is no cost associated with aligning the spikes. As q increases, the metric becomes increasingly sensitive to spike time.

For this metric, an efficient algorithm for calculating the minimum-cost sequence, eq. (22), can be constructed based on the dynamic programming algorithm used for the comparison of genetic sequences [17]. This algorithm exploits properties of a minimum cost sequence. For example, in a minimal cost sequence, a spike is never moved after it is added because adding it into its final place would have the same effect at a lower cost. Similarly, a spike is never moved before it is deleted; the same result is accomplished by deleting it without first moving it. Also, a spike in one spike train is never moved past the position of a spike in the other – since the same result could be accomplished by keeping the spikes in their original order. As described in [23], these conditions imply that the last spike of one spike train, **u** say, can be treated in a way that considers only the last spike of **v**, the other spike train. This leaves just three possibilities: (a) the last spike in **u** is moved to match the last spike in **v**, (b) the last spike of **u** is deleted, or (c) a new last spike is is inserted in **u** to match the last spike in **v**. This gives an inductive method for calculating the distance. Note, however, that this relationship is best used in the forward direction via dynamic programming, gradually building up the distance between **u** and **v** from the distances between shorter sub-trains, rather than as a recursion.

This can be done in the following way. Let $G_{i,j}$ denote the distance between the truncated spike trains formed by the first i spikes of **u** and the first j spikes of **v**. Obviously $G_{i,0} = i$ and $G_{0,j} = j$; the other entries can be filled in one by one starting with $G_{1,1}$ using

$$G_{i,j} = \min\{G_{i-1,j-1} + q|u_i - v_j|, G_{i-1,j} + 1, G_{i,j-1} + 1\} \tag{23}$$

where the three option correspond to the three possibilities, (a), (b) and (c), described above. The distance is then the bottom right hand corner entry, $G_{m,n}$.

Another kind of edit-length metric can be defined in terms of elementary steps in which *intervals* are inserted, deleted, or stretched. This kind of metric is sensitive to intrinsic spike patterns, not to their absolute timing. The above algorithmic structure applies to these “interval” metrics in a straightforward manner [24]. Still other edit-length distances can be created by combining spike-time and spike-interval elementary steps, or by adding other kinds of elementary steps. One such kind of step consists of translating a subset of contiguous spikes by a fixed amount, capturing the idea that a “burst” plays a specific role in coding. One could also allow translation of a subset of *non*-contiguous spikes, capturing the idea that “motifs” of precisely timed spikes are important [2]. However, it is not obvious how to extend the above algorithm to calculate the minimum-cost paths for these metrics

One interesting aspect of the Victor-Purpura metric is that the cost function for the elementary step which moves a spike time can be changed to a different positive increasing function. As mentioned in [23], it is possible to change the Victor-Purpura metric by using some other, convex increasing positive, distance function: convexity is required by the triangle inequality. The dynamic programming algorithm also applies for other distance functions. Lots of other distance functions suggest themselves, for example, sub-threshold electrodynamics events in neurons are generally discounted in an exponential decay process suggesting a distance function which decays exponentially towards two. However, in a similar way to the van Rossum metric, the performance of the metric primarily depends on the time-scale of the distance function, not its detailed profile.

2.4 Comparing edit-length and kernel-based distances.

Edit-length distances and kernel-based distances have a different mathematical structure. Moreover, because edit-length distances have different kinds of steps, they distinguish between “jitter” and “unreliability” in describing the difference between two spike trains: some spikes are moved, others deleted or added. Kernel-based distances do not match up individual spikes in this way. Nevertheless, the Victor-Purpura and van Rossum metrics often lead to very similar results in terms of spike train classification. To understand the reason for this, [8], we replace the L^2 van Rossum metric by its L^1 analog,

$$d(f,g) = \int_0^T dt|f-g|. \tag{24}$$

An L^1 van Rossum metric whose kernel is the boxcar filter of eq. (15) with $\delta_T = 2/q$ yields precisely the same distance as the spike time distance when applied to pairs of single-spike spike trains, and to pairs of spike trains in which one spike train is empty. By extension, exact equality between the two metrics also holds when the spike trains being compared have widely separated spikes, because when spikes are widely separated, they do not interact. However, this correspondence is not exact: when spikes are close together, the metrics differ. For example, if one spike train has two spikes $\mathbf{u} = \{0, u\}$ where $u < 2/q$ and the other spike train has just one spike $\mathbf{v} = \{v\}$, where $0 \leq v \leq u$ then the distance between $\mathbf{u}$ and $\mathbf{v}$ measured using the L^1 van Rossum metric with boxcar filter is one, independent of the value of v, while the distance measured by the Victor-Purpura metric will be $1 + qv$ or $1 + q(u - v)$, whichever is less.

In a similar fashion, different Victor-Purpura distance functions can be chosen to match other kernel functions when applied to single-spike spike trains. For example, the L^1 van Rossum metric with exponential kernel will measure the same distance as a Victor-Purpura metric when the cost of moving a spike by an amount t is given by $2\left(1 - e^{-t/\tau}\right)$ and for the L^1 van Rossum metric with Gaussian filter, the corresponding cost for moving a spike is the error function $\operatorname{erf}(t/\sqrt{2}\sigma)$.

