



# Fast algorithm for the metric-space analysis of simultaneous responses of multiple single neurons

Dmitriy Aronov\*

Department of Biological Sciences, Columbia University, 1002 Fairchild, Mail Code 2436, 1212 Amsterdam Avenue, New York, NY 10027, USA

Received 30 July 2002; received in revised form 7 January 2003; accepted 11 January 2003

## Abstract

Spike train metrics quantify the notion of dissimilarity, or distance, between spike trains and between multineuronal responses (J. Neurophysiol. 76 (1996) 1310, Network 8 (1997) 127). We present a new algorithm for the implementation of a metric based on the timing of individual spikes and on their neurons of origin. This algorithm surpasses the earlier approach in speed by a factor that grows exponentially with the number of neurons, substantially extending the applicability of metric space methods to the study of coding in larger neuronal populations.

© 2003 Elsevier Science B.V. All rights reserved.

*Keywords:* Spike trains; Multineuronal analysis; Metric spaces; Dynamic programming algorithms

## 1. Introduction

Investigations of neural coding require methods to analyze neuronal discharges that make few non-empirical assumptions about their underlying temporal structure. Spike train metric spaces have been introduced for this purpose by Victor and Purpura, (1996, 1997) as a quantitative measurement of dissimilarity, or distance, between trains of action potentials. This approach takes into account the number of spikes in individual spike trains and parameterizes the importance of temporal coding. Metric space methods have also been extended to simultaneous responses of multiple neurons (Aronov et al., 2003) using an approach that parameterizes the importance of distinguishing individual neurons in population coding.

Spike train metrics have been used to analyze aspects of temporal coding in a number of sensory systems, including the olfactory (MacLeod et al., 1998) and auditory (Machens et al., 2001) systems in insects, the electric sensory system in fish (Kreiman et al., 2000), and

various parts of the visual system in several different species (Victor and Purpura, 1996; Keat et al., 2001). Their applications include estimation of transmitted information (Victor and Purpura, 1997; Mechler et al., 2002), analysis of burst events (Keat et al., 2001), analysis of variability (Kreiman et al., 2000) and reconstruction of response spaces and investigation of joint coding by pairs of neurons (Aronov et al., 2003).

Implementation of the metric space approach requires fast algorithms for the calculation of distances between pairs of spike trains or pairs of multineuronal responses. A related problem of calculating distances between genetic sequences has been solved with the efficient dynamical programming algorithm introduced by Sellers, (1974). This algorithm was adapted for single-unit spike train metrics based on either spike times or interspike intervals (Victor and Purpura, 1997) and extended to multi-unit spike time metrics by Aronov et al., (2003). The computation time for these algorithms is a polynomial function of the number of spikes. The degree of this polynomial is  $2L$ , where  $L$  is the number of neurons. Here, we present a novel multi-unit algorithm, in which the degree of the polynomial is only  $L+1$ . This dramatic improvement makes metric space methods applicable to larger populations of neurons.

\* Tel.: +1-212-854-5023; fax: +1-212-865-8246.

E-mail address: da2006@columbia.edu (D. Aronov).

## 2. Methods

All algorithms described in this paper were implemented and tested in C and MATLAB programming environments.

## 3. Results

We present a novel algorithm for metric space calculations of distances between multi-unit neuronal discharges. We first review the definitions of spike time metrics and discuss some of their relevant properties. We then describe the existing algorithms, which serve as a basis for the current result. Finally, we present the derivation of the new multi-unit algorithm.

