Chapter 7
Spike Metrics

Jonathan D. Victor and Keith P. Purpura

Abstract Important questions in neuroscience, such as how neural activity represents the sensory world, can be framed in terms of the extent to which spike trains differ from one another. Since spike trains can be considered to be sequences of stereotyped events, it is natural to focus on ways to quantify differences between event sequences, known as spike-train metrics. We begin by defining several families of these metrics, including metrics based on spike times, on interspike intervals, and on vector-space embedding. We show how these metrics can be applied to single-neuron and multineuronal data and then describe algorithms that calculate these metrics efficiently. Finally, we discuss analytical procedures based on these metrics, including methods for quantifying variability among spike trains, for constructing perceptual spaces, for calculating information-theoretic quantities, and for identifying candidate features of neural codes.

7.1 Introduction

7.1.1 Mathematics and Laboratory Data

Mathematical analysis of laboratory data plays a crucial role in systems neuroscience. Perhaps the most fundamental reason is that often, the questions that we ask of data are abstract. A prime example is the investigation of neural coding—delineation of the relationship between stimuli, actions, and/or behavioral states, and the activity of one or more neurons.

An invited chapter for "Analysis of Parallel Spike Trains" (S. Rotter and S. Grün, eds.)

J.D. Victor (✉)
Division of Systems Neurology and Neuroscience, Department of Neurology and Neuroscience, Weill Cornell Medical College, 1300 York Avenue, New York, NY 10065, USA
e-mail: jdvicto@med.cornell.edu
url: http://vivo.cornell.edu/individual/vivo/individual6150

S. Grün, S. Rotter (eds.), *Analysis of Parallel Spike Trains,*
Springer Series in Computational Neuroscience 7,
DOI 10.1007/978-1-4419-5675-0_7,

To apply mathematical analysis to laboratory data in a principled way, a necessary first step is to choose an appropriate mathematical framework. This would be simple if laboratory data corresponded precisely to mathematical entities, but this is rarely the case (Slepian 1976). Laboratory data do not permit one to apply methods that require taking limits as time goes to infinity or to zero, and one does not have access to statistical ensembles, just individual experiments. Thus, explicitly or implicitly, the neuroscientist must identify and abstract the essential features of the data that are relevant to the problem at hand or the goal of the analysis. We therefore begin with a discussion of these considerations as they relate to the electrical activity of neurons and to the problem of neural coding.

7.1.2 Representing Spike Trains as Samples of Point Processes

The electrical activity of individual neurons can be recorded extracellularly, intracellularly, and somewhat more indirectly by optical means. The electrical activity, as often measured with micro- or macroelectrodes, consists of a combination of small voltage fluctuations (on the order of 10 mV), upon which are superimposed larger, brief "action potentials": stereotyped voltage transients of approximately 100 mV that last a fraction of a millisecond. Since action potentials propagate without loss and result in the release of neurotransmitter, they are generally considered to represent the components of a neuron's activity that are "seen" by the rest of the nervous system. Consequently, the sequences of action potentials emitted by individual neurons (i.e., "spike trains") are a natural focus for the study of brain activity at the level of cells and circuits (Segundo and Perkel 1969; Abbott 2000; Sen et al. 1996). To abstract the brief, stereotyped nature of spike trains, we choose to represent them as instances of point processes, i.e., event sequences.[1]

7.1.3 Analyzing Point Processes: The Rationale for a Metric-Space Approach

The point-process representation (see Chap. 1) has substantial implications for the choice of signal-processing strategies. Had we chosen a vectorial representation[2]

[1] Waveforms of real action potentials are not completely stereotyped, and at least some of the waveform-to-waveform differences are systematic. By representing neural activity as a point process, we make the conscious decision to ignore these differences, as is typically done both in this book and in general. However, this simplification is not as restrictive as it might at first seem. By representing action potentials as instances of point processes, we are not assuming that every action potential has the same effect on post-synaptic neurons; we are merely assuming that the differences in the effects of each action potential can be understood from their temporal pattern. This makes good biological sense, since temporal pattern (in particular, the time since the preceding spike) is a crucial factor in governing both transmitter release and the subtle variations in action potential shape.

[2] More specifically, a Hilbert space—because the dot-product is defined.

(e.g., had we represented neural activity as a continuous voltage record), certain algebraic operations would immediately be defined—the usual vector-space operations of addition, multiplication by scalars, and inner ("dot") product. These operations are the basis of signal-processing methods such as filtering, averaging, spectral estimates, and signal detection. For point processes, these operations are not applicable, and these vector-based signal-processing methods cannot be directly applied.

While this might at first seem to be a severe disadvantage of the point process framework, further consideration shows that this is not the case. The reason is that vector-based procedures require the imposition of a high degree of mathematical structure, and this structure is often not appropriate or practical. One reason for this is that in a vector space, linearity plays a fundamental role, and this is arguably at odds with the nature of neural dynamics. A second reason is that the dot-product induces an intrinsic Euclidean geometry. By representing spike trains as samples of point processes, we are driven to analytical procedures that do not require us to define vector-space operations (e.g., how two spike trains can be added). The point process perspective can thus allow the researcher to investigate a wider gamut of behavior in the spike trains.

Two examples from sensory physiology emphasize this point. Human color vision is a three-parameter space determined by the three cone absorption spectra, and thus, one might hope that a three-dimensional vector space would provide an appropriate representation for color percepts. However, experimental measurement of perceptual distances within this three-dimensional space shows that perpendiculars need not be unique (Wuerger et al. 1995)—in contrast to the requirements of a Euclidean space. In olfaction, the situation is far more complex. Olfactory perceptual space may not even have a well-defined dimension, mixing of odorants need not lead to intermediate percepts, and gross violations of linearity are present (Hopfield 1995). Thus, the Euclidean geometry implied by vector spaces may be too confining to support a correspondence between neural activity and sensory perception.

To broaden our scope, we consider analytic procedures in which relationships between pairs of spike trains play a central role, but we do not require that we know how to "add" spike trains or to multiply them by scalars. Instead, our basic operation is a way to compute a distance (i.e., a measure of dissimilarity) between two spike trains, $d(A, B)$. In keeping with the philosophy of limiting our assumptions, we only require that the distances $d(A, B)$ are a "metric" (defined below). That is, spike trains are considered to be points in a metric space, rather than a vector space. While distances derived from vector spaces are necessarily metrics, the distances derived from metric spaces typically cannot be mimicked by a vector space. Vector space distances are unchanged by linear transformation; no corresponding property holds for metrics in general. Typical metric spaces, including the several of those we consider below, do not have Euclidean geometry (Aronov and Victor 2004). Thus, metric spaces (and the distances that define them) are much more general than vector spaces.

The reader might wonder about mathematical frameworks that are even more general. For example, the notion of a "topology" on the set of spike trains is even more general than that of a metric (see Singh et al. 2008 for a recent application of

this notion to spike-train analysis). A topology merely requires that we define relationships between spike trains via a system of "open sets", while a metric requires that we quantify distances (and these distances in turn define the open sets). Fundamentally, there is no a priori reason to exclude nonmetrizable topologies, though such spaces are typically quite strange and nonintuitive. Conversely, the notion of "distance" is widely applicable to our current understanding of perception and behavior, and can be readily quantified, at least in principle, for example, as the number of "just noticeable differences" between two stimuli. Moreover, a compilation of the distances between all pairs of points describes the intrinsic geometry of a space. Thus, the metric-space framework is capable of capturing the essence of what one hopes to account for (the intrinsic structure of the space represented by neural activity) and carries with it only minimal restrictions. Finally, we point out that under some circumstances, perceptual comparisons (Tversky and Gati 1982; Tversky 1977; Maloney and Yang 2003) and other components of cognition and learning may be nonmetric in that they violate either the postulate of symmetry or the triangle inequality. Generalizations of metrics that accommodate this behavior are straightforward; the reader is referred to (Victor et al. 2007) for further details.

We emphasize that we are not suggesting a single best way to define distances in neurophysiological data. Rather, we present the metric approach (especially as it applies to neural coding) as a general strategy to formalize biologically motivated hypotheses concerning the meaningful features of spike trains and to determine whether these hypotheses are supported by observed neural activity associated with sensory and motor events.

7.1.4 Plan for this Chapter

We begin with some formal preliminaries. We then define several families of metrics for spike trains, first considering metrics applicable to single-neuron responses and then metrics applicable to multineuronal activity. We then describe several algorithms by which these metrics can be calculated. We conclude by discussing a range of analytical procedures based on these metrics that can provide insight into spike trains and their relationship to behavior. These procedures include methods of quantifying variability among spike trains, describing their relationship to perceptual spaces, and information-theoretic techniques aimed at identifying candidate features of neural codes.

7.2 Spike Train Metrics

In this section, we define several families of metrics for spike trains. We emphasize "edit-length" metrics, which are specifically applicable to event sequences and yield distances that are fundamentally distinct from vector-space distances. Following this, we consider a second class of metrics that are consequences of embedding spike trains into vector spaces.

7.2.1 Notation and Preliminaries

We represent the activity of a single neuron as a sequence of stereotyped events. Correspondingly we represent multineuronal activity as a sequence of labeled events. More formally, we represent a spike train A observed over a period $[0, T]$ by a sequence of distinct real numbers $t_1, \ldots, t_{M(A)}$, the times of occurrence of the spikes. $M(A)$ is the total number of spikes (and may be 0), and the spike times are assumed to be listed in increasing order. When considering the activity of multiple neurons, we augment this framework by adding labels $l_1, l_2, \ldots, l_{M(A)}$. The labels l_j are drawn from a set $\{1, \ldots, L\}$ of abstract tags, indicating which neuron was responsible for each spike.

