Non-uniform quantization of neural spike sequences through an information distortion measure[☆]

Alexander G. Dimitrov[a,*], John P. Miller[a], Zane Aldworth[a], Tomáš Gedeon[b]

[a]*Center for Computational Biology, Montana State University, Bozeman, MT 59717-3505, USA*
[b]*Department of Mathematical Sciences, Montana State University, Bozeman, MT 59717, USA*

Abstract

There have been various suggestions about how information is encoded in neural spike trains: by the number of spikes, by the temporal correlations, or by complete patterns. The latter scheme is most general, and encompasses many others. However, the search for pattern codes requires exponentially more data than the search for mean rate or correlation codes. Here we describe a method that enables optimal use of whatever quantity of data is available. This method allows spike trains to be studied with variable, non-uniform temporal precision. Precision is optimized to provide a best lower bound for the information content of spike patterns given the available data. © 2001 Elsevier Science B.V. All rights reserved.

Keywords: Neural coding; Quantization; Information theory; Cricket

1. Introduction

A quantitative analysis of neural decoding involves parsing a string of spikes from one or several neurons, determining the relevant codewords and assigning corresponding sensory stimuli to the different spike patterns. Tools in Information Theory [1,7] allow us to characterize the quantity of information carried by a particular coding scheme, by estimating its entropy H and the mutual information I between the neural code and the input stimulus [2,6]. The knowledge of H and I places important bounds on the performance of any encoding scheme. Estimating these quantities from data is not a trivial task [2,4,9].

We present an approach to reduce the uncertainties in assessing the viability of a particular coding scheme due to limitations in the amount of available data. The approach quantizes spike patterns using a similarity measure which is intrinsic to the processed signal and the task at hand. This method allows spike trains to be studied with variable, non-uniform temporal precision. The coarsest representation is equivalent to a spike count code, while the most finely grained one corresponds to a precise spike timing code. The transition between representations is governed by a single parameter. When quantizing just the spike data, an appropriate measure is the entropy of spike patterns, which puts an upper bound on the amount of information that can be encoded with such a signal. If the spike patterns are related to a driving stimulus, a more appropriate measure to be maximized is the mutual information between stimulus and neural response. We show how these measures can be used to provide optimal use of all available data.

2. Quantization of spike patterns

In Information Theory, the entropy H and mutual information I have very specific meaning [7]. There are about $2^{nH(X)}$ distinct messages that can be transmitted with sequences of length n from a random source X. These are the *typical sequences* of X. For a pair of sources (X, Y), there are about $2^{nH(X,Y)}$ *jointly typical* sequence pairs. Due to noise or mismatch in the signal dimensionality, this mapping is not one to one. The number of clusters in (X^n, Y^n) that can be reliably communicated is about $2^{nI(X,Y)}$. Thus the knowledge of H and I places important bounds on the performance of any coding scheme. Estimating these quantities requires enormous amounts of data, which grow exponentially with the complexity of the coding scheme [4].

2.1. Quantization theory

The usual definitions of H and I are $H(X) = E_{p(x)} \log[1/p(x)]$ and $I(X, Y) = E_{p(x,y)} \log[p(x, y)/p(x)p(y)]$. Alternative definitions allow us to control the exponential growth of required data and obtain tighter estimates on lower bounds of these quantities. This is done through the use of *quantizers* [3]. A *quantizer* is any simple measurable function $f: X \to X_f$ from X to a *reproduction* space X_f with finitely many elements $\{x_f^i\}$. The quantizer f induces a partition $\{Q_f^i\}$ in X. The information quantities in X_f are

$$H(X_f) = E_{p(x_f)} \log \frac{1}{p(x_f)} = E_{p(Q_f)} \log \frac{1}{p(Q_f)}, \tag{1}$$

$$I(X_f, Y_g) = E_{p(x_f, y_g)} \log \frac{p(x_f, y_g)}{p(x_f)p(y_g)} = E_{p(Q_f, Q_g)} \log \frac{p(Q_f, Q_g)}{p(Q_f)p(Q_g)}. \tag{2}$$

A quantizer h refines f ($h > f$) if the partition Q_h of X induced by h refines the partition Q_f induced by f. If $h > f$, then

$$H(X_h) \geqslant H(X_f), \quad I(Y; X_h) \geqslant I(Y; X_f). \tag{3}$$

For any X and any quantizer f, $I(Y;X) \geqslant I(Y;X_f)$. If X is itself a discrete space, $H(X) \geqslant H(X_f)$, that is, estimates in the quantized spaces are always lower bounds of the actual information quantities. When X is continuous, $H(X)$ diverges with refinements. $I(Y;X)$ on the other hand can always be obtained as the least upper bound over all refinements.

