ELSEVIER

Journal of Neuroscience Methods 161 (2007) 351–360

JOURNAL OF NEUROSCIENCE METHODS

www.elsevier.com/locate/jneumeth

Dynamic programming algorithms for comparing multineuronal spike trains via cost-based metrics and alignments

Jonathan D. Victor[a,*], David H. Goldberg[b], Daniel Gardner[a,b]

[a] *Department of Neurology and Neuroscience, Weill Medical College of Cornell University, 1300 York Avenue, New York City, NY 10021, United States*

[b] *Laboratory of Neuroinformatics, Department of Physiology and Biophysics, Weill Medical College of Cornell University, 1300 York Avenue, New York City, NY 10021, United States*

Received 9 August 2006; received in revised form 31 October 2006; accepted 1 November 2006

Abstract

Cost-based metrics formalize notions of distance, or dissimilarity, between two spike trains, and are applicable to single- and multineuronal responses. As such, these metrics have been used to characterize neural variability and neural coding. By examining the structure of an efficient algorithm [Aronov D, 2003. Fast algorithm for the metric-space analysis of simultaneous responses of multiple single neurons. J Neurosci Methods 124(2), 175–79] implementing a metric for multineuronal responses, we determine criteria for its generalization, and identify additional efficiencies that are applicable when related dissimilarity measures are computed in parallel. The generalized algorithm provides the means to test a wide range of coding hypotheses.

Keywords: Neural coding; Spike trains; Metric space; Sequence comparison; Informatics

1. Introduction

Spike train metrics (Victor and Purpura, 1996, 1997) are used to characterize neural variability and neural coding in a range of neurophysiologic contexts, especially sensory systems; see recent review by Victor (2005). Metric approaches formalize spike train neural activity as a sequence of events (Segundo and Perkel, 1969). This viewpoint contrasts with two ways to represent neural activity in a vector space: as a rate (i.e., a continuous function of time) or as a function of discretely sampled time (Rieke et al., 1997). The choice of viewpoint has implications for the overall approach to data analysis. Vector spaces (including spaces of time series) have a natural means of defining a Euclidean "distance", based on their inner (scalar) product. In contrast, distances that naturally arise from sequence comparisons are typically non-Euclidean (Aronov and Victor, 2004). Non-Euclidean distances are necessary to account for some aspects of neural coding (Hopfield, 1995; Wuerger et al., 1995).

However, non-Euclidean distances determined by spike train metrics are more difficult to compute than distances in a vector space. For sequences of activity of a single neuron, the algorithmic problem for a simple spike train metric, $D^{\text{spike}}[q]$, where q is the cost per unit time for moving a spike, is closely analogous to that of biological sequence comparison (Needleman and Wunsch, 1970; Sellers, 1974), and a very similar dynamic programming algorithm is applicable (Victor and Purpura, 1996, 1997). Computational complexity is M^2, where M is the typical number of spikes in the responses to be compared. Multineuronal activity, recordings of which are becoming more and more widely available (Gray et al., 1995; Kralik et al., 2001; Nicolelis et al., 2003), can be considered as a sequence of labeled events (in which the label indicates the neuron of origin). The metric-space method readily extends to this context (Aronov et al., 2003; Samonds and Bonds, 2004). A straightforward extension of the algorithm for the single-neuron metric $D^{\text{spike}}[q]$ leads to an algorithm for a simple multineuronal metric, $D^{\text{spike}}[q,k]$. This algorithm has a computational complexity of M^{2L}, where L is the number of distinct neurons. Recently, Aronov (2003) dramatically improved this to M^{L+1}, via a dynamic-programming algorithm that treats the compared spike trains asymmetrically.

Generically, spike train metrics have parameters that indicate the extent to which the metric is sensitive to various features of the spike train (e.g., sensitivity to timing in $D^{\text{spike}}[q]$ and $D^{\text{spike}}[q, k]$ is determined by q, and sensitivity to neuron of ori-

[*] Corresponding author.
E-mail address: jdvicto@med.cornell.edu (J.D. Victor).

0165-0270/$ – see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.jneumeth.2006.11.001

gin in $D^{\text{spike}}[q, k]$ is determined by k). In this paper, we extend this algorithm to a wide range of single- and multineuronal spike train metrics. We also show how the algorithm can be modified to calculate spike train metrics for many values of the parameters in an efficient manner. As part of an ongoing effort to provide information-theoretic tools to the neuroscience community, implementations of algorithms for $D^{\text{spike}}[q]$ and $D^{\text{spike}}[q, k]$ described below are available in the Spike Train Analysis Toolkit (Goldberg et al., 2005) and can be obtained from the website http://neuroanalysis.org.

After some preliminary definitions, we review the notion of a cost-based metric, and then consider a partially distinct approach to define dissimilarities of event sequences, based on "alignments." We observe that the intersection of cost-based metrics and alignments includes the basic multineuronal cost-based spike time metric $D^{\text{spike}}[q, k]$. As we then show, the efficient dynamic programming algorithm (Aronov, 2003) for $D^{\text{spike}}[q, k]$, when viewed as an algorithm to identify alignments, is capable of substantial generalization. We then comment on matters of implementation and provide examples.