### 3.1. Spike time metrics

Spike time metrics characterize the similarity of neuronal discharges based on the timing of individual spikes. For spike train metrics in general, the distance between two spike trains is defined as the minimal ‘cost’ of transforming one spike train into the other via a sequence of elementary steps. For spike time metrics, the following elementary transformation steps are allowed: insertion of a spike (for a cost of 1), deletion of a spike (for a cost of 1), and shifting a spike by an amount of time  $\Delta t$  (for a cost  $q|\Delta t|$ ). Here,  $q$  is a parameter measured in  $s^{-1}$  that quantifies the temporal precision relevant to spike timing. When  $q = 0$ , there is no cost for shifting a spike in time, and the metric reduces to a comparison of spike trains based only on the number of spikes they contain. When  $q > 0$ , a spike in one train is considered a potential match for a spike in the other train only if the two spikes occur within  $2/q$  of each other. If the interval between two spikes is longer than  $2/q$ , the spikes are not considered to occur at sufficiently similar times to correspond, since deleting one of the spikes and inserting it at the same point in time as the other (for a cost of 2) is cheaper than coinciding the two spikes by shifting one of them. For further details and discussion, see Victor and Purpura, (1996, 1997).

Metric spaces can be extended to define distances between simultaneous discharges of multiple neurons and to parameterize the importance of distinguishing spikes fired by different cells. A simultaneous recording of several neurons can be represented by a single sequence of labeled events, or a ‘labeled spike train’. The label assigned to each spike in this sequence indicates its neuron of origin. For notation purposes, the integers  $1, 2, \dots, L$  are used as labels for a population of  $L$  neurons. These integers are abstract tags and do not imply any ordering. Cost based metrics that quantify the dissimilarity of labeled spike trains allow all elementary transformations used by the single-unit

metrics. In addition, multi-unit metrics allow an elementary step in which a spike’s label is changed, for a cost of  $k$ .

The dimensionless parameter  $k$  quantifies the importance of distinguishing spikes fired by different neurons. When  $k = 0$ , there is no cost for reassigning a spike’s label, and the entire population discharge is viewed as a sequence of spikes fired by a single-unit. When  $k \geq 2$ , spikes fired by different neurons are never considered similar, since deleting two spikes with different labels (for a cost of 2) is not more expensive than matching their labels. For values of  $k$  between 0 and 2, spikes fired by different neurons some time  $\Delta t$  apart can be matched in a transformation if the cost of this transformation step,  $q|\Delta t| + k$ , is less than 2. Thus, for these values of  $k$ , spikes fired by different neurons can be considered similar if they occur within  $(2-k)/q$  of each other.

### 3.2. Algorithms

Calculation of the distance between a pair of spike trains is straightforward, provided one knows a sequence of elementary steps that minimizes the cost of transforming one spike train into the other. However, this minimal path can be chosen from the infinite number of possible transformation sequences. Dynamic programming algorithms introduced by Sellers, (1974) for genetic sequences and their adaptations for spike trains identify a minimal path by eliminating most paths that are not minimal and leaving a relatively small number of possibilities. Here, we present a novel algorithm for multi-unit metric calculations that is a major improvement on the multi-unit algorithm used by Aronov et al., (2003), which was in turn a direct adaptation of the Sellers algorithm. We review the single-unit algorithm of Victor and Purpura, (1997) first, since it provides the basis for the new multi-unit algorithm.

### 3.3. Single-unit algorithm

This algorithm is based on several attributes that any minimal path must possess. A minimal path cannot include an insertion of a spike that is later deleted, since the cost of such path can be reduced by eliminating both steps. In addition, a minimal path cannot include an insertion of a spike that is later shifted, since the cost of such path can be reduced by inserting the spike at its final position in time and eliminating the shift. Similarly, a spike that is later deleted cannot be shifted in a minimal path. Furthermore, a spike can only be shifted in one direction, since replacing multiple shifts in different directions by a single shift reduces the cost. The final property of minimal paths can be understood from a diagrammatic representation of spike ‘trajectories’ (Fig. 1A). In a minimal path, the trajectories of