A metric $d(A, B)$ is a mapping from pairs of spike trains to the nonnegative real numbers. It satisfies three properties, namely (i) $d(A, B) > 0$, with equality only when $A = B$, (ii) symmetry: $d(A, B) = d(B, A)$, and (iii) the triangle inequality, $d(A, C) \leq d(A, B) + d(B, C)$. With these three conditions, the present use of the term "metric" is consistent with the topological definition of this term (Gaal 1964) and endows the set of event sequences (the spike trains) with the properties of a topological "metric space".[3]

7.2.2 Cost-Based (Edit Length) Metrics

7.2.2.1 General Definition

A simple, intuitive strategy enables the construction of metrics that formalize a range of biologically motivated notions of similarity. The common element in these metrics is that of a set of "elementary steps" between two spike trains, each of which has an associated nonnegative cost. We require that the cost $c(X, Y)$ of an elementary step from X to Y is symmetric and that for any two spike trains A and B, it is possible to find some sequence of elementary steps that begins at A and ends at B.

Any set of elementary steps satisfying these conditions leads to a metric between spike trains: the cheapest total cost to transform A to B via elementary steps. More formally, we define

$$d(A, B) = \min\left\{ \sum_{j=0}^{n-1} c(X_j, X_{j+1}) \right\}, \tag{7.1}$$

[3] Below we will also want to consider a very simple distance that is not, strictly speaking, a metric. This is the "spike count distance" $D^{\text{spike}}[0]$. For the spike-count distance, the distance between two spike trains is given by the difference in the number of spikes they contain. Thus, we can have $D^{\text{spike}}[0](A, B) = 0$ for distinct spike trains A and B if they contain the same number of spikes. While this distance is not strictly a metric, the formal structure of a metric space still applies, because we can think of $D^{\text{spike}}[0]$ as acting on the equivalence classes of "distinguishable" spike trains, rather than on the spike trains themselves.

where $\{X_0, X_1, \ldots, X_n\}$ is a sequence of spike trains with $X_0 = A$ and $X_n = B$. Metrics defined in this fashion are guaranteed to satisfy the triangle inequality, since the cheapest path from A to C cannot be more expensive than a path that is constrained to stop at B. These cost-based distances are analogous to "edit-length" distances for symbol sequences, including those used in analysis of EEG data (Wu and Gotman 1998), and, more prominently, in comparison of genetic sequences (Sellers 1974; Needleman and Wunsch 1970). In this analogy, each of the elementary steps can be considered to be a way of "editing" the spike train. The minimum total cost of the elementary transformations, the "edit-length", quantifies the dissimilarity of the spike trains. Thus, efficient dynamic programming algorithms for comparison of genetic sequences (Sellers 1974; Needleman and Wunsch 1970) can be adapted to calculate many kinds of spike metrics (Victor and Purpura 1997, 1996). Interestingly, although these algorithms have been in use for almost 40 years for genetic sequence comparison, their use in neuroscience is much more recent.

7.2.2.2 Spike Time Metrics

As a first example of the strategy described above, we create a family of metrics that are sensitive to the timing of individual spikes (Victor and Purpura 1997, 1996). The biological motivation for these metrics is that neurons have a firing threshold. As a consequence, neurons can often be regarded as coincidence detectors (Softky and Koch 1993; Egger et al. 1999; Kuba et al. 2005; Abeles 1982): one spike arriving on a presynaptic neuron will not cause a postsynaptic neuron to fire, but a sufficient number of spikes arriving within a sufficiently narrow time window will cause the postsynaptic neuron to fire. Thus, the "meaning" of a spike train depends on the timing of its spikes.

To capture this dependence in a metric, we invoke the above machinery with two kinds of elementary steps (Fig. 7.1). The first elementary step consists of inserting or deleting a spike and is assigned a cost of 1. This rule ensures that every spike train can be transformed to any other spike train by *some* path: the path that successively deletes all spikes from one train and then successively inserts all spikes into the second train. The second elementary step defines the sensitivity to spike timing. It consists of moving a single spike, and the associated cost is proportional to the amount of time that the spike is moved. That is, if two spike trains X and Y are identical except for a single spike that occurs at t_X in X and t_Y in Y, then

$$c(X, Y) = q|t_X - t_Y|. \tag{7.2}$$

Along with (7.1), this set of elementary steps defines a spike time metric, $D^{\text{spike}}[q]$. Note that these metrics constitute a parametric family, parameterized by the value of q that specifies the cost per unit time to move a spike. Since moving a spike by an amount $\Delta T = 1/q$ has the same cost as deleting it altogether, q can be viewed as determining the relative sensitivity of the metric to spike count and spike timing.

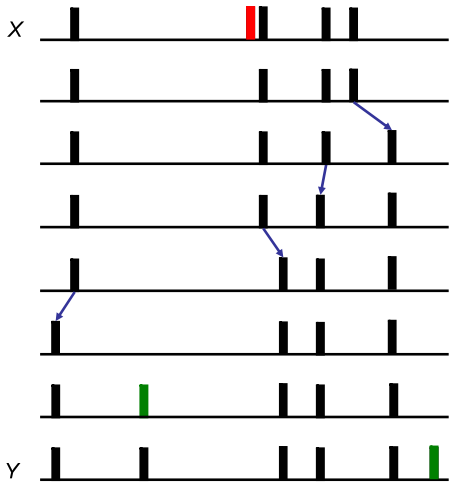

Fig. 7.1 A diagram of a sequence of elementary steps that transforms spike train X into spike train Y. Each elementary step is one of three types: deletion of a spike (deleted spike shown in *red*), insertion of a spike (inserted spike shown in *green*), or shifting a spike in time (*blue arrows*)

The character of $D^{\text{spike}}[q]$ depends strongly on q. When q is small, then the times of individual spikes have little influence on the calculated distance between spike trains. In the limit of $q = 0$, spikes can be shifted freely in time, and $D^{\text{spike}}[0](A, B) = |M(A) - M(B)|$. So $D^{\text{spike}}[0]$ corresponds to comparing spike trains in terms of their spike counts alone. As q increases, the metric $D^{\text{spike}}[q]$ becomes increasingly sensitive to spike timing: a change in the time of a spike by $1/q$ sec has the same cost as deleting the spike altogether. That is, for neurons that act like a coincidence detector with integration time (or temporal resolution) $1/q$, spike trains will have similar postsynaptic effects if they are similar in the sense quantified by $D^{\text{spike}}[q]$. Since the effective temporal resolution is typically not known in advance, it is useful to carry out analyses across a range of values of q, rather than specifying the resolution a priori.

7.2.2.3 Spike Interval Metrics

Within the same framework of (7.1), we next define a contrasting metric (Victor and Purpura 1997, 1996), one that is sensitive to the pattern of spike intervals, rather than individual times. The biological motivation is that the postsynaptic effects of a spike may depend strongly on the recent activity at that synapse (Sen et al. 1996; Dan and Poo 2004; Markram et al. 1997). For transformations dominated by this dependency, the meaning of a spike train does not depend on the spike times themselves (as captured by $D^{\text{spike}}[q]$) but rather on the sequence of intervals. To quantify this notion, we define $D^{\text{interval}}[q]$ by (7.1), along with a different set of elementary steps. For $D^{\text{interval}}[q]$, the first kind of elementary step is insertion or deletion of *an interspike interval*, at a cost of 1. The second elementary step consists of shortening or extending an existing interspike interval. The cost of this step is equal to $q\Delta T$, where ΔT is the amount of time by which the interval has been lengthened

or shortened.[4] Since changing the length of an interval changes the time of occurrence of all subsequent spikes at no additional cost, $D^{\text{interval}}[q]$ and $D^{\text{spike}}[q]$ have fundamentally different topological characteristics (Victor and Purpura 1997).

7.2.2.4 Multineuronal Cost-Based Metrics

We now consider ways of extending the above metrics to multineuronal data. While multineuronal extensions can be constructed for either of the above single-neuron metrics, the extension is more natural for $D^{\text{spike}}[q]$, so we focus on it.

Recall that our formal representation of multineuronal activity A is a sequence of $M(A)$ spike times t_j, each of which is associated with a label $l_j \in \{1, \ldots, L\}$ that indicates its neuron of origin. To extend $D^{\text{spike}}[q]$ to the multineuronal context (Aronov et al. 2003), we add another elementary step, consisting of changing the label associated with an event. The simplest way to do this is to assign the same cost, a parameter k, to any label change. These three steps, along with (7.1), define a two-parameter family of metrics, $D^{\text{spike}}[q, k]$, where for $k = 0$, the metric ignores the label associated with each event. The metrics with $k = 0$ correspond to a "summed population" of neural activity. Conversely, when $k = 2$, the impact of the label is maximal, since it costs as much to change the label as it does to delete a spike and then reinsert it with a new label. Metrics with $k = 2$ correspond to a "labeled lines" interpretation of neural activity.

7.2.2.5 Other Cost-Based Metrics

There are many directions in which the above examples can be generalized, and we present some of the more important ones here. In each case, additional degrees of freedom are added to the metrics described above. This flexibility arguably provides a closer approximation to biologic reality but often carries the penalty that a greater amount of data is required for a meaningful analysis. We also indicate which extensions are readily incorporated into the dynamic programming algorithms described in Sect. 7.2.2.6.

More Flexible Assignments of Costs to the Elementary Steps First, the cost of moving a spike by an amount of time Δt (or the amount by which an interval is changed) need not be proportional to Δt.

[4] Special consideration needs to be applied to the interval between the start of data collection and the first spike and to the interval between the last spike and the end of data collection. These are not interspike intervals, since they are bounded by the limits of data collection, rather than a second spike. It is therefore natural to take the view that each of these "intervals" could be considered an exact match to any interspike interval that is at least as long. This is equivalent to minimizing the distance to any spike train that matches the observed spike train within the data collection interval. A simpler (but arguably more arbitrary) strategy is simply to place an artificial spike at the beginning and end of data collection.