The statements above suggest that quantizations could provide lower bound estimates of H and I of a neural spike sequence with controlled precision, since now the size of the pattern set is fixed by the size of the reproduction and could be potentially much lower than the size of the original patterns space.

2.2. Quantization functions for spike patterns

One particular quantization of spike trains changes a string of spikes to a shorter string of integers. The choice of specific quantizers is not critical to the estimation procedure since it can always be refined. For physiological recordings, there is always the ultimate refinement imposed by the temporal resolution of the instruments. Thus, the only condition that we impose on the quantizers is that at the finest resolution they recover the original spike pattern.

Let us first consider the single neuron case. One way to represent a neural activity pattern is as a string of ones and zeros of certain length. One possible quantization is to subdivide this string in a fixed number of consecutive substrings and to each assign an "activity" number. We currently assign the number of spikes in a substring as its activity, but other quantizations are possible. An example can be seen in Fig. 1. The quantization here is fine enough so that the resultant patterns are still binary, but much reduced in complexity (B). The coarsest representation consists of only a single substring, and the quantization function represents each pattern with the number of spikes contained in it.

A longer quantized string is a refinement of a shorter quantized string and can potentially provide a better estimate of the information quantities. On the other hand, a finer representation requires more data to support its estimates. We will look for an optimal refinement, which provides the best lower bound on information quantities given the available data. Given a fixed string size, we can look for the quantization (i.e., the configuration of substrings) that gives us the largest estimate of H or I. Such quantization will provide the greatest lower bound for this string length. Thus patterns in the quantization space have as much as possible of the information-carrying capacity of the raw spike train (if maximizing H), or retain as much as possible of the information about the stimulus (if maximizing I). The quantized patterns can serve as basis for extracting further details about the coding scheme used, to the extent that these are supported by the amount of data.

Multiple neurons can be handled in a very similar manner. The quantization function in this case first groups the neurons in clusters. Inside the cluster, neurons are considered indistinguishable, thus a member spike is a cluster spike. The output of a cluster is again a binary string of spikes, which is then quantized with the single unit quantization function.

3. Results

First, we present an example of the optimization landscape for a fixed string length. For visualization purposes we chose the case with three substrings, which is determined by two free parameters (envision Fig. 1 with only 2 bars in it, which define three boxes). The form of the cost functions, which we maximize, can be seen in Fig. 2. In part (A) we show the entropy estimate as a function of different partitions. The figure is symmetric with respect to exchange of T_1 and T_2. What it conveys is that different configurations affect the estimate of H. Some produce rather low estimates (e.g., all the configurations with $T_i \in [0,10]$ ms). The case where all patterns are represented only by the number of spikes is at $T = (0,0)$ and has the lowest entropy value here, hence the lowest information-carrying capacity. The coarser one-parametric representation can be observed on the diagonal $T_1 = T_2$ or on the sides $T_1 = 0$ or $T_2 = 0$. Several things are notable about this figure: (a) The cost function is a relatively smooth function of configurations. One can use an arbitrary optimization routine to find the optimum. Nevertheless it is not unimodal, so a mild stochastic correction to the optimization routine is necessary to find the global maximum. (b) Refinement matters. The optimal H estimate is more than twice larger than the spike counting case and about 1.5 times larger than the optimal 1d case at $T = [0,24]$ ms. (c) Configurations also matter. It is crucial to note that the entropy estimate is also about 1.5 larger than the *uniform* 2d configuration $T_1 = 16$ ms, $T_2 = 32$ ms, which is the one typically used when discussing spike timing codes (usually at finer quantization of $\delta T \approx 1$ ms). This suggests that a non-uniform quantization like the one performed here can provide a patterns representation *which retains more of the information carrying capacity* of a spike train than a uniform quantization.

Similar results can be observed when I is the quantity of interest. This is a much more interesting case, since now we are maximizing the *information carried by the quantization* about an external stimulus. The cost function show much greater
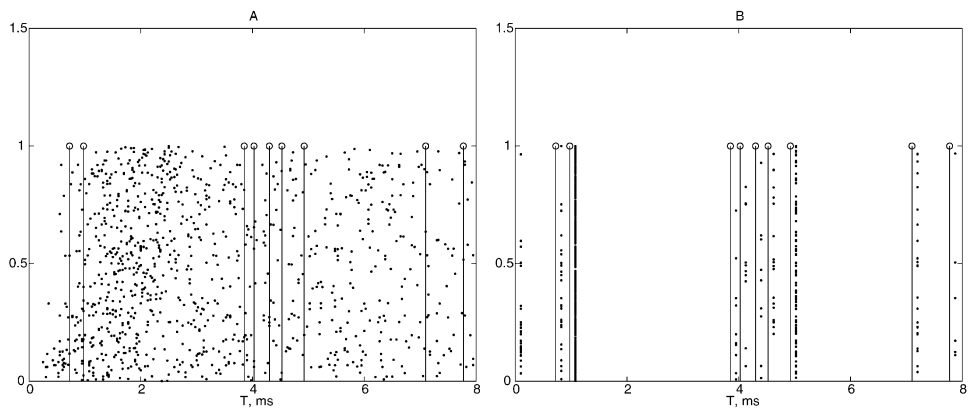