An intuitive approach to defining a spike train metric would be to define the distance as the number of spikes in each spike train that is not co-incident with a spike in the other spike train. This, in a sense, defines the spike distance as $|\mathbf{u} - \mathbf{v}|$ where the minus operation on spike trains is defined as “canceling” co-incident spikes. This approach is also appealing from a physiological point of view: because neurons have a firing threshold, they often can be regarded as coincidence detectors, one spike arriving on a pre-synaptic neuron will not cause a post-synaptic neuron to fire, but a sufficient number of spikes arriving within a sufficiently narrow time window will cause the post-synaptic neuron to fire. This is probably the most obvious way for spike times to play an important role in neuronal computation.

Of course, one difficulty is that two spikes are never precisely coincident; coincidence must be defined relative to some fixed scale. This suggests a “windowed coincidence detection” distance where the distance between two spike trains is given by the number of spikes that are not sufficiently close to being coincident. However, this does not give a metric; this distance does not satisfy the triangle inequality. As an example of a violation, say the precision timescale is δ_T, then the single-spike spike trains $\mathbf{u}_1 = \{0\}$ and $\mathbf{u}_2 = \{3\delta_T/4\}$

are a distance zero from each other, as are $\mathbf{u}_2$ and $\mathbf{u}_3 = \{3\delta_T/2\}$; however, the spikes in $\mathbf{u}_1$ and $\mathbf{u}_3$ are separated by more than δ_T and so the distance between these to spike trains is two. In a way the Victor-Purpura metric can be thought of as resolving this difficulty by adding a cost to moving a spike. The linear cost function is the least-convex convex function and so the Victor-Purpura metric is, in a sense, the metric that comes closest to coincidence detection.

3 Evaluating the functional relevance of metrics

A metric can be viewed as a formal hypothesis for how the brain represents information. Here, we present a machinery for testing this hypothesis. The basic idea is that if a metric expresses the meaningful similarities and dissimilarities of neural messages, then the distances between spike trains should correspond to the perceptual or behavioral distances between the entities that they represent.

This test is most straightforward to implement in a sensory system. The basic idea is to use the candidate metric to cluster a set of spike trains, and to compare this clustering with a "gold standard", namely, clustering the spike trains according to the stimuli that elicited them. To implement this method, one needs a set of spike trains recorded during repeated presentations of a set of stimuli. These, for example, could be spiking responses recorded from a visual area during repeated presentations with different types of visual stimuli, or, as will be used as an example here, spiking responses recorded from the primary auditory neurons of zebra finch during repeated playback of songs from a standard repertoire. The metric is scored by how well the metric clustering matches the stimulus-based clustering.

In one version of this approach, introduced in [23], clustering is compared via a *confusion matrix*, N. The confusion matrix is a square matrix whose ijth entry, N_{ij}, is the number of responses from stimulus i which is clustered by the metric into the cluster corresponding to stimulus j. Thus, the diagonal entries count the number of spike trains that are closest to the correct cluster (i.e., correctly decoded); the off-diagonal elements count the number of spike trains that are closer to a cluster corresponding to a different stimulus (i.e., incorrectly decoded).

In [23], the confusion matrix is calculated by a leave-out-one procedure, as follows. Given n responses and K stimuli, the responses are first grouped into K clusters $\{C_1, C_2, \ldots, C_K\}$ according to their stimulus. Starting with a $K \times K$ matrix N of zero entries, each response r is considered in turn. It is temporarily removed and the distance $d(r,s)$ between it and each spike train s is calculated. A weighted average distance between it and the spike trains in each stimulus-defined cluster is calculated: for each $k \in \{1, 2, \ldots, K\}$

$$d_k = \left[\frac{1}{|C_k|} \sum_{s \in C_k} d(r,s)^z \right]^{1/z}. \tag{25}$$

This weighted average (typically $z = -2$) is used to reduce the effect of outliers. This gives a set of average distances of the response r to each of the K clusters: $\{d_1, d_2, \ldots, d_K\}$. If the smallest distance in this set is d_j and the response being considered was taken from ith stimulus cluster, a tally is added to N_{ij}.[5] This tally indicates that the particular response, which actually was elicited by the ith stimulus, was seen by the metric as belonging to the jth stimulus-based cluster. When all the responses have been considered, the elements of N will add to give the total number of responses, n:

$$n = \sum_{i=1}^{K} \sum_{j=1}^{K} N_{ij}. \tag{26}$$

If the metric faithfully recovers the stimulus-based clustering, the confusion matrix will be largely diagonal. If it does not, the off-diagonal entries will be large. A natural nonparametric measure of where the metric lies on this continuum is the transmitted information of the confusion matrix, h:

$$h = \frac{1}{n} \sum_{ij} N_{ij} \left(\ln N_{ij} - \ln \sum_k N_{kj} - \ln \sum_k N_{ik} + \ln n \right). \tag{27}$$

[5] To complete the recipe for construction of the confusion matrix, one needs a convention to handle ties. One way to do this is that if several, b say, of the d_K are tied for a minimum, then $1/b$ is added to each N_{ij}, for which d_j is equal to the common minimum. Another, more flattering, convention adds one to the relevant diagonal entry of N if the distance to the spike trains own cluster is among those drawn at the minimum. Note that ties are much more likely to occur for a metric that has discrete values, such as the spike count metric.

For perfect clustering of K equally likely stimuli, the maximum value of the transmitted information h will be obtained: $h = \ln K$. For convenience, a normalized information, $\tilde{h} = h/\ln K$ will be used here.