For example, $D^{\text{spike}}[q]$ can be generalized by replacing the cost defined in (7.2) by

$$c(X,Y) = Q\big(|t_X - t_Y|\big) \tag{7.3}$$

for any nondecreasing cost function $Q(|\Delta T|)$ satisfying $Q(0) = 0$ and

$$Q\big(|\Delta T| + \big|\Delta T'\big|\big) \le Q\big(|\Delta T|\big) + Q\big(\big|\Delta T'\big|\big). \tag{7.4}$$

This does not interfere with the dynamic programming algorithms, since (7.4) means that breaking down a single elementary step into two smaller ones can never decrease the total cost.

Another generalization is that the cost to move a spike could depend on its absolute time and on the distance moved. General dependences of this sort will interfere with the dynamic programming algorithm, since they allow for a minimal path in which a spike moves twice, once to enter a region of low-cost movement and once to move to its final position. But certain forms of this dependence are consistent with the dynamic programming algorithm. One such form is

$$c(X,Y) = q\big(\big|F(t_X) - F(t_Y)\big|\big), \tag{7.5}$$

where F is an increasing function. That is, $\tau = F(t)$ represents a distortion of time, and the cost to move a spike is uniform in the distorted time τ. Because of this distortion, the cost assignment of (7.5) allows for the metric to have a nonuniform dependence on spike times in the original time t. In particular, for small displacements, (7.5) can be approximated by

$$c(X,Y) \approx F'\left(\frac{t_X + t_Y}{2}\right) q\,\Delta T. \tag{7.6}$$

So for example, a decelerating function F would lead to a stronger dependence of the metric on the time of early spikes than on the time of late spikes.

Since the cost of moving a spike determined by (7.5) is related to a uniform cost through a time-distortion the corresponding metric (7.1) can be calculated from $D^{\text{spike}}[q](F(A), F(B))$, where $F(X)$ denotes a spike train derived from X by the time-distortion $\tau = F(t)$.

Other Kinds of Elementary Steps The kinds of elementary steps used to define a cost-based metric can be thought of as formalizing a hypothesis that certain aspects of spike train structure are meaningful. Above we have considered two contrasting metrics, focusing on absolute spike times and on spike intervals; these are but two of many possibilities. As another example, we consider the notion that spike times are important, but absolute time is uncertain. This arises in the analysis of spike trains associated with spontaneous motor activity (i.e., not synchronized to an external event or clock). Under these circumstances, spike trains that differ only by an overall translation in time have the same meaning. To capture this in a metric, one could augment $D^{\text{spike}}[q]$ by a step that allows an entire spike train to be translated en bloc at reduced cost per unit time, say, q', with $q' \ll q$.

For this metric, the dynamic programming algorithm for $D^{\text{spike}}[q]$ enables a calculation in polynomial time (see below). The reason is that in this metric, a minimal-cost path between two spike trains can always be found in which the en bloc translation moves at least one spike in train A into coincidence with a spike in train B. To see this, assume that no spikes are in coincidence and consider the effects of changing the size of the block movement by an infinitesimal (signed) amount dT. Say that this block movement is part of a path (7.1) in which there are $n_{A\to B}$ individual spikes that move forward in time between A and B, and $n_{B\to A}$ individual spikes that move backward. Then this infinitesimal en bloc translation incurs a net cost of $(q' - qn_{A\to B} + qn_{B\to A})\,dT$. If this number is nonzero, then either an infinitesimal translation forward or backward would reduce the total cost further. If it is zero, then an infinitesimal movement in either direction would leave the cost unchanged. In either case, further infinitesimal translations could be applied, without incurring additional cost, until at least one spike pair was in coincidence. Thus, to calculate this metric, it suffices to calculate $D^{\text{spike}}[q]$ for all $M(A)M(B)$ block translations that move any spike in train A into coincidence with any spike in train B.

Another kind of elementary step is motivated by the notion that motifs of spikes (Abeles and Prut 1996) (i.e., a set of three spikes, not necessarily contiguous, with specific interspike intervals) are meaningful. To capture this notion, one adds an elementary step that allows subsets of noncontiguous spikes to be moved as a block. One could also combine the rules of $D^{\text{spike}}[q]$ and $D^{\text{interval}}[q]$, perhaps associated with different costs. Unfortunately, it is unclear how to incorporate these generalizations into a polynomial-time algorithm, since in either case, there is the possibility that a minimal-cost sequence of transformations will require several movements of individual spike-train components.

Further Generalizations For the multineuronal metric $D^{\text{spike}}[q,k]$, the cost to change the label (neuron) associated with an event, k, is independent of the label itself. This restriction is inessential to the definition of the metric; the cost to change a label from l to l' can be an arbitrary symmetric function $k(l,l')$ of the labels. In its full generality, this extension incurs a substantial increase in the number of parameters, but this parameter explosion can be mitigated by a priori hypotheses on the form of $k(l,l')$. For each choice of values of $k(l,l')$, the time required for calculation of the metric does not increase, but the storage requirements of the "parallel" algorithm (see below) increase dramatically.

7.2.2.6 Algorithms

Identification of a minimal-cost path (7.1) might at first seem to be a daunting task. However, because the cost-based metrics $D^{\text{spike}}[q]$ and $D^{\text{interval}}[q]$ are similar to the edit-length distances used for comparison of genetic sequences, the efficient dynamic programming algorithms developed for sequence comparison (Sellers 1974; Needleman and Wunsch 1970) are readily adapted to calculate spike train metrics (Victor and Purpura 1997, 1996). For all of these algorithms, the number of computations required to calculate a distance between two responses is bounded by a

polynomial function of the number of spikes. This efficiency (in contrast to the very laborious search of all possible sequences of elementary steps) is important in that it makes the application of metric-space methods practical.

The algorithms described below have been implemented in public-domain software, available at http: http://neuroanalysis.org/toolkit/ or http://www.apst.spiketrain-analysis.org/ and described in (Goldberg et al. 2009).

The Basic Dynamic Programming Algorithm To describe this algorithm, we use a formulation (Victor et al. 2007) that, while perhaps more elaborate, provides an easier path to generalization. We focus on $D^{\text{spike}}[q]$, but the same algorithmic structure is applicable to $D^{\text{interval}}[q]$. A simpler formulation, but one that is less-readily generalized, can be found in (Victor and Purpura 1997, 1996).

We begin by deducing certain properties that any minimal-cost path (7.1) must have. First, we observe that we can always reorganize a minimal-cost path so that the first steps consist of deleting spikes that occur in train A but not in train B, the intermediate steps consist of moving some spikes that occur at times $A_{a_1}, A_{a_2}, \ldots, A_{a_R}$ in train A to "linked" spikes that occur at times $B_{b_1}, B_{b_2}, \ldots, B_{b_R}$ in train B, and the final steps consist of inserting spikes that occur in train B but not in train A (Fig. 7.1). This reorganization is always possible because any movement of a spike before it is deleted (or insertion of a spike followed by moving it) is inefficient—the spike should simply be deleted before it is moved, or inserted where it is needed. Moreover, the cost of moving a spike is independent of the times of moving any other spike, so performing the deletions first and the insertions last does not change the costs of the moves.

Therefore, a minimal-cost path can be summarized as an "alignment": a designated subset of R spikes in train A and a designated subset of R spikes in train B to which they are linked by R elementary steps that move a single spike. Given this alignment, the distance between the two spike trains can be expressed as

$$d(A,B) = \big(M(A) - R\big) + q \sum_{j=1}^{R} |A_{a_j} - B_{b_j}| + \big(M(B) - R\big), \tag{7.7}$$

where the first term is the cost of the deletions, the final term is the cost of the insertions, and the middle term is the cost of the moves, namely, the total link length multiplied by the cost, q.

Since (by hypothesis) the total cost of these steps is minimal, then the total link length $\sum_{j=1}^{R} |A_{a_j} - B_{b_j}|$ must also be minimal, among all possible links between R pairs of spikes. This in turn implies that the two sequences of spike times $A_{a_1}, A_{a_2}, \ldots, A_{a_R}$ and $B_{b_1}, B_{b_2}, \ldots, B_{b_R}$ must be monotonic. In other words, the links between spikes cannot cross, since if they did, then the total link length could be reduced by uncrossing them.

Let us now assume that we have found the minimal cost path between spike trains A and B. We now focus on the final spike in the two trains, at times $A_{M(A)}$ and $B_{M(B)}$. We use $X^{(k)}$ to denote a spike train consisting of the first k spikes in X. For example, $A^{(M(A)-1)}$ is the spike train A with the last spike deleted. If

the final spikes are connected by a link, then removing those spikes and their link must yield a minimal-cost path between the truncated spike trains. That is, if the final spikes are linked, then $d(A, B) = d(A^{(M(A)-1)}, B^{(M(B)-1)}) + q|A_{a_{M(A)}} - B_{b_{M(B)}}|$. Alternatively, if the final spikes are not connected to each other by a link, it must be that at least one of the spikes is unlinked, since otherwise, their links would cross. If the final spike in A is unlinked, then $d(A, B) = d(A^{(M(A)-1)}, B) + 1$; if the final spike in B is unlinked, then $d(A, B) = d(A, B^{(M(B)-1)}) + 1$. Since at least one of these possibilities must hold, we have the relationship

$$\begin{aligned} d(A, B) = \min\big\{ & d\big(A^{(M(A)-1)}, B^{(M(B)-1)}\big) + q|A_{a_{M(A)}} - B_{b_{M(B)}}|, \\ & d\big(A^{(M(A)-1)}, B\big) + 1, d\big(A, B^{(M(B)-1)}\big) + 1\big\}. \end{aligned} \tag{7.8}$$

That is, we have expressed the distance between two spike trains in terms of the distances between spike trains that have been shortened by one spike.