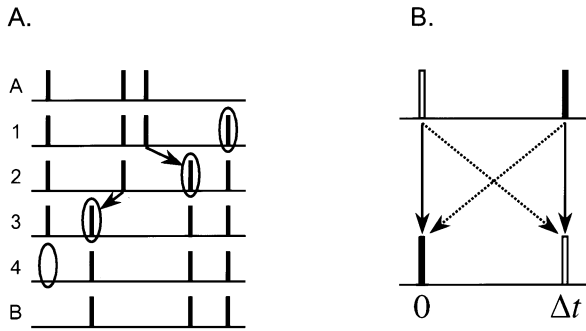


Fig. 1. Algorithm for single-unit metric calculations. (A) A minimal transformation path between two spike trains, A and B. The transformation consists of four elementary steps: insertion of a spike (step 1), shifting of spikes (steps 2 and 3), and deletion of a spike (step 4). Trajectories of different spikes in the minimal path diagram do not intersect. (B) A simple example of simultaneous two-neuron discharges for which the single-unit algorithm fails to find the minimal path. Each spike in the two labeled spike trains is shaded according to the neuron that fired it. The transformation path without spike trajectory crossings (solid arrows) consists of two label reassignments that cost  $2k$ . The path that includes such crossing (dotted arrows) consists of two spike shifts that cost  $2q\Delta t$ . When  $\Delta t < k/q$ , trajectory intersection minimizes the cost of the transformation.

individual spikes cannot intersect, since uncrossing them reduces the amount of shifting, and thus, the total cost of the transformation.

The above observations imply that, in a minimal path, each spike can be either deleted or shifted once to coincide with a spike in the other spike train. Also, a spike can be inserted at a time that matches the occurrence of a spike in the other spike train. However, the same result can be achieved for the same cost by deleting the corresponding spike in the other spike train. Therefore, paths that include insertions need not be considered if deletions from both spike trains are allowed.

We now consider calculating the distance between a spike train  $S_a$ , consisting of spikes at times  $a_1, a_2, \dots, a_m$ , and a spike train  $S_b$ , consisting of spikes at times  $b_1, b_2, \dots, b_n$ . It is possible that either the last spike in  $S_a$  or the last spike in  $S_b$  is deleted (for a cost of 1) in the minimal path. If neither of them is deleted, the above observations imply that the two spikes must be shifted to coincide with each other. The cost of this shift is  $q|a_m - b_n|$ . The two deletions and the shift form three alternatives for the minimal path. The distance between  $S_a$  and  $S_b$  is given by the minimum of three quantities corresponding to each of these alternatives. This provides an inductive algorithm for determining the distance between  $S_a$  and  $S_b$ , since calculating the cost of each alternative requires finding distances between smaller spike trains. We use  $G_{i,j}$  to denote the distance between a spike train composed of the first  $i$  spikes of  $S_a$  and a spike train composed of the first  $j$  spikes of  $S_b$ . Then, the recursion step can be written as

$$G_{i,j} = \min\{G_{i-1,j} + 1, G_{i,j-1} + 1, G_{i-1,j-1} + q|a_i - b_j|\} \quad (1)$$

Computationally, it is much more efficient to implement this idea in an inductive manner, rather than use explicit recursions. In the forward direction, the calculation can be viewed as a two-dimensional spreadsheet, in which the cell in the  $i$ th row and the  $j$ th column contains  $G_{i,j}$ . The first row and column of the spreadsheet can be filled by noting that  $G_{i,0} = i$  and  $G_{0,j} = j$ . This is due to the fact that the transformation path between any spike train and the null spike train consists of deleting all spikes. The cell in row  $m$  and column  $n$  contains  $G_{m,n}$ , which is the desired distance between  $S_a$  and  $S_b$ . The number of entries in the spreadsheet that need to be calculated is  $m n$ , or  $O(N^2)$ , where  $N$  is the typical number of spikes in a spike train.

### 3.4. Multi-unit algorithm

The single-unit algorithm cannot be directly applied to multi-unit metrics because not all stated observations about minimal paths hold true for labeled spike trains. Contrary to one of these observations, ‘trajectories’ of spikes can intersect in a minimal path between two-labeled spike trains. This occurs when crossing trajectories decreases the cost of reassigning labels sufficiently to compensate for the increase in the cost of shifting spikes. A simple example of such situation is shown in Fig. 1B. In this example,  $k < 2$ . Without crossing trajectories, the minimal path (solid arrows) would consist of two label reassignments that cost  $2k$ . If crossings are allowed, however, there is an alternative path (dotted arrows) that contains no label reassignments and consists of two spike shifts that cost  $2q\Delta t$ . Thus, crossing trajectories is necessary to minimize the cost of the transformation if  $\Delta t < k/q$ . For multi-unit metrics, the prohibition of crossing trajectories must be modified: If, in one spike train, two spikes have the same label, then their trajectories cannot intersect, since uncrossing them decreases the cost.