For illustrative purposes a data set recorded from the primary auditory area of zebra finch will be considered. These electro-physiological data were previously analyzed in [13, 26, 7], and [13] should be consulted for a detailed description of the experimental and primary data processing procedures. The recordings were taken from field L of anesthetized adult male zebra finches and data were collected from sites which showed enhanced activity during song playback. 24 sites are considered, of these, six are classified as single-unit sites and the rest as consisting of between two and five units [13]. The average spike rate during song playback is 15.1 Hz with a range across sites of 10.5-33 Hz. At each site, ten responses to each of 20 zebra finch songs were recorded so, $K = 20$ and $n = 200$. In all the analysis here, one second of each recording is used, beginning at the onset of song playback. It should be noted that Field L is a primary auditory area, the metric properties of the spike trains recorded here are unlikely to be typical of areas higher in ascending pathway; let alone in other sensory modalities or species.

In Fig. 4 a comparison is given between the clustering performance of the spike count distance, the van Rossum metric with boxcar filter, Gaussian and exponential filter, the synapse-like metric and the Victor-Purpura metric. One striking aspect of this comparison is how similar to each other the performances of the three van Rossum metrics and the Victor-Purpura metric are. The profile of the filter seems to make little difference and – although the Victor Purpura metric differs from the van Rossum metrics in that it actually matches particular spikes in one spike train with spikes in the other – its performance on these data is very similar.

As illustrated in Fig. 5 for the van Rossum metric with boxcar filter, the performance depends significantly on the timescale of the filter; δ_T, the size of the window, in this case. For very small windows the performance is poor: it improves rapidly to the best average performance at $\delta_T = 28$ms. The decline in performance is very slow. As mentioned above, the value of $\tilde{h}$ for the spike count distance depends on a tie-breaking convention. However, as illustrated in Fig. 4, even for the most favorable convention, the spike count metric only achieves modest values of $\tilde{h}$.[6]

[6] In the definition of the van Rossum metric there is a slight awkwardness at the edges: in the boxcar case, for example, the window extends beyond the experimental period $[0, T]$

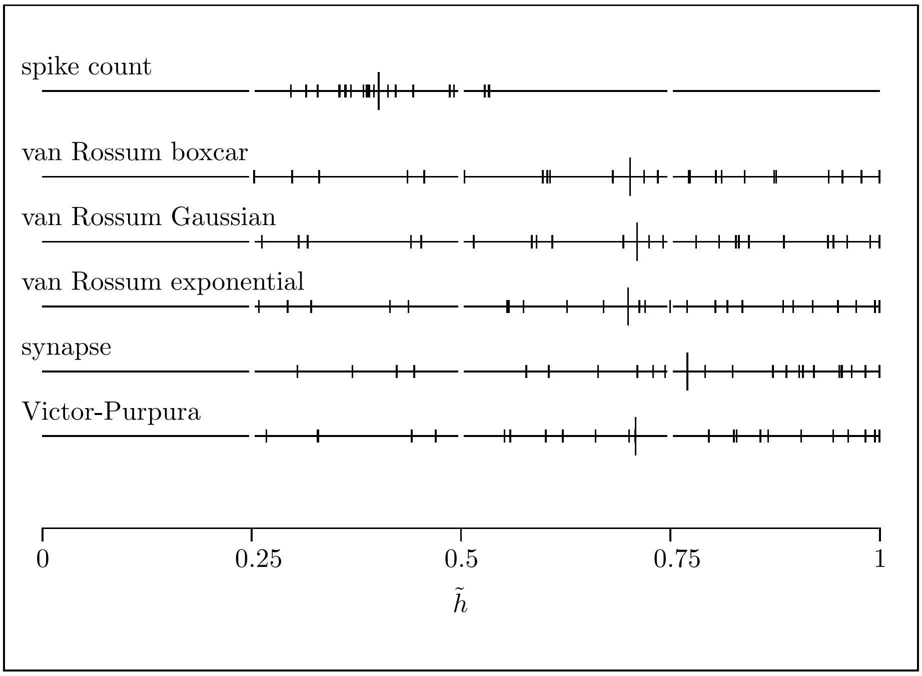


Figure 4: Comparing the spike count distance and various metrics. In this figure the $\tilde{h}$ value has been plotted for each of the 24 sites in the zebra finch data. Each horizontal line corresponds to the performance of a single metric, the line runs from zero to one, as a visual aid a tiny gap is left at 0.25, 0.5 and 0.75. Along each line a small stroke corresponds to a single site, the long stroke corresponds to the average value. The spike count line shows $\tilde{h}$ calculated for the spike count distance. The remaining lines each show the average best performance of a metric: in each case the value of the parameter, or parameters, has been chosen to give the best average performance.

van Rossum boxcar	$\delta_T = 28$ms
van Rossum Gaussian	$\sigma = 7$ms
van Rossum exponential	$\tau = 13$ms
synapse	$\mu = 0.6$ and $\tau = 13$ms
Victor-Purpura	$2/q = 34$ms

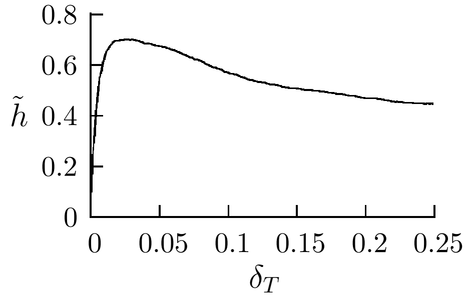


Figure 5: Clustering is optimized by a value of δ_T of approximately 28ms.. This graph shows the transmitted information $\tilde{h}$ of the van Rossum metric with boxcar filter, averaged across all 24 sites in the zebra finch data. It is plotted against δ_T, the window width, with values of δ_T from 1ms to .25s.