It is far more efficient to implement (7.8) as a forward iteration than as a recursion. To make this explicit, we note that (7.8) holds for all spike trains A and B, so it may be recast as

$$\begin{aligned} d\big(A^{(\alpha)}, B^{(\beta)}\big) = \min\big\{ & d\big(A^{(\alpha-1)}, B^{(\beta-1)}\big) + q|A_\alpha - B_\beta|, \\ & d\big(A^{(\alpha-1)}, B^{(\beta)}\big) + 1, d\big(A^{(\alpha)}, B^{(\beta-1)}\big) + 1\big\}. \end{aligned} \tag{7.9}$$

Equation (7.9) amounts to an iterative procedure for calculating $D^{\text{spike}}[q](A, B) = d(A^{(M(A))}, B^{(M(B))})$ by building up each spike train one spike at a time. The iteration is initialized (at $\alpha = 0$ or $\beta = 0$ when one spike train is empty) by noting that $d(A^{(\alpha)}, B^{(0)}) = \alpha$ (since all spikes must be deleted) and $d(A^{(0)}, B^{(\beta)}) = \beta$ (since all spikes must be inserted). At each stage of the algorithm, one spike train or the other is elongated by a single spike, and the resulting distance is the minimum of three simply calculated quantities. Thus, the computational burden is proportional to the product of the number of spikes in each spike train, i.e., $O(M^2)$, where M is the typical number of spikes in each train to be compared.

We also mention that fast algorithms for identifying minimal-cost alignments can also be based on weighted bipartite matching (Dubbs et al. 2009), and these are applicable not only to $D^{\text{spike}}[q]$ but also to variants that may be more suitable for Euclidean embedding and multidimensional scaling.

Extensions of the Dynamic Programming Algorithm $D^{\text{spike}}[q](A, B)$ can be calculated for multiple values of the cost parameter q in an efficient, parallel fashion (Victor et al. 2007). The key observation is that the dependence of $D^{\text{spike}}[q](A, B)$ on q is piecewise linear. The breakpoints occur when the cost of a link is equal to 2 (so that removing it is equal to the cost of deleting and inserting its paired spikes). The positions of the breakpoints can be calculated by a dynamic programming algorithm whose computational burden is $O(M^3)$.

The above algorithms can also be applied to the multineuronal spike-time metric $D^{\text{spike}}[q, k]$. The main hurdle is that in a minimal-cost alignment of multineuronal spike trains, links between paired spikes may cross if their labels are different (Fig. 7.2A). Thus, it would appear that the iterative process of (7.9) would have to

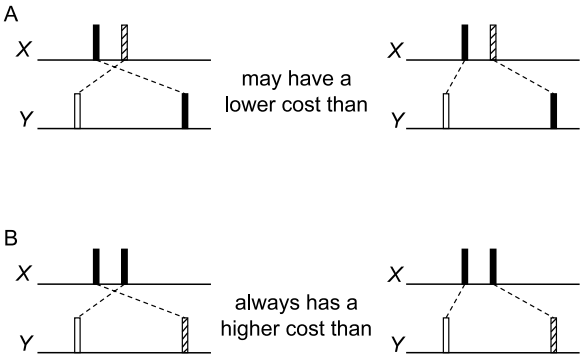

Fig. 7.2 The noncrossing rule for alignment of multineuronal responses. Spikes from different neurons (i.e., with different labels) are diagrammed as *filled*, *open*, and *hashed*. Panel **A** shows a configuration in which an alignment with crossed links may have a lower cost than an uncrossed alignment. Although the total link length is longer in the crossed alignment, one linked pair of spikes has the same label, allowing it to have a lower cost. In configuration **B**, both spikes in one of the spike trains (here, train X) have the same label. Because of this, the minimal-cost alignment must be uncrossed, since uncrossing the links does not influence whether the links connect spikes of the same label or of different labels

be implemented in nested loops for each of the L labels on each of the two spike trains. This would yield an algorithm with a computational burden of $O(M^{2L})$.

However, a closer look (Aronov 2003) yields an algorithm with a computational burden of only $O(M^{L+1})$. The reason for this reduction is the observation (Aronov 2003) that for certain configurations of labeled spikes, the noncrossing rule still holds (Fig. 7.2B). In particular, noncrossing can be guaranteed if both spikes in one train have the same label, since in this case, uncrossing the links does not affect whether any labels need to be changed and thus, cannot incur an additional cost k. To exploit this observation, the two spike trains are treated asymmetrically. That is, the iterative process of (7.9) is applied to one spike train independently of label and to the other spike train on a label-by-label basis. Details of the algorithm are provided in Aronov (2003).

7.2.3 Spike-Train Metrics Based on Vector-Space Embeddings

Above, we began with the notion of spike trains as event sequences and defined distances based on the "cost" of transformations between these sequences. Here, we describe another class of metrics that are based on embedding the spike trains into a vector space. Typically (van Rossum 2001; Richmond and Optican 1987), the embedding is linear, so that the resulting metric respects linearity. But this is not a prerequisite for this approach, and, more recently, Houghton (2009) has developed metrics based on nonlinear embeddings.

There are two main reasons for considering vector-space embeddings. First, the embedding process (i.e., the transformation of an event sequence $t_1, \ldots, t_{M(A)}$ into

a function of time $A(t)$) can be considered as an abstraction of what happens at a synapse: a sequence of action potentials results in a pattern of transmitter release, and this, in turn, has an effect $A(t)$ on the postsynaptic conductance. Second, in many cases, these metrics are simpler to calculate than the cost-based metrics.

7.2.3.1 Single-Neuron Metrics Based on Vector Space Embeddings

We use a formulation that encompasses the metrics considered by van Rossum (2001) and Houghton and Sen (2008). The first step is to represent a spike train A as a function of continuous time, $A(t)$. To do this, the event sequence $t_1, \ldots, t_{M(A)}$ is taken to be a sum of delta-functions

$$\delta_A(t) = \sum_{j=1}^{M(A)} \delta(t - t_j), \tag{7.10}$$

and the resulting sum $\delta_A(t)$ is convolved with a kernel function $K(t)$. This is equivalent to replacing each spike by the kernel waveshape and adding the waveshapes when they overlap. More formally,

$$A(t) = (\delta_A * K)(t) = \int_{-\infty}^{\infty} \delta_A(\tau) K(t - \tau)\, d\tau = \sum_{j=1}^{M(A)} K(t - t_j). \tag{7.11}$$

With the above embedding, any vector-space distance can be used to define a distance between the two spike trains, $d(A, B)$; in particular, the L^p-norm yields the distance

$$d(A, B) = \left(\int_{-\infty}^{\infty} \left| A(t) - B(t) \right|^p dt \right)^{1/p}. \tag{7.12}$$

For all metrics defined in this fashion, the distance between two spike trains that differ by inserting or deleting one spike is given by the L^p-norm of the kernel K, i.e., $(\int_{-\infty}^{\infty} |K(t)|^p\, dt)^{1/p}$.

van Rossum (2001) focuses on the L^2-distance (the Euclidean distance) and the exponential kernel

$$K^{VR}(t_c; t) = \begin{cases} \frac{1}{\sqrt{t_c}} e^{-t/t_c}, & t \geq 0, \\ 0, & t < 0. \end{cases} \tag{7.13}$$

We denote the metric that results from combining (7.11), (7.12) and (7.13) by $D^{VR}[t_c](A, B)$, where t_c is a parameter that controls the sensitivity of the metric to temporal detail. The kernel is normalized so that for two spike trains that differ by the addition or deletion of a single spike, $(D^{VR}[t_c](A, B))^2 = 1/2$, regardless of t_c.

$D^{VR}[t_c](A, B)$ is designed to be similar to $D^{\text{spike}}[q](A, B)$, with $1/t_c$ playing a role analogous to that of q in the cost-based metrics. That is, small values of $1/t_c$ compare spike trains based on the number of spikes, while large values of $1/t_c$ compare spike trains based on precise times of occurrence. To see this, we consider

several simple cases. For two spike trains that have only a single spike for which the times differ by an amount ΔT, the distance is given by $(D^{VR}[t_c](A, B))^2 = 1 - e^{-|\Delta T|/t_c}$, a quantity that is 0 at $\Delta T = 0$ and, for small ΔT, increases proportionally to $|\Delta T|/t_c$. In the limit as $t_c \to \infty$, $2(D^{VR}[t_c](A, B))^2$ is the square of difference in the number of spikes in the two trains (similar to the count-dominated behavior of $D^{\text{spike}}[q](A, B)$ as $q \to 0$), while in the limit as $t_c \to 0$, $2(D^{VR}[t_c](A, B))^2$ is the number of spikes in the two spike trains that occur at distinct times (matching the timing-dominated behavior of $D^{\text{spike}}[q](A, B)$ as $q \to \infty$).

Houghton and Sen (2008) explicitly consider L^1-norms and a kernel that makes the correspondence to $D^{\text{spike}}[q](A, B)$ even closer. For their kernel choice

$$K^{HS}(q; t) = \begin{cases} q/2, & 0 \le t < 2/q, \\ 0, & \text{otherwise}, \end{cases} \tag{7.14}$$

and $p = 1$ in (7.12), the correspondence of the distance $D^{HS}[q](A, B)$ to $D^{\text{spike}}[q](A, B)$ is exact when one spike train has no spikes, when both spike trains have one spike, when all spikes within each train are widely separated, when $q \to 0$, or when $q \to \infty$. However, like $D^{VR}[t_c](A, B)$, the correspondence of $D^{HS}[q](A, B)$ to $D^{\text{spike}}[q](A, B)$ is typically not exact when there are many spikes in each train, and the spikes occur with separations less than $2/q$ (or $O(t_c)$). The main qualitative difference is that $D^{VR}[t_c](A, B)$ and $D^{HS}[q](A, B)$ are Euclidean or can be converted to a Euclidean distance by a power-law transformation, while $D^{\text{spike}}[q](A, B)$ cannot (Aronov and Victor 2004).