We consider calculating the multi-unit metric distance between two labeled spike trains,  $S_a$  and  $S_b$ , consisting of spikes with  $L$  different labels. Spikes with label  $w$  ( $w = 1, 2, \dots, L$ ) are located at times  $a_1^{(w)}, a_2^{(w)}, \dots, a_{m_w}^{(w)}$  in  $S_a$  and at times  $b_1^{(w)}, b_2^{(w)}, \dots, b_{n_w}^{(w)}$  in  $S_b$ . For reasons that will become clear later, we introduce a second notation for  $S_a$ . We use  $\alpha_1, \alpha_2, \dots, \alpha_M$  to specify the times of all spikes in  $S_a$  and  $r_1, r_2, \dots, r_M$  to specify their labels. Here,  $M = \sum m_w$  is the total number of spikes in  $S_a$ .

We now consider possible transformations of the last spike in  $S_a$  in a minimal path (Fig. 2A). One alternative is that this spike is deleted for a cost of 1. If the last spike in  $S_a$  is not deleted, it must be ‘linked’ to one of the spikes in  $S_b$  by a shift.  $S_b$  is considered to be composed of  $L$  subtrains, each containing spikes of one of the

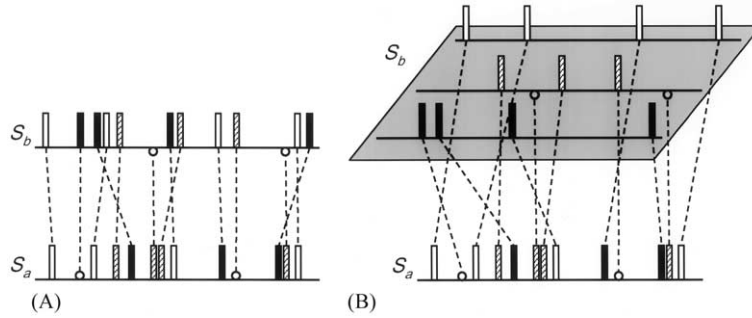


Fig. 2. Algorithm for multi-unit metric calculations. (A) A minimal transformation path between two labeled spike trains,  $S_a$  and  $S_b$ , that contain simultaneously-recorded discharges of three neurons. Each spike is shaded according to the neuron that fired it. Dotted lines connect pairs of spikes that are 'linked' in the transformation path by a shift and, where necessary, a label reassignment. Circles indicate deletions of spikes. (B) The multi-unit algorithm separates one of the labeled spike trains (in this case  $S_b$ ) into subtrains, each containing spikes fired by one of the neurons. Every link between spikes in the resulting transformation path diagram lies in one of three planes, defined by  $S_a$  and each of the three subtrains. The algorithm takes advantage of the fact that, in a minimal path, trajectories of spikes cannot intersect in any of these planes.

labels (Fig. 2B). That is, one set of possibilities is that the last spike in  $S_a$  is linked to the last spike in one of the subtrains. These possibilities form  $L$  additional alternatives for the minimal path, one corresponding to each of the  $L$  subtrains. Another possibility is that the last spike in  $S_a$  is linked to a spike in  $S_b$  that is not last in its subtrain. In such case, the last spike in the corresponding subtrain of  $S_b$  must be deleted to prevent crossing its trajectory with the trajectory of a spike that has the same label. This forms  $L$  more alternatives for the minimal path, corresponding to the deletion of the last spike in each of the subtrains of  $S_b$ .