Finally, we mention that many studies that have used this approach to demonstrate that spike timing adds to the information that can be decoded from a spike train; for a review, see [21]. It has also been shown that spike timing adds not only quantitatively to what can be decoded, but qualitatively as well: it allows a single neuron to represent the four primary qualities of taste space [6]. While these studies, and many others, demonstrated that spike timing carries information, they stopped short of showing that this information is actually used. This significant step has now been taken in the retina, by comparing behavioral performance to optimal decoding of neural populations [9].

3.1 The synapse van Rossum metric

Figure 4 shows that the synapse van Rossum metric appears to show a clear improvement over the other metrics for the example data set. The average performance of the metric is plotted in Fig. 6; it is seen that increasing μ

when $t < \delta_T/2$ and $t > T - \delta_T/2$, this is easily solved by treating the spike trains as being defined on the extended interval $[-\delta_T/2, T+\delta_T/2]$, but with no spikes outside of $[0, T]$ and this is what is generally done. Here we use a scheme of this sort with the exponential and Gaussian filters. However, for filters with larger timescales this has an undesirable effect: it gives less weight to spikes near the two end-points. Because we examine a broad range of interval widths for the boxcar filter, the other obvious scheme, mapping the spike train onto a circle, is used in that case. In fact, this awkwardness makes very little difference provided the width of the interval is modest compared to the length of the experiment, something that is normally the case.

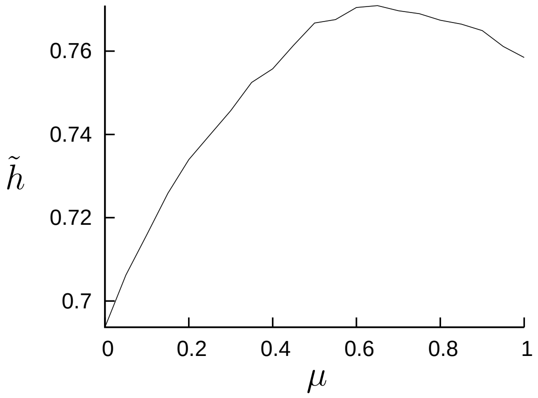


Figure 6: An intermediate value of the depletion parameter μ optimizes transmitted information. Transmitted information, $\tilde{h}$, is plotted against μ with $\tau = 13$ms. When the parameter μ is zero the synapse metric is the same as the van Rossum metric with an exponential filter: each spike adds one to the function. When μ is one the spikes do not have an accumalating effect, each spike returns the function to one. There is a noticeable rise as μ increases, indicating that for this data set the performance of the metric improves.A comparable effect is not seen for other models of the synaptic response.

improves the clustering performance. In Houghton [7] a variety of other models for the synaptic response are tested, models which incorporate long term depression and facilitation and a finite rise time for the conductance. None of these other models has as large an effect on the clustering performance. Of course, this relates only to one particular data set and may change when other data sets are considered.

There are two qualitative differences between the synapse van Rossum metric and the simpler, kernel-based, variants of the van Rossum metric. First, the time of the first spike in a burst is more significant under the synapse metric than the time of the spikes later in the burst. This appears to underlie the improved performance, but of course it is unclear whether this will generalize to other data sets. The second difference is that the synapse-lie metric is not invariant under time reversal. That is, denoting the time reversal of a spike train $\mathbf{u} = \{u_1, u_2, \ldots, u_m\}$by $\bar{\mathbf{u}} = \{T - u_m, T - u_{m-1}, \ldots, T - u_1\}$, we find that typically,:

$$d(\mathbf{u}, \mathbf{v}) \neq d(\bar{\mathbf{u}}, \bar{\mathbf{v}}), \tag{28}$$

However, this asymmetry does not appear to be crucial to the performance of the metric, which is largely the same whether it is carried out on the original zebra finch data or on the time reversed version.

We mention time reversal asymmetry because the dynamics of signaling between neurons is obviously not invariant under time reversal: there are depressing and facilitating effects in both spike production and synaptic responses. It might be expected that these effects are exploited by, or even explained by, the structure of information coding in spike trains. However, if this is the case, it has not been demonstrated by including binding site depletion in the van Rossum metric in this data set.

4 Multineuronal metrics

Data sets consisting of multiple simultaneously recorded neurons are increasingly available. Both kinds of metrics – kernel-based and edit-length metrics – are readily extended to this context. We outline them below, noting that although the multineuronal edit-length metrics are perhaps the more natural, multineuronal kernel-based metrics are considerably easier to calculate when the number of units is large. We describe the edit-length metrics first, since they are simple extensions of the single-unit edit-length metrics discussed above.

In both cases, a metric is an assignment of a distance to pairs of multineuronal responses, $d(\mathcal{U},\mathcal{V})$, where $\mathcal{U}$ and $\mathcal{V}$ each represent a multineuronal response. We can think of a multineuronal response $\mathcal{U}$ as a sequence of "labeled" events: each event is a neuronal firing, and the label is the neuron of origin. We can also think of $\mathcal{U}$ as a set of sub-sequences, one for each neuron. In this view an N-neuron response is $\mathcal{U} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_N\}$ where $\mathbf{u}_1$ to $\mathbf{u}_N$ are all spike trains and,

$$\mathbf{u}_i = \{u_{i1}, u_{i2}, \ldots, u_{im_i}\} \tag{29}$$

and u_{i1} to u_{im_i} are all spike times.