In addition to the exponential (van Rossum 2001) and boxcar (Houghton and Sen 2008) kernel, other kernel shapes, such as a Gaussian, have been used in this context (Schreiber et al. 2004). Metrics on spike trains can also be derived by binning the spike times (Lim and Capranica 1994). Here, the temporal function $A(t)$ used in (7.12) is the mean firing rate in each bin. These metrics can be viewed as approximate versions of the metrics considered above, in which the convolution integral (7.11) is replaced by a discrete sum, and the kernel (for a bin width $2/q$) is given by (7.14).

Finally, we point out that the transformation from the sequence of delta-functions (7.10) to the temporal function $A(t)$ (7.11) need not be linear. Indeed, since the latter can be considered to represent the postsynaptic effect of an impulse train, it is reasonable to consider nonlinear transformations that caricature biophysical mechanisms, such as an incomplete return to resting potential when spikes are in rapid sequence. Houghton (Houghton and Sen 2008) has recently done this, implementing a simple model of short-term synaptic adaptation (Sen et al. 1996).

7.2.3.2 Multineuronal Metrics Based on Vector-Space Embeddings

Houghton and Sen (2008) have extended the above strategy to multineuronal metrics. The construction begins by extending, to multiple neurons, the representation (10) of a single neuron's spike train A as a sequence of delta-functions $\delta_A(t)$. To do this, they augment the delta-function corresponding to each spike by a unit vector $\vec{c}_l$;

the direction of this unit vector represents the label l (neuron of origin) of that spike. However, rather than simply assigning a separate orthogonal unit vector $\vec{e}_l$ as the direction for each of the L labels l, they allow the directions to be nonorthogonal, i.e., $\vec{c}_l = \sum c_{l,r}\vec{e}_r$. As we will see below, orthogonal labels correspond to labeled lines (the $k = 2$ extreme for $D^{\text{spike}}[q, k]$), collinear labels correspond to a summed population code (the $k = 0$ extreme for $D^{\text{spike}}[q, k]$), and intermediate choices correspond to intermediate behaviors.

With these preliminaries, a multineuronal spike train (with L neurons) with events at times t_j and associated with labels l_j is represented by an L-dimensional array of sums of scaled delta-functions,

$$\vec{\delta}_A(t) = \sum_{j=1}^{M(A)} \vec{c}_{l_j}\delta(t - t_j). \tag{7.15}$$

In coordinates, $\vec{\delta}_A(t) = (\delta_{A,1}(t), \ldots, \delta_{A,L}(t))$, where each scaled delta-function $\delta_{A,r}(t)$ is given by

$$\delta_{A,r}(t) = \sum_{j=1}^{M(A)} c_{l_j,r}\delta(t - t_j). \tag{7.16}$$

As in the single-neuron metrics, temporal factors are taken into account by convolving the delta-function array (7.15) by a kernel, yielding $\vec{A}(t) = (\vec{\delta}_A * K)(t)$, with the convolution carried out separately for each coordinate (7.11). (In principle, different kernels can be assigned to each coordinate, or different kernels can be assigned to each neuron prior to mixing them in (7.15).) Finally, the distance between two multineuronal spike trains A and B is the L^p-norm between their associated temporal functions $\vec{A}(t)$ and $\vec{B}(t)$, namely,

$$d(A, B) = \left(\int_{-\infty}^{\infty} \sum_{l=1}^{L} \left| A_l(t) - B_l(t) \right|^p dt \right)^{1/p}, \tag{7.17}$$

where $A_l(t)$ and $B_l(t)$ are, respectively, the lth component of $\vec{A}(t)$ and $\vec{B}(t)$. We note that (7.17) is equivalent to

$$d^p(A, B) = \sum_{l=1}^{L} d^p(A_l, B_l). \tag{7.18}$$

As an example of this construction, Houghton and Sen (2008) consider the two-neuron case and assign the unit vector $\vec{c}_1 = \binom{1}{0}$ to the first neuron, and $\vec{c}_2 = \binom{\cos\theta}{\sin\theta}$ to the second neuron. $\theta = 0$ corresponds to summing the spikes independent of neuron of origin, because spikes of all labels are represented by the same direction ($\vec{c}_1 = \vec{c}_2$). On the other hand, $\theta = \pi/2$ (for which $\vec{c}_1$ and $\vec{c}_2$ are orthogonal) corresponds to a labeled line code. In this case, each $A_l(t)$ and $B_l(t)$ consists of a sequence of delta-functions corresponding to the spikes of an individual neuron, and (7.18) shows that their contributions to the overall squared distance, (7.17), merely add.

7.2.3.3 Computational Considerations

Computation of metrics based on vector-space embeddings is straightforward. One strategy is to approximate the integrals in (7.12) or (7.17) or by a discrete sum; in this situation, the computational burden is $O(ML^2D)$, where M is the typical number of spikes from each neuron, L is the number of neurons, and D is the number of points used for the discretization (Houghton and Sen 2008). For the typical norms of $p = 1$ or $p = 2$, these integrals can be reduced to sums over all spikes ($p = 1$) or spike pairs ($p = 2$), resulting in a further saving if the density of spikes is low. Note that the computational burden of these metrics grows much more slowly as a function of the number of neurons (L) than for the cost-based metrics ($O(M^{L+1})$, see above).

7.2.4 Applications

7.2.4.1 Overview

In this section, we discuss several ways in which spike metrics can be used to ask biological questions about neuronal spike trains. (Here, we no longer distinguish between single-unit spike trains and multineuronal "labeled" spike trains; the material in this section applies to both.) We will mention applications to specific neural systems, but we organize our discussion in terms of the nature of the analytical goal. To do this, it is helpful to take a geometrical view, in which each spike train is represented by a point in an abstract space. The most direct application of spike metrics is simply to use the distances to quantify the variability within a set of spike trains. In the geometric view, the statistics of these distances provides a description of the cloud of neural responses.

However, quantifying variability does not probe the manner in which the spike trains relate to perception or behavior. One way to take this next step is to analyze the pairwise distances by standard multivariate techniques, such as multidimensional scaling. Via multidimensional scaling, the individual responses are then embedded into a neural "response space". In the neural response space, the distances between the points representing individual responses approximate the metric distances between the spike trains (as determined by a spike metric), and each kind of stimulus or behavior corresponds to a labeled cloud of points in the response space. One can then examine the positions of the clouds of spike trains that correspond to each stimulus or behavior and ask whether similarities in these domains correspond to similar kinds of neural activity. In other words, one can ask whether the notion of similarity provided by the spike metric provides a neural *representation* of the perceptual or behavioral phenomena.

From the point of view of neural coding, it is crucial to measure the quality of the representation of the perceptual or behavioral domain by the spike trains. Geometrically, this corresponds to asking whether the clouds that correspond to

each stimulus or behavior are distinct, vs. overlapping. The former corresponds to a high-fidelity representation, while the latter corresponds to a noisy one. Because of their nonparametric nature, the tools of information theory are natural to use in this context. If a spike metric leads to a high-fidelity representation, then the temporal features that it captures are candidates for neural codes. Conversely, when it is possible to show that behavioral performance exceeds that of a candidate code (Jacobs et al. 2009), the neural code can be ruled out.

7.2.4.2 Assessment of Variability

Measures of variability for time series depend in general on the time scale of interest; for example, a standard way of describing the variability of a continuous-time series is to compute its power spectrum, which quantifies variability as a function of frequency. This holds for point processes as well. In the present context, this observation means that in assessing the variability of spike trains, it makes sense to use spike metrics that are sensitive to a range of timescales (e.g., $D^{\text{spike}}[q]$ for a range of values of q).

Some examples of the use of spike metric for this purpose include the study of Kreiman et al. (2000), who examined a class of afferents in the weakly electric fish *Eigenmannia* whose spike trains are loosely phase-locked to the periodic discharge of its electrosensory organ. Kreiman et al. (2000) chose to quantify the variability by D^{spike}, since (because of the phase-locking) measures based on spike count or its variance proved ineffective in capturing trial-to-trial variability.

Grewe et al. (2003) used D^{spike}, along with a Euclidean metric, to examine variability in a motion-sensitive visual neuron (H1) of the blowfly. By examining variability in the presence of varying amounts of added sensory noise, they determined that performance of the H1 neuron was limited by its internal noise, rather than photon noise.

In response to full-field random flicker, retinal and lateral geniculate neurons often fire in discrete "firing events" consisting of several spikes, at times that are reproducible across trials. Reinagel and Reid (2002) used D^{spike} to quantify this reproducibility and, moreover, to show that these firing patterns were conserved not only across trials, but also across animals.

Finally, several authors have used D^{spike} to evaluate spike train variability of synthetic data for the purpose of evaluating models and describing their qualitative behavior (Keat et al. 2001; Tiesinga 2004; Banerjee et al. 2008).