The above possibilities for the last spike in  $S_a$  form  $2L+1$  alternatives that need to be considered in the inductive algorithm. The distance between  $S_a$  and  $S_b$  is the minimum of  $2L+1$  quantities corresponding to each of these alternatives. We let  $G_{i;j_1,j_2,\dots,j_L}$  be the distance between two 'reduced' spike trains: the spike train consisting of the first  $i$  spikes of  $S_a$  and the spike train consisting of the first  $j_w$  spikes of each label  $w$  from  $S_b$ . The recursion step can then be written as

$$G_{i;j_1,j_2,\dots,j_L} = \left\{ \begin{array}{l} G_{i-1;j_1,j_2,\dots,j_L} + 1, \\ \min_{w;j_w > 0} G_{i-1;j_1,\dots,j_w-1,\dots,j_L} + q|\alpha_i - b_{j_w}^{(w)}| + k(1 - \delta_{w,i}), \\ \min_{w;j_w > 0} G_{i;j_1,\dots,j_w-1,\dots,j_L} + 1 \end{array} \right\} \quad (2)$$

Here, the Kronecker delta (defined as  $\delta_{i,j} = 1$  for  $i = j$  and 0 otherwise) is used to indicate that  $k$  must be added to the cost only if labels of linked spikes are different. The first line of Eq. (2) corresponds to the alternative of deleting the last spike in  $S_a$ . The second line corresponds to the  $L$  alternatives of linking the last spike in  $S_a$  with one of the spikes in  $S_b$ . The condition  $j_w > 0$  prevents operations with subtrains that contain no spikes. The third line corresponds to  $L$  alternatives in which a spike from  $S_b$  is deleted.

The above procedure can be viewed as a spreadsheet that has  $L+1$  dimensions. The coordinate in the first

dimension indicates the number of spikes in the reduced spike trains of  $S_a$ . Dimensions 2 through  $L+1$  are devoted to the reduced spike trains of  $S_b$ . These coordinates indicate the numbers of spikes of each label. The cell that is last along every dimension in the spreadsheet contains  $G_{M;n_1,n_2,\dots,n_L}$ , which is the desired distance between  $S_a$  and  $S_b$ . The initial conditions are based on the fact that the minimal transformation path between any spike train and the null spike train consists of deleting all spikes. For the above inductive algorithm, these conditions are  $G_{i;0,0,\dots,0} = i$  and  $G_{0;j_1,j_2,\dots,j_L} = \sum j_w$ .

#### 4. Discussion

Computation time of the derived algorithm can be deduced from the size of the inductive spreadsheet. If, as above,  $S_b$  is separated into  $L$  subtrains during computation, the total number of entries that need to be calculated in the spreadsheet is  $\sum m_w \Pi n_w$ . Interestingly, the new algorithm does not treat  $S_a$  and  $S_b$  symmetrically. Thus, the same procedure as above can be performed with the roles of the two spike trains switched. If this is done and  $S_a$  is separated into subtrains instead, the total number of entries is  $\Pi m_w \sum n_w$ . Clearly, an efficient implementation of the algorithm can select the arrangement for which the number of entries is smaller. In either case, if  $N$  is the typical number of spikes discharged by a single neuron, the typical number of entries to calculate is approximated by  $LN^{L+1}$ . As described above, each entry is a minimum of  $2L+1$  quantities. Finding this minimum requires  $2L+1$  operations for each entry. Thus, the total number of operations is approximately  $(2L+1)LN^{L+1}$ , or  $O(N^{L+1})$  with respect to the number of spikes per neuron.

The multi-unit algorithm used in our previous work (Aronov et al., 2003), which was the straightforward generalization of the Sellers algorithm, also implemen-

ted a multidimensional recursive spreadsheet, but required  $N^{2L}$  entries to be calculated. Each entry was a minimum of  $L^2 + 2L$  quantities, requiring more computation at each cell of the spreadsheet as well. Thus, the present result is an improvement of this algorithm by a factor of approximately  $[(L+2)/(2L+1)]N^{L-1}$ , or  $O(N^{L-1})$ . Thus, the new algorithm is typically more efficient even for pairs of neurons. For  $L=1$ , the algorithm reduces to the single-unit algorithm described above.