Fig. 1. A raster-plot of raw spike patterns with a particular quantization patterns superimposed on them (A) and the corresponding quantized patterns (B).
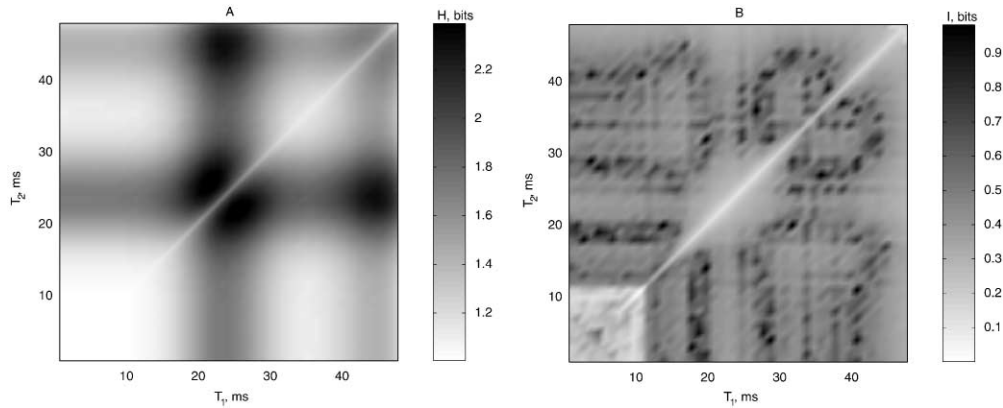
Fig. 2. The optimization landscape for two boundaries for an entropy (A) and mutual information (B) cost functions.
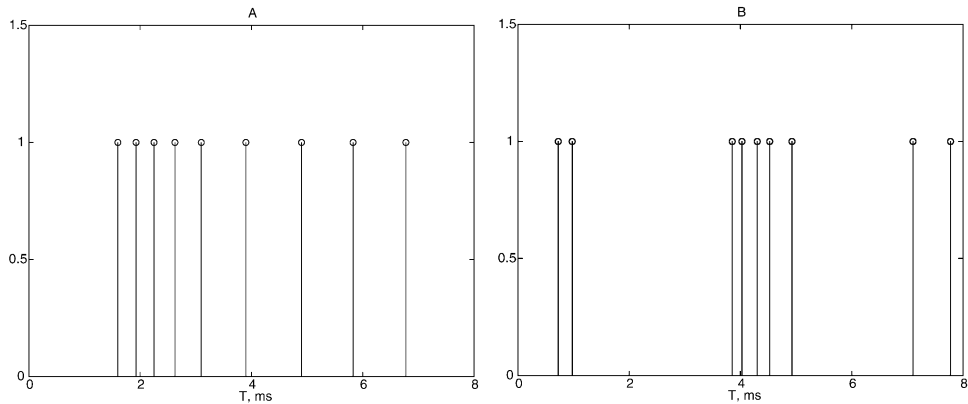

Fig. 3. Optimal configurations for the entropy (A) and mutual information (B) estimates in the 10 substrings (9 parameters) case. $H_{max} = (2.93 \pm 0.03)$ bits, $I_{max} = (1.4 \pm 0.11)$ bits.

irregularity than in the former case, thus it's maximization is now a non-trivial matter. We are still developing algorithms to improve on the Monte-Carlo method we currently use for this task. Yet all the points made for the entropy case are also valid here. The refinement improves tremendously the mutual information estimates (more than ten times!). The optimal configuration improves the estimate of the uniform configuration by about 60%. All the figures here are highly significant, since the bootstrap estimate of the standard deviation for these quantities was under 0.01 bits for the entropy case (A) and about 0.05 bits for the mutual information case.

We also present the results of applying this method at a finer quantization (Fig. 3). When maximizing the quantized entropy (A), the algorithm produces a relatively uniform configuration. The distribution of spikes in each bin is even more uniform

(not shown). It seems that the method is trying to approximate a uniform distribution of spikes across the bins, which is the maximum entropy distribution in this case. The results are radically different for the quantized mutual information case. It seems that the method is describing certain regions of the spike patterns rather precisely (e.g., the region between 4 and 5 ms, where spikes are tracked with sub-millisecond precision), while quantizing rather coarsely other regions ([1,4] ms). From previous work [2] we were aware that this sensory system contains units which carry information through temporal spike precision of under 100 μs. This was the first indication that the temporal structure of some regions within a spike patterns could be much more important than in others.