The central question in the analysis of multineuronal activity is, to what extent patterns of activity across the population are crucial for coding. One extreme is that cross-neuron population patterns are simply irrelevant: all that matters is whether any neuron fired, not which one. In short, this is a "summed population code": the message conveyed by the neural population is the same as the message of a single spike train that included all of the population's spikes. At the other extreme, each neuron encodes has a unique and distinguishable role in describing the stimulus. That is, the population is best decoded neuron by neuron, keeping faithful track of which neuron fired which spike. This second possibility is a "labeled line code". The multineuronal metrics formalize these extremes and interpolate between them.

4.1 A multineuronal edit-length metric

As an example of a multineuronal edit-length metric, we show how the Victor-Purpura spike time metric can be extended to the multineuronal context. This can be done simply by adding an extra elementary step: changing the label of a spike. Relabeling a spike is equivalent to moving it from one neuron to another, without changing its time [24, 4]. The other two elementary steps are unchanged; they act on each of the individual spike trains, $\mathbf{u}_i$, in $\mathcal{U}$ to change them into the corresponding spike train, $\mathbf{v}_i$ in $\mathcal{V}$.

The cost assigned to the label-changing step is a parameter, k, that characterizes the metric on the gamut between the summed population code and the labeled line code. If $k = 0$, there is no cost associated with relabeling a spike. The distance is unaffected by which neuron fired each spike, and thus, represents a summed population code. Conversely, if $k \geq 2$, it is at least as expensive to relabel a spike as it is to delete a spike from one spike train and add it to another. In this case, the distance between $\mathcal{U}$ and $\mathcal{V}$ is the sum of the distances between each of the individual spike trains. Thus, $k = 2$ represents a labeled line code. Values of k between these extremes correspond to intermediates between a summed population code and a labeled line code. For example, a value of $k = 1/M$ means that moving M spikes to another neuron has the same effect on the meaning of a response as deleting one spike. Intermediate values of k – in the range of 0.5 to 1 – have been found to provide optimal clustering in an analysis of phase coding in visual cortex [4].

The dynamic programming technique used to calculate the single-unit Victor-Purpura metric can be extended to the multineuronal case. Surprisingly, an efficient strategy for doing this begins by treating the two responses $\mathcal{U}$ and $\mathcal{V}$ in an asymmetric fashion [3]. Specifically, one response, $\mathcal{U}$, is treated as a single labeled event sequence; the other response, $\mathcal{V}$, is treated as a set of single-unit sub-trains. Let $G_{i,j_1,j_2,\ldots,j_N}$ be the distance between, on the one hand, the set of spike trains formed from $\mathcal{U}$ by taking the first i spikes, counted in order by ignoring which spike train they belong to, and on the other hand, the set of spike trains formed from $\mathcal{V}$ by taking the first j_1 spikes in $\mathbf{v}_1$, the first j_2 in $\mathbf{v}_2$ and so on. Initially, we know the values along the edges of the array G where either $\mathcal{U}$ or $\mathcal{V}$ is empty:

$$G_{i,0,0,\ldots,0} = i \tag{30}$$

$$G_{0,j_1,j_2,\ldots,j_N} = \sum_{w=1}^{N} j_w \tag{31}$$

Starting from these edges, the other values can be filled in iteratively using

$$G_{i,j_1,j_2,\ldots,j_N} = \min \left\{ \begin{array}{l} G_{i-1,j_1,j_2,\ldots,j_N} + 1 \\ \min_{w:j_w>0} G_{i-1,j_1,\ldots,j_w-1,\ldots,j_N} + c(i,j_w) \\ \min_{w:j_w>0} G_{i,j_1,\ldots,j_w-1,\ldots,j_N} + 1 \end{array} \right\} \tag{32}$$

where

$$c(i,j_w) = q|u_i - v_{wj_w}| + k[1-\delta(i,j_w)] \tag{33}$$

is the cost of making the ith spike in $\mathcal{U}$ match the j_w spike in $\mathcal{V}$. In this expression, u_i is the time of the ith spike and v_{wj_w} is the time of the j_w spike in $\mathbf{v}_w$; $\delta(i,j_w)$ is one if the two spikes are from the same neuron, and zero otherwise. Once the iteration has finished,the distance between the two spike trains is $G_{m,n_1,n_2,\ldots,n_N}$ where $\mathbf{v}_1$ has n_1 spikes, and so on, and m is the total number of spikes summed over the N spike trains in $\mathcal{U}$.

The drawback of the algorithm is that it requires a large array G and a computation time that is proportional to $m\bar{n}^N$ where m is the total number of spikes in $\mathcal{U}$ and $\bar{n}$ the average number of spikes in the spike trains in $\mathcal{V}$. This often limits the practicality of the approach to recordings of no more than three to five neurons, depending on the length of the spike trains.

As described above, the metric and algorithm posits a single parameter, k, that determines the cost of changing an event's label. When there are spikes from more than two neurons, it is of potential interest to consider metrics in which the cost to change a label depend on the particular label pair. This introduces additional parameters,a symmetric array $k_{w_1w_2}$, the cost to change the label from w_1 to w_2, but no further complexity into the above algorithm. Moreover, the above algorithm can be modified to allow for efficient parallel calculation of the metric for all values of a parameter set k or $k_{w_1w_2}$ [22].