7.2.4.3 Construction of Response Spaces

A variety of multivariate techniques can be used to visualize the geometry of the neural "response spaces" that correspond to a spike-train metric. The basic strategy for doing this is multidimensional scaling (MDS) (Kruskal and Wish 1978). In the framework of MDS, given a set of spike trains A and a particular spike metric D,

one seeks an embedding φ from the spike trains A to n-dimensional vectors $\varphi(A) = (\varphi_1(A), \ldots, \varphi_n(A))$ for which the spike metrics $D(A, B)$ are well approximated by the standard Euclidean distances between $\varphi(A)$ and $\varphi(B)$:

$$D(A, B)^2 \approx \left|\varphi(A) - \varphi(B)\right|^2 = \sum_{j=1}^{n} \big(\varphi_j(A) - \varphi_j(B)\big)^2. \tag{7.19}$$

Standard Multidimensional Scaling To determine an embedding φ that satisfies (7.19), we carry out the classical MDS recipe (Kruskal and Wish 1978). The crucial step is to diagonalize the symmetric matrix M_{AB}:

$$M_{AB} = \frac{1}{2}\Bigg(-D(A, B)^2 + \frac{1}{N}\sum_{S} D(A, S)^2 + \frac{1}{N}\sum_{S} D(B, S)^2 - \frac{1}{N^2}\sum_{S,S'} D\big(S, S'\big)^2\Bigg), \tag{7.20}$$

where N is the number of spike trains in the dataset, and the summations range over all spike trains S or pairs of spike trains S and S'. Rows and columns of M are indexed by each of the spike trains in the data set, so the entries in each of its eigenvectors $\phi^{[j]}$ are indexed by the spike trains as well. Thus, we may write

$$M_{AB} = \sum_{j=1}^{N} \lambda_j \phi_A^{[j]} \phi_B^{[j]}, \tag{7.21}$$

where $\phi_A^{[j]}$ denotes the value of the jth eigenvector at spike train A. Provided that the eigenvalues λ_j are all nonnegative, this allows us to write the desired embedding φ as

$$\varphi(A) = \big(\varphi_1(A), \ldots, \varphi_n(A)\big) = \big(\sqrt{\lambda_1}\phi_A^{[1]}, \ldots, \sqrt{\lambda_n}\phi_A^{[n]}\big) \tag{7.22}$$

(see Kruskal and Wish 1978).

The analysis also yields embeddings in lower-dimensional spaces that approximate the metric $D(A, B)$. To find an approximate embedding, we simply number the eigenvalues in descending order and only use the first $K < N$ of them for coordinates in (7.22). These approximate embeddings (e.g., with $K = 2$ or $K = 3$) can be used to visualize the relative similarities of a set of spike trains, as determined by $D(A, B)$.

Examples The above procedures have been used to characterize the geometry of response spaces in several sensory systems. In the auditory system, Victor and Purpura (1997) applied this approach to the data of Middlebrooks et al. (1994), who recorded responses of single neurons in cat ectosylvian gyrus to sounds that varied in spatial location. The original data showed that these neurons' responses could distinguish the origin of the sound source in a panoramic (360°) fashion and that spike timing at a resolution of ca. 4 ms was critical for this (Middlebrooks et al. 1994; Furukawa and Middlebrooks 2002). The reanalysis (Victor and Purpura 1997),

which applied MDS to the distances yielded by D^{spike} and D^{interval}, added to this picture by showing that the geometry of these responses (i.e., their relative distances) recovered the circular geometry of the stimulus space.

In the visual system, Aronov et al. (2003) used $D^{\text{spike}}[q, k]$ to characterize the representation of spatial phase across pairs of neurons in primary visual cortex. The geometry of the stimulus set (the circle of spatial phase) could be recovered by applying MDS to the response similarities, as quantified by $D^{\text{spike}}[q, k]$. Moreover, response spaces became more nearly circular for nonzero values of k, indicating that within a local cluster, the neuron of origin of a spike, as well as its timing, contributes to the representation of spatial phase.

Aronov et al. (2003) also introduced a technique that can be used to interpret the axes obtained in the MDS procedure. Essentially, for each dimension m in (7.22), they regressed the coordinate value m of each response, $\phi_A^{[m]}$, against a binned post-stimulus histogram $A(t)$. By showing that $\phi_A^{[m]}$ could be approximated by a linear weighting of the response time course, i.e.,

$$\phi_A^{[m]} \approx \int A(t) L_m(t)\, dt, \tag{7.23}$$

they identified "temporal profiles" $L_m(t)$ that could be associated with each dimension of the embedding.

Recently, Di Lorenzo et al. (2009) applied this procedure to single-neuron recordings in the nucleus tractus solitarius of the rat, during stimulation by each of the four primary tastants (salt, sweet, sour, bitter) and their binary mixtures. Neurons that were broadly tuned in terms of spike count were nevertheless able to distinguish among these tastants via temporal coding. The neural response space, as constructed by the above procedure, recapitulated the tetrahedral geometry anticipated from gustatory psychophysics (Erickson 1984).

Implications of Non-Euclidean Nature of Spike Metrics The above approach (standard MDS) seeks to embed spike trains into a Euclidean vector space in a distance-preserving manner. Thus, the embedding can only be exact if the spike metric itself is Euclidean, which is not the case (Aronov and Victor 2004). The above procedure fails to be exact for non-Euclidean metrics because some of the eigenvalues of M are negative. Consequently, the bilinear form (7.21) is not positive definite (and hence, not an inner product), and the coordinates (7.22) are not real. Nevertheless, since the first several eigenvalues of M are typically positive, the above procedure finds an Euclidean approximation of the spike metric that suffices for visualization of the spike trains. Other approaches to deal with this problem are described below.

Relationship to the "Kernel Trick" and van Rossum-Type Metrics The "kernel trick" is applicable to a situation in which one wishes to classify a set of objects (here, the spike trains) which either do not have a linear structure, or in which the important features are nonlinearly related to the objects. To do this, one introduces an embedding ψ, which maps the objects into a vector space. Then, the original objects can be classified via linear classifiers in the feature space.

This is a generalization of the approach of van Rossum (2001), Richmond and Optican (1987), in which spike trains are embedded into an infinite-dimensional vector space (functions of time) by convolving them with a smoothing kernel (e.g., (7.11)). Suppose that we have a set of spike trains A_k that correspond to temporal functions $A_k(t)$ via this transformation. It is natural to ask what would happen if we apply the above MDS procedure to their L^2-distances, (7.12). Since this is manifestly a Euclidean distance, an exact embedding must be possible. However, we cannot recover the original vector space of functions of time, because the latter is infinite-dimensional, but the MDS procedure necessarily yields a finite-dimensional embedding. Instead, we would find that the successive eigenvectors of M correspond to the successive principal components of the set of temporal functions $A_k(t)$ (minus their overall mean). That is, although the MDS procedure does not recover the smoothing kernel that defines the distance, it finds the vector space that contains all of the temporal functions that result from applying this kernel.

As a converse, we mention that these relationships imply that the non-Euclidean spike metrics cannot be calculated via a "kernel trick", since if this were possible, then the metrics could be recovered *exactly* by an MDS procedure (see below).

Nonlinear Scaling Above we have focused on finding an embedding φ that approximates a spike metric by a Euclidean distance (7.19). However, since spike metrics typically do not correspond to any Euclidean distance (Aronov and Victor 2004), there is no guarantee that embeddings that satisfy (7.19) can be found. This motivates several other strategies for response space construction, which we now briefly mention. One such strategy is to seek embeddings into a curved space. That is, the Euclidean distance on the right-hand side of (7.19) is replaced by the geodesic distance between $\varphi(A)$ and $\varphi(B)$ within a data-defined curved manifold. This can be accomplished by the isomap (Tenenbaum et al. 2000) and geometric diffusion (Coifman et al. 2005) methods of dimensionality reduction. A second approach is to replace $D(A, B)$ in (7.19) by $f(D(A, B))$ for some monotonic function f. The resulting embedding preserves the rank order of the distances, and, if f is concave, $f(D(A, B))$ remains a metric. $f(D(A, B)) = D(A, B)^\beta$ for $0 < \beta < 1$ is a natural choice for this transformation, since it is scale-invariant. However, while specific choices of β (e.g., $\beta = 1/2$) may be useful to "euclideanize" particular datasets, it is possible to show (Aronov and Victor 2004) that there is no single choice of $\beta > 0$ that universally suffices to transform $D^{\text{spike}}[q, k]$ into a Euclidean distance.

7.2.4.4 Applications to Information-Theoretic Analysis

Finally, we consider applications of spike metrics to information-theoretic analysis of neural data. Information theory (IT) (Shannon and Weaver 1949) (see Cover and Thomas 1991 for a general review) forms a natural framework for the analysis of neural coding (Rieke et al. 1997) (Chap. 13). However, the application of IT to neural systems can be difficult: estimates of mutual information from laboratory data can be biased, imprecise, or both. The estimation problem, whose origin is that the

space of possible responses is undersampled, is compounded for analyses of multineuronal activity, because the dimensionality of the response domain is proportional to the number of neurons.

To motivate some strategies to mitigate this difficulty, we consider a naïve attempt to estimate mutual information from laboratory data. The mutual information between a set of stimuli S and a set of responses R is defined as

$$I(S,R) = -\sum_{s\in S} p(s)\log p(s) + \sum_{r\in R} p(r)\sum_{s\in S} p(s|r)\log p(s|r), \qquad (7.24)$$

where the first term is the entropy of the unconditioned distribution of stimuli, and the second term subtracts the average entropy of the distribution of stimuli, conditioned on observing a particular response $r \in R$. Typically, the distribution of stimuli is known, so the estimation of information rests on the conditional probabilities $p(s|r) = p(s,r)/p(r)$. Below we will make use of a reorganization of (7.24) into a symmetric form,

$$I(S,R) = -\sum_{s\in S} p(s)\log p(s) - \sum_{r\in R} p(r)\log p(r) + \sum_{s\in S, r\in R} p(s,r)\log p(s,r), \qquad (7.25)$$

which states that mutual information is the difference between the entropies of the stimulus and response distributions (the first two terms) and the entropy of their joint distribution (the third term).