4. Discussion

In this work we present a novel method for transforming neural spike trains in a manner that greatly increases confidence in our assessment of their neural coding scheme. We achieve this by quantizing the neural spike patterns while preserving as much as possible of their information measures, H or I. The neural spike train is quantized to a reproduction space of finite size. For a fixed size reproduction, the maximum entropy/information configuration is chosen to represent the spike patterns.

The information quantization method uses intrinsic similarity measures. In this way it avoids problems with biases induced by the interaction of an arbitrary distance measure with the coding scheme and preserves most of the informativeness of the original spike sequence. It can easily be used with other method for estimating information quantities [5,8] to improve their estimates.

References

[1] T. Cover, J. Thomas, Elements of Information Theory, Wiley Series in Communication, New York, 1991.
[2] A.G. Dimitrov, J.P. Miller, Natural time scales for neural encoding, Neurocomputing 32–33 (2000) 1027–1034.
[3] R.M. Gray, Entropy and Information Theory, Springer, Berlin, 1990.
[4] D.H. Johnson, C.M. Gruner, K. Baggerly, C. Seshagiri, Information-theoretic analysis of the neural code, J. Comput. Neurosci. (2001), in press.
[5] S. Panzeri, S.R. Schultz, A. Treves, E.T. Rolls, Correlations and the encoding of information in the nervous system, Proc. Roy. Soc. London B 266 (1999) 1001–1012.
[6] F. Rieke, D. Warland, R.R. de Ruyter van Steveninck, W. Bialek, Spikes: Exploring the Neural Code, The MIT Press, Cambridge, MA, 1997.
[7] C.E. Shannon, A mathematical theory of communication, Bell Sys. Technol. J. 27 (1948) 623–656.
[8] S.P. Strong, R. Koberle, R.R. de Ruyter van Steveninck, W. Bialek, Entropy and information in neural spike trains, Phys. Rev. Lett. 80 (1) (1998) 197–200.
[9] A. Treves, S. Panzeri, The upward bias in measures of information derived from limited data samples, Neural Comput. 7 (1995) 399–407.

Alexander Dimitrov is a postdoctoral research fellow at the Center for Computational Biology at Montana State University in Bozeman, Montana. He majored in Physics at St. Kliment Ohridski University of Sofia,

Bulgaria. He proceeded with his graduate studies at the University of Chicago, where he received his M.Sc. in Physics in 1993 and Ph.D. in Applied Mathematics in 1998, under the guidance of Dr. Jack D. Cowan. He is currently a postdoctoral fellow of Dr. John Miller under a personal training grant from the NIH. His main interests are in computational neuroscience and problems of neural coding.

John P. Miller is Professor of Biology, and the Director of the Center for Computational Biology at Montana State University, in Bozeman. He received his B.A. in Physics at the University of California, Berkeley, in 1972, and his Ph.D. in Biology at the University of California, San Diego, 1980. He did postdoctoral research at NIH in Bethesda, MD., with Dr. Wilfrid Rall and Dr. John Rinzel. Dr. Miller was a faculty member at U.C. Berkeley until 1997, when he moved to become the founding director of the Center for Computational Biology at MSU. His research interests include neural encoding, mechanisms of synaptic integration and system-level function. He was one of the six founding editors of the Journal of Computational Neuroscience. Along with Jim Bower of Cal Tech, Dr. Miller also established and co-organized the annual Computational Neuroscience (CNS) Meetings from 1992 until 1997. He currently serves as a member of the President's Information Technology Advisory Committee (PITAC).

Zane Aldworth is a graduate student of Dr. John Miller in the Center for Computational Biology at Montana State University, Bozeman, Montana. Zane has a B.S. in Physics from the University of Puget Sound in Tacoma, WA, and a B.S. in Biology from MSU, Bozeman. He is pursuing graduate studies as part of the NSF-sponsored IGERT program in Complex Biological Systems, with an emphasis in the neural basis of information processing.

Tomas Gedeon is an Associate Professor of Mathematics at Montana State University, Bozeman. He received his B.A. in Mathematics at Comenius University in Bratislava, Slovakia in 1989. He proceeded with his graduate studies at Georgia Institute of Technology, where he received his Ph.D. in Mathematics in 1994 under the guidance of Dr. Konstantin Mischaikow. He was a Visiting Professor at the Department of Mathematics at Northwestern University until 1995, when he moved to his current position at Montana State University. His interests include computational neuroscience, dynamical systems and neural coding.