4.2 A multineuronal kernel-based metric

In this section we show how kernel-based metrics can be extended to the multineuronal context. For definiteness, our starting point is the van Rossum metric, but the strategy is general. Recall that the van Rossum metric is defined via a map of spike trains to functions of time, and then the L^2 metric on the space of functions is used to induce a metric back on the space of spike trains. To extend this idea to a multineuronal metric, we begin by extending the target space of the map, [8]: namely, each of the N spike trains in the set is mapped to a vector of N functions of time. We will then use a standard distance in this larger space to induce a metric on the original multineuronal spike trains. The key consideration is to include sufficient flexibility into the initial map so that these metrics can describe population codes, labeled line codes, and the gamut between them. The way we do this is to map each spike train into its own "direction" in the function space. We anticipate that when the directions are aligned, the metric corresponds to a summed population code. When the directions diverge, the metric corresponds to a labeled line code.

To implement this idea, suppose that the single-unit metric is based on the map

$$\mathbf{u}_w \mapsto f(t; \mathbf{u}_w). \tag{34}$$

This could be the linear kernel-based map (13), but the approach here is general, and, $f(t; \mathbf{u}_w)$ could equally well be the synapse map. To extend this map to the multineuronal context, we replace eq. (34) by a map

$$\mathbf{u}_w \mapsto f(t; \mathbf{u}_w)\mathbf{e}_w \tag{35}$$

where $\mathbf{e}_w$ is a N-dimensional unit vector associated with the wth neuron. That is, the unit vectors $\mathbf{e}_w$ determine the direction in which the wth spike train is mapped. Adding these vectors, one from each single-unit spike train, gives an N-dimensional vector of functions of time:

$$\mathcal{U} \mapsto \mathbf{f}(t; \mathcal{U}) = \sum_{w=1}^{N} f(t; \mathbf{u}_w)\mathbf{e}_w. \tag{36}$$

We now define a metric on the multineuronal responses in terms of a metric on the space of vectors of functions. The obvious choice is based on the norm:

if

$$\mathbf{g}(t) = \begin{pmatrix} g_1(t) \\ g_2(t) \\ \vdots \\ g_N(t) \end{pmatrix} \tag{37}$$

is a vector of functions of $t \in [0, T]$, then the norm is

$$\|\mathbf{g}(t)\| = \sqrt{\int_0^T dt(g_1^2 + g_2^2 + \ldots + g_N^2)}. \tag{38}$$

The corresponding metric induced on the space of multineuronal responses is given by

$$d(\mathcal{U}, \mathcal{V}) = \|\mathbf{f}(t;\mathcal{U}) - \mathbf{f}(t;\mathcal{V})\|. \tag{39}$$

We still need to specify the unit vectors $\mathbf{e}_w$ used in the embedding. As long as the metric is determined by a Euclidean distance in the target space, only the angles between these vectors, not their absolute directions, influence the value of the metric space. If the $\mathbf{e}_w$ are all parallel, then the above map eq. (36) corresponds to summing the individual response vectors $f(\mathbf{u}_w)\mathbf{e}_w$. This is equivalent to superimposing the spike trains before mapping them in to function space, and thus, precisely corresponds to a summed population code. Conversely, if all the vectors are orthogonal, for example, if $\mathbf{e}_w$ has a one for its w component and is otherwise zero, then the multineuronal metric is a Pythagorean sum of the individual van Rossum distances between the individual spike trains: this is a labeled line code.

In the two neuron case, there are two vectors, $\mathbf{e}_1$ and $\mathbf{e}_2$ and, the metric is defined by the angle between them. This angle plays the same role as k in the multineuronal Victor-Purpura multineuronal metric: it maps out a one-parameter interpolation between the summed population code and the labeled line code. When there are more than two vectors, the situation is more complicated: there are angles between each pair of vectors, and $N(N-1)/2$ parameters are required to specify all these angles. This is strictly analogous to the situation that arises for edit-length distances: one could force all of these parameters to have the same value, corresponding to a single choice of the relabeling parameter k, or one could allow these values to correspond to a symmetric matrix, corresponding to different costs for each kind of relabeling. An explicit construction for a set of basis vectors that share a common angle is given in [8].

There is also an L^1 version of this multineuronal metric:

$$d(\mathcal{U},\mathcal{V}) = \|\mathbf{f}(t;\mathcal{U}) - \mathbf{f}(t;\mathcal{V})\|_1. \tag{40}$$

with

$$\|\mathbf{g}(t)\|_1 = \int_0^T dt(|g_1| + |g_2| + \ldots + |g_N|) \tag{41}$$

and $\mathbf{g}(t)$ a vector of function as in Eq. (37). The vectors $\mathbf{e}_w$ are again unit vectors, but note that they must be chosen to be unit vectors according to the l^1 norm. That is, the absolute values of their components must add to one. Thus, for example, in the two-neuron case a suitable pair of vectors would be given by $\mathbf{e}_1 = (1,0)$ and $\mathbf{e}_2 = (1-k/2, k/2)$ for k between zero and two. The parameter $k/2$ has been chosen so that there is a distance k between the two one-spike spike trains

$$\begin{aligned}\mathcal{U} &= \{\mathbf{u}_1 = \{u\}, \mathbf{u}_2 \text{ empty}\}\\ \mathcal{V} &= \{\mathbf{v}_1 \text{ empty}, \ \mathbf{v}_2 = \{v\}\}\end{aligned}$$

with $u = v$: spikes are at the same time but correspond to different neurons. With the boxcar kernel,this metric partially mimics the multineuronal edit-distance metric in that $k = 0$ corresponds to a population code, and $k = 2$ corresponds to a labeled line code. However, the two metrics give different distances even for one-spike spike trains. If $2/q > v - u = t > 0$ the multineuronal van Rossum metric gives $qt + k - qk/2$ compared to the multineuronal edit distance which gives $qt + k$, or two, whichever is smaller.