2. Results

2.1. Preliminaries

A spike train (a neural response) is considered to be a sequence of events, each occurring at a specific time and associated with a discrete label (the neuron of origin). More formally, a *spike train* A is a sequence of $M(A)$ spike times $A_1, A_2, \ldots, A_{M(A)}$, each with an associated label, $a_1, a_2, \ldots, a_{M(A)}$. The spike times A_k are non-decreasing real numbers, and may coincide. The labels are drawn from a set $\{1, \ldots, L\}$ of abstract tags. We use $A[w]$ to denote the subsequence of A that includes just the spikes associated with a label w. Thus, $M(A[w])$ is the number of spikes in A with the label w. We also use $\vec{M}(A)$ to denote a vector whose wth element is $M(A[w])$. A spike train is the disjoint union of its subtrains $A[w]$, i.e.,

$$\bigcup_{w=1}^{L} A[w] = \{A_1, \ldots, A_{M(A)}\}, \quad \text{and} \quad \sum_{w=1}^{L} M(A[w]) = M(A).$$

2.2. Metrics, alignments, and strains

We will consider two ways to measure dissimilarity between two spike trains A and B. The first, "cost-based metrics" (Victor, 2005; Victor and Purpura, 1996, 1997) is based on a set of elementary transformations between spike trains. Each elementary transformation (e.g., deleting a spike, inserting a spike, shifting a spike in time, or changing the label of a spike) is associated with a cost. In a cost-based metric, the distance (dissimilarity) between two spike trains is the minimum total cost of any sequence of elementary steps that transforms A into B.

The second way of measuring dissimilarity is based on the notion of an *alignment* between two spike trains, $X(A, B)$ (or simply X). An alignment (Fig. 1A) indicates a correspondence between the components of spike train A and those of spike train B, via a set of links between pairs of spikes. As shown in Fig. 1A, an alignment may leave some spikes unlinked in one or both of the two trains. We will assign a *strain* to each alignment, and the measure of dissimilarity between A and B will be taken to be the smallest strain of all possible alignments $X(A, B)$.

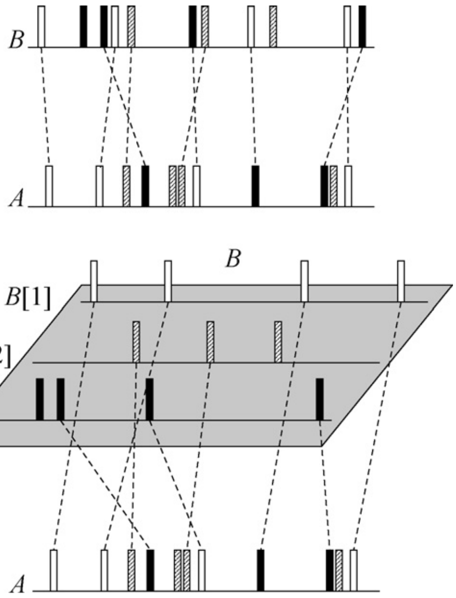

Fig. 1. (A) A candidate alignment of two multineuronal spike trains. The neuron of origin (the "label") of each spike is indicated by shading, and an alignment consists of links between pairs of individual spikes. An alignment may link spikes from different neurons, and all spikes need not be linked. The alignment shown is not necessarily an "efficient" alignment, but is typical of one considered by the dynamic programming algorithm (Aronov, 2003) for $D^{\text{spike}}[k, q]$. (B) A necessary, but not sufficient, condition for an efficient alignment. The subtrains of spike train B are separated into individual parallel lines. This defines a unique plane for each label w, containing the subtrain $B[w]$ and the full train A. In each plane (i.e., the plane of links in which the second member of each pairing comes from the same subtrain $B[w]$) links cannot intersect. For further details, see text. Adapted from Aronov (2003) with permission.

We will restrict consideration to strains that depend only on the number of unlinked spikes and on the time differences between the spikes connected by the links. We will further restrict consideration to strains in which these components combine additively. More formally, we will consider strains of the form

$$S(X(A, B)) = \sum_{w=1}^{L} I_{1,w}(U_{1,w}) + \sum_{w=1}^{L} I_{2,w}(U_{2,w}) + \sum_{w_1=1}^{L}\sum_{w_2=1}^{L}\sum_{j} J_{w_1,w_2}\left(\left|T_{w_1,w_2,j}\right|\right). \quad (1)$$

In this equation, $U_{1,w}$ and $U_{2,w}$ are the number of unlinked spikes of label w in each train, and T is a table of lists $T_{w_1,w_2,j}$ of the time differences between linked spikes of label w_1 in A and label w_2 in B. $I_{1,w}$ and $I_{2,w}$ are non-decreasing functions that determine the contribution to the strain of unpaired spikes of each label (w) in each train (1 for A, 2 for B). Similarly, $J_{w_1,w_2}(t)$ is a non-decreasing function that determines the contribution to