Implementing (7.24) or (7.25) directly requires estimates of the joint stimulus-response probabilities $p(s,r)$. The obvious way to obtain these estimates is to count up the number of joint occurrences of each stimulus s and each response r, and divide by the number of events. To do this, the investigator must know how to partition the response domain R into different kinds of responses, $r_1, r_2, \ldots$. The Data Processing Inequality states that if probability estimates from different responses r_1 and r_2 are inadvertently pooled, then the resulting estimate of mutual information will be downwardly biased. Thus, it might appear that the most conservative approach would be to estimate $p(s,r)$ by separately tracking all distinguishable responses. This would then avoid the downward bias due to pooling responses.

The difficulty with this approach is that when the response domain is partitioned very finely, then the number of events that contributes to each estimate of $p(s,r)$ is small, often either 0 or 1. Since entropy and information are nonlinear functions of the underlying probabilities, an overly narrow binning of the response space incurs an *upward* bias (Treves and Panzeri 1995; Miller 1955; Carlton 1969; Victor 2000) in the estimate of mutual information. So the investigator has a dilemma: to avoid a downward bias due to the Data Processing Inequality, the stimulus domain must be sampled as finely as possible, but this leads to an upward bias because of the nonlinearity of the logarithm.

There are two general strategies that can be used to mitigate this dilemma (see Victor 2006 for further details). One strategy is to use a fine partitioning of the response domain but to use advanced estimation techniques (Paninski 2003;

Nemenman et al. 2004) to reduce the upward estimation bias inherent in naïve estimators. This strategy makes minimal assumptions about the nature of the neural code but still has large data requirements.

The second strategy, which is where spike metrics are relevant, is based on the observation that pooling distinct responses r_1 and r_2 only reduce the estimate of mutual information when the responses are associated with distinct a posteriori probabilities, $p(s|r_1)$ and $p(s|r_2)$. Conversely, if two responses r_1 and r_2 have the same "meaning" (i.e., lead to identical a posteriori estimates of what the stimulus was), then probability estimates can be pooled without incurring an upward bias.

This places the focus on determining which neural responses have the same meaning. Each spike metric, in essence, is a formal hypothesis about exactly this: two neural responses are hypothesized to have the same meaning if their distance is zero, and increasing distances indicates progressively different meanings.

Motivated by this observation, a spike metric can be used to provide an estimate of mutual information that is relatively unaffected by the undersampling problem but strongly sensitive to whether, in fact, the hypothetical distance is close to the correct one. A strategy for doing this for a discrete stimulus set is detailed in Victor and Purpura (1997); we summarize it here.

The first step in the analysis is to calculate all of the pairwise distances between the responses. Then, response clusters are created, with one cluster for each stimulus s_α. The clusters are formed based on the experimenter's knowledge of which stimulus elicited each response. In particular, a response r is placed into a cluster β if the average distance between r and all the responses elicited by the stimulus s_β is smaller than the average distance between r and the responses elicited by any other stimulus. (The averaging process is carried out following a negative-power law transformation, to emphasize near-matches. The exponent, typically -2, is the parameter z of Victor and Purpura 1997.) The result of applying this clustering to all responses is a table $N(\alpha, \beta)$ which counts the number of times that a response to the stimulus s_α is assigned to cluster β. From this table mutual information is estimated as

$$\begin{aligned} I(S, R) \approx & -\sum_{\alpha} p(\alpha, \bullet) \log p(\alpha, \bullet) - \sum_{\beta} p(\bullet, \beta) \log p(\bullet, \beta) \\ & + \sum_{\alpha, \beta} p(\alpha, \beta) \log p(\alpha, \beta), \end{aligned} \tag{7.26}$$

where $p(\alpha, \beta) = \frac{N(\alpha, \beta)}{\sum_{\alpha, \beta} N(\alpha, \beta)}$, $p(\alpha, \bullet) = \sum_\beta p(\alpha, \beta)$, and $p(\bullet, \beta) = \sum_\alpha p(\alpha, \beta)$.

The reason that (7.26) represents a useful approach to the undersampling strategy is that $p(s, r)$ has been replaced by $p(\alpha, \beta)$. The former requires keeping track, separately, of the number of occurrences of each response, while the latter only requires keeping track of the number of occurrences of responses in the same cluster. That is, we have dealt with the undersampling problem by introducing a clustering procedure: we have lumped responses together (into the same cluster β) if they are

similar to responses that are known to be elicited by the stimulus s_β.[5] The Data Processing Inequality guarantees that this will result in an underestimate of information (assuming that the clusters are sufficiently well sampled), but the key point is that the extent of this underestimate is determined by whether or not the metric-based clustering properly classifies spike trains that have the same meaning.

In sum, estimates of information based on (7.26) reflect the extent to which the chosen spike metric accurately captures meaningful differences between spike trains. That is, the dependence of the information estimates of (7.26) on the choice of spike metric (e.g., for $D^{\text{spike}}[q, k]$, the temporal precision parameter q and the spike label parameter k) characterizes the informative features of the neural response. In particular, if $q_{\max}$ is the value of q for which estimates of information via (7.26) based on $D^{\text{spike}}[q]$ (or $D^{\text{spike}}[q, k]$) achieves its highest value, then $1/q_{\max}$ can be viewed as the "informative temporal precision" of a spike, namely, the amount of time by which moving a spike has the same impact on the meaning of the spike train as deleting the spike altogether.

Examples Victor and Purpura (1996) used this approach extensively to characterize how individual neurons in primary (V1) and secondary (V2 and V3) visual cortices carried information about multiple stimulus attributes, such as contrast, orientation, and texture. They found that greater information was recovered from clustering based on $D^{\text{spike}}[q]$ than from that based on $D^{\text{interval}}[q]$. Moreover, using $D^{\text{spike}}[q]$, they found that the informative precision of a neuron's response depended on the stimulus attribute—with the highest precision for contrast (10 to 30 ms) and lowest precision for texture ($\sim$100 ms). Thus, individual spike trains can be considered to carry information about several stimulus attributes in a temporally multiplexed fashion.

Estimates of informative temporal precision obtained by the above approach are necessarily averages across the entire response. This is underscored by the study of Reich et al. (2000), who used D^{spike} to show that most of the information about contrast could be extracted from the latency of the first spike and that, if only the first spike is considered, the informative temporal precision can be as low as 2 to 5 ms.

Samonds and Bonds (2004) and Samonds et al. (2003), examined signaling of orientation in cat primary visual cortex with D^{spike} and D^{interval}, including multineuronal extensions of these measures. Their analysis showed that large orientation differences could be signaled by firing rate but that orientation differences of 10° or less were signaled by the temporal fine structure (2 to 10 ms) of spike times and spike intervals.

Additional information-theoretic applications of spike metrics in the visual system include the work of Mechler et al. (1998), who showed that temporal structure

[5] The "hard clustering" used in this procedure might also lead to an underestimate of information, in comparison to a procedure that gives soft, or probabilistic, assignments to each response. Recently, I. Nelken (2009) has proposed a procedure that circumvents this difficulty, by applying the "binless embedding" method (Victor 2002) to the distances calculated by spike metrics.

played a much larger role in the coding of edges (square-wave gratings) than of smooth variations (sine gratings), and the study of Chichilnisky and Rieke (2005), who found that for near-threshold responses in retinal ganglion cells, the informative temporal precision was approximately 100 ms.

There have also been a number of applications of this approach in other sensory systems (e.g., Machens et al. 2001 in the grasshopper auditory system, Di Lorenzo and Victor 2003, 2007 in the gustatory system). Particularly noteworthy are the combined behavioral and neurophysiological experiments of Laurent and colleagues (MacLeod et al. 1998) in the olfactory system of the locust. They used D^{spike} to show that increasing timing jitter of spike trains in projection neurons leads to loss of behavioral discrimination of similar odors, though coarse odor discrimination remains intact (Stopfer et al. 1997). These elegant experiments demonstrate the functional relevance of precise spike timing for fine sensory discriminations.

Several of the above studies (e.g., Victor and Purpura 1996; Di Lorenzo and Victor 2003, 2007) made use of a surrogate data technique, "exchange resampling". The authors reanalyzed surrogate data sets in which individual spikes were swapped across pairs of responses to the same stimulus. These surrogate data sets, by construction, have the same number of spikes in each trial as the original data and have the same time-varying firing rates as the original data, but spike correlations within trials are destroyed. Information estimates from these surrogate data sets were somewhat less than the estimates obtained from the original data, indicating that the pattern of spikes in individual trials, and not just the time course of firing probability, was informative.

It is important to emphasize that these methods quantify the amount of information that is available in the neural response and the spike train features that carry this information. In order to claim that the information is actually used, the analyses must eventually be coupled to behavior (MacLeod et al. 1998; Jacobs et al. 2009).

7.3 Conclusion

Spike metrics provide a formal structure for analyzing neural activity as event sequences and provide a means to assess variability, to visualize patterns of response similarity, and to estimate information-theoretic quantities. While many simple and useful spike metrics can be calculated by efficient dynamic programming algorithms, extensions of the approach to additional metrics present a range of algorithmic challenges.

Acknowledgements This work is supported in part by National Eye Institute Grant 1RO1 EY9314 to J. Victor and National Institute of Mental Health Grant 1RO1 MH68012 to Daniel Gardner. We would like to thank Sebastien Louis for his thorough reading of a draft and his thoughtful and helpful comments.