To pursue the goal of selecting a metric on the basis of maximizing the faithfulness of clustering – or optimizing the transmitted information – it is necessary to carry out repeated computation of the metric distance for different values of the metric parameters. Here, multineuronal van Rossum metrics has the advantage of generally being far faster to compute numerically; the computational difficulty is quadratic in the total number of spikes, and does not have an exponential dependence on the number of neurons – as the edit-length distances do. Thus, it should be possible to use van Rossum metrics to study data sets with large numbers of neurons, allowing the optimal angle parameter to be calculated as a description of the nature of population coding for these neurons.

4.3 The Earth Mover Distance

The strategies used above to extend the edit-length and kernel-based metrics to the multineuronal context have a common thread: they both can be considered to use "neuron of origin" as a second domain variable, along with time. Here we describe the "Earth Mover Distance", because it can be viewed as taking this idea one step further, making them applicable to images [15, 16].

To make this transition, we begin with the viewpoint of the multineuronal edit-length metrics: spikes can be moved across time, or across neurons. That is, time and neuron-of-origin are two "domain" variables, and they are treated separately.[7] The class of metrics represented by the Earth Mover Distance is the result of considering these two variables as a single two-dimensional domain. This change in viewpoint means that a spike at time x and neuron y simply becomes an "event" at (x, y). We can now define the cost of an elementary step to move an event from (x_1, y_1) to (x_2, y_2) in terms of the distance between (x_1, y_1) and (x_2, y_2), and allow moves in oblique directions. To complete the connection with the Earth Mover Distance, we note that an array of "events" has an obvious reinterpretation as images: the event density, $I(x, y)$, at (x, y) corresponds to image intensity at (x, y). As the name implies, if we interpret an image as a pile of earth at height (x, y), the Earth Mover Distance indicates how much labor, volume×distance, must be done to convert one image into another. The Earth Mover Distance has been found useful for color-based image retrieval where pictures are mapped to their color histograms and the Earth Mover Distance is used to measure the similarity of these histograms [16, 15] .

As defined above, the Earth Mover Distance is built up from elementary steps, each of which preserves the total amount of "earth", that is, the total event count $\int I(x, y)dxdy$ is preserved by the elementary step. Thus, to ensure that a distance always exists, one needs at least one more ingredient – such as an elementary step that adds or subtracts earth, or, a convention that all images have been normalized to contain the same amount of "earth". We also mention that many variations on the above theme are readily framed. To compare flat images, the domain variable is naturally a two-dimensional rectangle; but movies, volume-based "images", and volume-

[7] This viewpoint can also be applied to the van Rossum metrics: the target of the mapping is a function of time and neuron with the mapping from each neuron-of-origin to the neuron direction in the target space being determined by the choice of the unit vector $\mathbf{e}_w$.

based movies, such as functional imaging data, can be similarly compared on a three or four-dimensional domain of spatial variables and time. In all of these cases, the cost for moves need not be isotropic, and need not even reflect a Euclidean distance.

5 Conclusions

We have presented strategies for metric-based analysis of spike trains in a relatively abstract manner, to emphasize their conceptual underpinnings and ways in which they can be generalized.[8] It is hoped, above all, to find a framework for describing spike trains that will make it easier to understand the relationship between the spiking activity of neurons and the computational function of neuronal circuits. We have tried to give some motivation for the two main families of spike train metrics, the single neuron metrics are perhaps easier to relate to our current understanding of spike coding and of physiology. The multineuronal metrics are primarily justified by the fact that they give a full gamut of metrics interpolating between the labeled line and summed population codes. It seems very clear that there are many open questions at this interesting interface of mathematics and neuroscience, and we wish to close by making some of them explicit.

The largest question is whether the metric-space framework is sufficiently general. It is well-known that there are cognitive phenomena that do not map readily into metric-space descriptions [12, 18, 19]. Perhaps it is possible to model such non-metric behaviors at the level of the decision process, but we cannot rule out the possibility that a more general mathematical structure will be needed even at the level of spike trains.

As an example of a more general structure for spike trains, one might consider a non-metrizable topological space. Here, the notion of a "distance" is replaced by the notion of "neighborhoods": a collection of open sets within the space. A metric space is necessarily a topological space, but the latter is more general: there are topological spaces where the topology (i.e., the set of neighborhoods) can not be re-created from a metric. An obvious, relevant, example of a non-metric topological space is the topological space induced by a semimetric, a distance function which does not satisfy the triangle inequality. The windowed coincidence detection distance mentioned in Subsect. 2.4

[8] From a practical point of view, robust implementations of many of the metrics discussed here can be found on `neuroanalysis.org`

is a semimetric. semimetric spaces do not, in general, have nice geometries with useful properties;perhaps a metric is required to endow the space of spike trains the properties we would like it to have. However, it should be noted that algorithms for distance-based clustering and classification generally work equally well with a semimetric distance function [10, 11]. The fundamental issue is,what properties we require the space of spike trains to possess from a biological point-of-view and how these requirements translate into constraints on the topological or geometric structure of the space of spike trains.