The algorithm derived in this paper applies to some generalizations similar to those described for the single-unit algorithm (Victor and Purpura, 1997). For instance, the present algorithm can be used for spike time metrics in which the cost of shifting a spike is not the linear function  $q|\Delta t|$ , but any increasing, concave down function of  $|\Delta t|$ , such as  $1 - \exp\{-q|\Delta t|\}$ . In addition, it can be applied to metrics in which the cost of relabeling spikes is generalized from  $k$  to  $k_{ij}$  that depends on the neurons of origin  $i$  and  $j$ .

The inductive nature of the algorithm allows for simple pruning techniques that reduce the amount of computation from the amount deduced theoretically. In induction, the minimal transformation path between two spike trains is built up from paths between smaller spike trains. Thus, the cost of these paths can be compared to some upper bound at any time during computation. If the cost of some path exceeds the upper bound, all paths between larger spike trains that are built up from it can be immediately eliminated. The upper bound can be the cost of any non-minimal path that can be found fairly quickly. For instance, the fast single-unit algorithm can be used to find such path by either not allowing spike trajectories to be crossed or by not allowing the linkage of differently-labeled spikes. Despite pruning, computation time of the procedure grows exponentially with the number of neurons. Nevertheless, metric space methods are suitable for populations of several neurons, given an efficient implementation of the algorithm.

The new algorithm can quickly extract a large number of pairwise metric distances between simultaneous multineuronal responses recorded in a typical experiment. Such distances model response dissimilarities without making excessively strong assumptions (such as linearity) about their underlying structure. The metric quantifications can then be used for further analysis of neural coding, such as for estimating information content in multiple-trial recordings of responses to different stimuli or for analyzing the structure of the response space (Victor and Purpura, 1997). The extension of these methods to simultaneous responses of

multiple neurons can be used to address various issues of population coding, such as the similarity and redundancy of coding in neighboring neurons (Aronov et al., 2003). In particular, the parameter  $k$  in the model parameterizes the importance of distinguishing individual neurons, providing means for analyzing a continuum of coding behaviors between the extremes of the ‘labeled line code,’ in which neurons are treated separately, and the ‘summed population code,’ in which the neuronal population is viewed as a single-unit. The question of how neuronal populations represent information, both spatially and temporally, becomes especially important for larger numbers of neurons (Reich et al., 2001). The algorithm developed in this paper makes metric space methods more applicable for addressing this question.

### Acknowledgements

This work was supported by EY9314. The author thanks Dr. Jonathan Victor for invaluable guidance and discussions, as well as for comments on the manuscript.

### References

- Aronov D, Reich DS, Mechler F, Victor JD. Neural coding of spatial phase in V1 of the macaque monkey. *J Neurophysiol* 2003; in press.
- Keat J, Reinagel P, Reid RC, Meister M. Predicting every spike: a model for the responses of visual neurons. *Neuron* 2001;30(3):646–7.
- Kreiman G, Krahe R, Metzner W, Koch C, Gabbiani F. Robustness and variability of neuronal coding by amplitude-sensitive afferents in the weakly electric fish *eigenmannia*. *J Neurophysiol* 2000;84(1):189–204.
- Machens CK, Stemmler MB, Prinz P, Krahe R, Ronacher B, Herz AV. Representation of acoustic communication signals by insect auditory receptor neurons. *J Neurosci* 2001;21(9):3215–27.
- MacLeod K, Backer A, Laurent G. Who reads temporal information contained across synchronized and oscillatory spike trains? *Nature* 1998;395(6703):693–8.
- Mechler F, Reich DS, Victor JD. Detection and discrimination of relative spatial phase by V1 neurons. *J Neurosci* 2002;22(14):6129–57.
- Reich DS, Mechler F, Victor JD. Independent and redundant information in nearby cortical neurons. *Science* 2001;294(5551):2566–8.
- Sellers PH. On the theory and computation of evolutionary distances. *SIAM J Appl Math* 1974;26:787–93.
- Victor JD, Purpura KP. Nature and precision of temporal coding in visual cortex: a metric-space analysis. *J Neurophysiol* 1996;76(2):1310–26.
- Victor JD, Purpura KP. Metric space analysis of spike trains: theory, algorithms and application. *Network* 1997;8:127–64.