References

Abbott LF (2000) Integrating with action potentials. Neuron 26:3–4
Abeles M (1982) Role of the cortical neuron: integrator or coincidence detector?. Isr J Med Sci 18:83–92
Abeles M, Prut Y (1996) Spatio-temporal firing patterns in the frontal cortex of behaving monkeys. J Physiol Paris 90:249–250
Aronov D (2003) Fast algorithm for the metric-space analysis of simultaneous responses of multiple single neurons. J Neurosci Methods 124:175–179
Aronov D, Victor JD (2004) Non-Euclidean properties of spike train metric spaces. Phys Rev E Stat Nonlin Soft Matter Phys 69:061905
Aronov D, Reich DS, Mechler F, Victor JD (2003) Neural coding of spatial phase in V1 of the macaque monkey. J Neurophysiol 89:3304–3327
Banerjee A, Series P, Pouget A (2008) Dynamical constraints on using precise spike timing to compute in recurrent cortical networks. Neural Comput 20:974–993
Carlton AG (1969) On the bias of information estimates. Psychol Bull 71:108–109
Chichilnisky EJ, Rieke F (2005) Detection sensitivity and temporal resolution of visual signals near absolute threshold in the salamander retina. J Neurosci 25:318–330
Coifman RR, Lafon S, Lee AB, Maggioni M, Nadler B, Warner F, Zucker SW (2005) Geometric diffusions as a tool for harmonic analysis and structure definition of data: diffusion maps. Proc Natl Acad Sci USA 102:7426–7431
Cover TM, Thomas JA (1991) Elements of information theory, Schilling DL (ed). Wiley, New York
Dan Y, Poo MM (2004) Spike timing-dependent plasticity of neural circuits. Neuron 44:23–30
Di Lorenzo PM, Victor JD (2003) Taste response variability and temporal coding in the nucleus of the solitary tract of the rat. J Neurophysiol 90:1418–1431
Di Lorenzo PM, Victor JD (2007) Neural coding mechanisms for flow rate in taste-responsive cells in the nucleus of the solitary tract of the rat. J Neurophysiol 97:1857–1861
Di Lorenzo PM, Chen J-Y, Victor JD (2009) Quality time: representation of a multidimensional sensory domain through temporal coding. J Neurosci 29(29):9227–9238
Dubbs AJ, Seiler BA, Magnasco MO (2009) A fast Lp spike alignment metric. arXiv:0907.3137v2
Egger V, Feldmeyer D, Sakmann B (1999) Coincidence detection and changes of synaptic efficacy in spiny stellate neurons in rat barrel cortex. Nat Neurosci 2:1098–1105
Erickson RP (1984) Ohrwall, Henning and von Skramlik; the foundations of the four primary positions in taste. Neurosci Biobehav Rev 8:105–127
Furukawa S, Middlebrooks JC (2002) Cortical representation of auditory space: information-bearing features of spike patterns. J Neurophysiol 87:1749–1762
Gaal SA (1964) Point set topology. Academic Press, New York
Goldberg DH, Victor JD, Gardner EP, Gardner D (2009) Spike train analysis toolkit: enabling wider application of information-theoretic techniques to neurophysiology. Neuroinformatics 7(3):165–178
Grewe J, Kretzberg J, Warzecha AK, Egelhaaf M (2003) Impact of photon noise on the reliability of a motion-sensitive neuron in the fly's visual system. J Neurosci 23:10776–10783
Hopfield JJ (1995) Pattern recognition computation using action potential timing for stimulus representation. Nature 376:33–36
Houghton C. (2009) Studying spike trains using a van Rossum metric with a synapse-like filter. J Computat Neurosci 26:149–155
Houghton C, Sen K (2008) A new multineuron spike train metric. Neural Comput 20(6) 1495–1511
Jacobs AL, Fridman G, Douglas RM, Alam NM, Latham PE, Prusky GT, Nirenberg S (2009) Ruling out and ruling in neural codes. Proc Natl Acad Sci USA 106:5936–5941
Keat J, Reinagel P, Reid RC, Meister M (2001) Predicting every spike: a model for the responses of visual neurons. Neuron 30:803–817

Kreiman G, Krahe R, Metzner W, Koch C, Gabbiani F (2000) Robustness and variability of neuronal coding by amplitude-sensitive afferents in the weakly electric fish eigenmannia. J Neurophysiol 84:189–204
Kruskal JB, Wish M (1978) Multidimensional scaling. Sage, Beverly Hills
Kuba H, Yamada R, Fukui I, Ohmori H (2005) Tonotopic specialization of auditory coincidence detection in nucleus laminaris of the chick. J Neurosci 25:1924–1934
Lim D, Capranica RR (1994) Measurement of temporal regularity of spike train responses in auditory nerve fibers of the green treefrog. J Neurosci Methods 52:203–213
Machens C, Prinz P, Stemmler M, Ronacher B, Herz A (2001) Discrimination of behaviorally relevant signals by auditory receptor neurons. Neurocomputing 38:263–268
MacLeod K, Backer A, Laurent G (1998) Who reads temporal information contained across synchronized and oscillatory spike trains?. Nature 395:693–698
Maloney LT, Yang JN (2003) Maximum likelihood difference scaling. J Vision 3:5. doi:10.1167/3.8.5
Markram H, Lubke J, Frotscher M, Sakmann B (1997) Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. Science 275:213–215
Mechler F, Victor JD, Purpura KP, Shapley R (1998) Robust temporal coding of contrast by V1 neurons for transient but not for steady-state stimuli. J Neurosci 18:6583–6598
Middlebrooks JC, Clock AE, Xu L, Green DM (1994) A panoramic code for sound location by cortical neurons. Science 264:842–844
Miller GA (1955) Note on the bias on information estimates. Information Theory in Psychology: Problems and Methods II-B:95–100
Needleman SB, Wunsch CD (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48:443–453
Nelken I (2009) Personal communication
Nemenman I, Bialek W, de Ruyter van Steveninck R (2004) Entropy and information in neural spike trains: progress on the sampling problem. Phys Rev E Stat Nonlin Soft Matter Phys 69:056111
Paninski L (2003) Estimation of entropy and mutual information. Neural Comput 15:1191
Reich D, Mechler F, Victor J (2000) Temporal coding of contrast in primary visual cortex: when, what, and why?. J Neurophysiol 85:1039–1050
Reinagel P, Reid RC (2002) Precise firing events are conserved across neurons. J Neurosci 22:6837–6841
Richmond BJ, Optican LM (1987) Temporal encoding of two-dimensional patterns by single units in primate inferior temporal cortex. II. Quantification of response waveform. J Neurophysiol 57:147–161
Rieke F, Warland D, de Ruyter van Steveninck R, Bialek W (1997) Spikes: exploring the neural code. MIT Press, Cambridge
Samonds JM, Bonds AB (2004) From another angle: differences in cortical coding between fine and coarse discrimination of orientation. J Neurophysiol 91:1193–1202
Samonds JM, Allison JD, Brown HA, Bonds AB (2003) Cooperation between area 17 neuron pairs enhances fine discrimination of orientation. J Neurosci 23:2416–2425
Schreiber S, Fellous JM, Tiesinga P, Sejnowski TJ (2004) Influence of ionic conductances on spike timing reliability of cortical neurons for suprathreshold rhythmic inputs. J Neurophysiol 91:194–205
Segundo JP, Perkel DH (1969) The nerve cell as an analyzer of spike trains. In: Brazier MAB (ed) The interneuron. University of California Press, Berkeley, pp 349–390
Sellers P (1974) On the theory and computation of evolutionary distances. SIAM J Appl Math 26:787–793
Sen K, Jorge-Rivera JC, Marder E, Abbott LF (1996) Decoding synapses. J Neurosci 16:6307–6318
Shannon CE, Weaver W (1949) The mathematical theory of communication. University of Illinois Press, Urbana
Singh G, Memoli F, Ishkhanov T, Sapiro G, Carlsson G, Ringach DL (2008) Topological analysis of population activity in visual cortex. J Vision 8:1–18

Slepian D (1976) On bandwidth. Proc IEEE 64:292–300
Softky WR, Koch C (1993) The highly irregular firing of cortical cells is inconsistent with temporal integration of random EPSPs. J Neurosci 13:334–350
Stopfer M, Bhagavan S, Smith BH, Laurent G (1997) Impaired odour discrimination on desynchronization of odour-encoding neural assemblies. Nature 390:70–74
Tenenbaum JB, de Silva V, Langford JC (2000) A global geometric framework for nonlinear dimensionality reduction. Science 290:2319–2323
Tiesinga PHE (2004) Chaos-induced modulation of reliability boosts output firing rate in downstream cortical areas. Phys Rev E Stat Nonlin Soft Matter Phys 69:031912
Treves A, Panzeri S (1995) The upward bias in measures of information derived from limited data samples. Neural Comput 7:399–407
Tversky A (1977) Features of similarity. Psychol Rev 84:327–352
Tversky A, Gati I (1982) Similarity, separability, and the triangle inequality. Psychol Rev 89:123–154
van Rossum MC (2001) A novel spike distance. Neural Comput 13:751–763
Victor JD (2000) Asymptotic bias in information estimates and the exponential (Bell) polynomials. Neural Comput 12:2797–2804
Victor JD (2002) Binless strategies for estimation of information from neural data. Phys Rev E 66:51903
Victor JD (2006) Approaches to information-theoretic analysis of neural activity. Biological Theory 1:302–316
Victor JD, Purpura KP (1996) Nature and precision of temporal coding in visual cortex: a metric-space analysis. J Neurophysiol 76:1310–1326
Victor JD, Purpura KP (1997) Metric-space analysis of spike trains: theory, algorithms and application. Network 8:127–164
Victor JD, Goldberg DH, Gardner D (2007) Dynamic programming algorithms for comparing multineuronal spike trains via cost-based metrics and alignments. J Neurosci Methods 161:351–360
Wu L, Gotman J (1998) Segmentation and classification of EEG during epileptic seizures. Electroencephalogr Clin Neurophysiol 106:344–356
Wuerger SM, Maloney LT, Krauskopf J (1995) Proximity judgments in color space: tests of a Euclidean color geometry. Vision Res 35:827–835