Even within the context of spike train metrics, we cannot claim that we have been comprehensive. We have presented two kinds of metrics, both because of their ease of use and the biological intuitions that they capture. However, we do not wish to suggest that these are the only families that should be considered. More generally, it would be desirable to have a first-principles "taxonomy" of biologically relevant metrics. Of course the preliminary challenge is formalizing the latter phrase: while we can readily come up with bizarre metrics – for examples, metrics based on pathologies of real analysis such as Cantor sets – it is not so easy to formalize the characteristics that a metric must have to make it worth considering in neuroscience. At its simplest such a characterization would list properties that are germane to spike trains, these might include a jitter requirement – a small change to spike times should cause only small change to distances – and an unreliability requirement: similar spike trains can have different numbers of spikes. A more ambitious scheme would attempt to derive a metric from a model for the genesis of the geometry of spike train space.

Finally, we speculate on a possible avenue to approach the latter problem. A neuron – or a neuronal network – can be thought of as carrying out a transformation T on multineuronal spike trains $\mathcal{U}$. This can also be viewed as a transformation $\widetilde{T}$ on the space of all metrics: that is, given any metric d that we assign to the way in which output activity $T(\mathcal{U})$ is evaluated, we can induce a metric $\widetilde{T}(d)$ on the input by asking how different the input trains appear, once they have been processed,

$$\widetilde{T}(d)(\mathcal{U},\mathcal{V}) = d(T(\mathcal{U}),T(\mathcal{V})). \tag{42}$$

This formalizes the idea that a distance between spike trains should reflect the difference between their effects on neural activity. Given this structure, one could then hope to study the structure of the metric d under the assumption that the transformation $\widetilde{T}$ acts on it in a straightforward way: the metric might be stable fixed points, or have parameters that undergo a simple transformation. This reflects the heuristic that the overall relationship between neural activity need not be re-invented as neural activity evolves over time.

Acknowledgments.

We are grateful to Kamal Sen for the use of the data analyzed here. CH is supported by Science Foundation Ireland grant 08/RFP/MTH1280, JV is supported by NIH EY9314 and NIMH MH68012 (the latter to D. Gardner).

References

[1] Abbott, L. F., Varela, J. A., Sen, K., and Nelson, S. B. (1997). Synaptic depression and cortical gain control. *Science*, 275:221–224.

[2] Abeles, M. (1982). *Local cortical circuits: an electrophysiological study.* Springer.

[3] Aronov, D. (2003). Fast algorithm for the metric-space analysis of simultaneous responses of multiple single neurons. *Journal of Neuroscience Methods*, 124:175–179.

[4] Aronov, D., Reich, D. S., Mechler, F., and Victor, J. D. (2003). Neural coding of spatial phase in V1 of the macaque monkey. *Journal of Neurophysiology*, 89:3304–3327.

[5] Aronov, D. and Victor, J. D. (2004). Non-euclidean properties of spike train metric spaces. *Physical Review E*, 69:061905.

[6] Di Lorenzo, P. M., Chen, J. Y., and Victor, J. D. (2009). Quality time: representation of a multidimensional sensory domain through temporal coding. *Journal of Neuroscience*, 29:9227–38.

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

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

[9] Jacobs, A. L., Fridman, G., Douglas, R. M., Alam, N. M., Latham, P. E., Prusky, G. T., and Nirenberg, S. (2009). Ruling out and ruling in neural codes. *Proceedings of the National Acadamy of Science U.S.A.*, 106:5936–41.

[10] Lance, G. N. and Williams, W. T. (1967a). A general theory of classificatory sorting strategies: I. Hierarchical systems. *The Computer Journal*, 9:373–380.

[11] Lance, G. N. and Williams, W. T. (1967b). A general theory of classificatory sorting strategies: II. Clustering systems. *The Computer Journal*, 10:271–277.

[12] Maloney, L. T. and Yang, J. N. (2003). Maximum likelihood difference scaling. *Journal of Vision*, 3:573–585.

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

[14] Ratliff, F., Knight, B. W., and Graham, N. (1969). On tuning and amplification by lateral inhibition. *Proc. Natl. Acad. Sci. U S A*, 62:733–40.

[15] Rubner, Y., Guibas, L., and Tomasi, C. (1997). The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. *Proceedings of the ARPA Image Understanding Workshop*, pages 661–668.

[16] Rubner, Y., Tomasi, C., and Guibas, L. J. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40:99–121.

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

[18] Tversky, A. (1977). Features of similarity. *Psychological Review*, 84:327–352.

[19] Tversky, A. and Gati, I. (1982). Similarity, separability, and the triangle inequality. *Psychological Review*, 89:123–54.

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

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

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

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

[24] Victor, J. D. and Purpura, K. P. (1997). Metric-space analysis of spike trains: theory, algorithms and application. *Network: Computation in Neural Systems*, 8:127–164.

[25] Volterra, V. (1932). *Theory of functionals and of integral and integro-differential equations*. Blackie.

[26] Wang, L., Narayan, R., Graña, G. M., Shamir, and Sen, K. (2007). Cortical discrimination of complex natural stimuli: can single neurons match behavior? *Journal of Neuroscience*, 27:582–9.

[27] Wuerger, S. M., Maloney, L. T., and Krauskopf, J. (1995). Proximity judgments in color space: Tests of a Euclidean color geometry. *Vision Research*, 35:827–835.