

Traditional waveform based spike sorting yields biased rate code estimates

Valérie Ventura¹

Department of Statistics and Center for the Neural Basis of Cognition, Carnegie Mellon University, 5000 Forbes Avenue, Baker Hall 132, Pittsburgh, PA 15213

Communicated by Stephen E. Fienberg, Carnegie Mellon University, Pittsburgh, PA, March 6, 2009 (received for review June 12, 2008)

Much of neuroscience has to do with relating neural activity and behavior or environment. One common measure of this relationship is the firing rates of neurons as functions of behavioral or environmental parameters, often called tuning functions and receptive fields. Firing rates are estimated from the spike trains of neurons recorded by electrodes implanted in the brain. Individual neurons' spike trains are not typically readily available, because the signal collected at an electrode is often a mixture of activities from different neurons and noise. Extracting individual neurons' spike trains from voltage signals, which is known as spike sorting, is one of the most important data analysis problems in neuroscience, because it has to be undertaken prior to any analysis of neurophysiological data in which more than one neuron is believed to be recorded on a single electrode. All current spike-sorting methods consist of clustering the characteristic spike waveforms of neurons. The sequence of first spike sorting based on waveforms, then estimating tuning functions, has long been the accepted way to proceed. Here, we argue that the covariates that modulate tuning functions also contain information about spike identities, and that if tuning information is ignored for spike sorting, the resulting tuning function estimates are biased and inconsistent, unless spikes can be classified with perfect accuracy. This means, for example, that the commonly used peristimulus time histogram is a biased estimate of the firing rate of a neuron that is not perfectly isolated. We further argue that the correct conceptual way to view the problem out is to note that spike sorting provides information about rate estimation and vice versa, so that the two relationships should be considered simultaneously rather than sequentially. Indeed we show that when spike sorting and tuning-curve estimation are performed in parallel, unbiased estimates of tuning curves can be recovered even from imperfectly sorted neurons.

clustering | encoding | tuning properties | inconsistent | tuning information

Much of neuroscience has to do with relating neural activity and behavior: how does the brain use its neurons to produce sensory integration, motor coordination, learning, emotions, etc., and how do neurons encode parameters associated with these behaviors? Such questions have been investigated by recording brain activity during behavioral tasks to uncover associations between the two. Different aspects of brain activity are captured by different tools like functional MRI, PET, magnetoencephalography, etc. Here, we consider the spike trains of neurons provided by electrodes implanted in the brain, that is, the sequence of times at which neurons fire action potentials, or spikes. The modulation of neurons' spiking rates by behavioral or environmental covariates is widely accepted to be one way that neurons encode information about these covariates. One common measure of association between behavior and neural activity is therefore the firing rates of neurons as functions of covariates of interest, often called tuning functions or receptive fields.

The spike trains needed to calculate tuning functions are typically obtained from extra-cellular electrodes, that is, from electrodes that are positioned outside of the neurons in the tissue. The signal collected at such electrodes is typically a mixture of the activities of nearby neurons and noise, from which individual neurons' spike trains must be extracted. This extraction process is

known as spike sorting. It is one of the most important data analysis problems in neuroscience, because it has to be undertaken prior to any analysis of neurophysiological data in which more than one neuron is believed to be recorded on a single electrode.

Spike sorting is a clustering problem: neurons produce spikes that have distinct, reproducible waveforms, so that the spikes recorded at an electrode can be clustered into homogeneous groups, which presumably correspond to different neurons. Clustering techniques for spike sorting are many and range from nonparametric approaches such as k-means (1), neural networks (2), to likelihood and Bayesian model-based clustering using mixtures of distributions (3). More extensive references are provided in refs. 4 and 5.

The sequence of first spike sorting based on waveforms, then estimating tuning functions, has long been the accepted way to proceed. However, the covariates c that modulate the neurons' firing also contain information about spike identities. To see this, consider an electrode that records two neurons. Imagine that one neuron spikes only when $c_1 < c < c_2$, and the other only when $c_2 < c < c_3$, so that they never spike together. A spike recorded at the electrode will be assigned to one of the two neurons based on features of its waveform, perhaps in error if waveform clusters overlap. But we cannot make a mistake if we use c for spike sorting. Indeed, if $c_1 < c < c_2$ when a spike is detected at the electrode, then the spike must have been produced by neuron 1. If $c_2 < c < c_3$, then it is necessarily neuron 2 that spiked.

Because they ignore the information in rate modulating covariates c , current spike sorters are suboptimal. But what is more troubling is that their misclassification rates are functions of c , so that spikes are not misclassified at random. This is intuitively problematic if the goal is to estimate how neurons are modulated by c . Our first contribution is a proof that tuning functions estimated from spikes sorted based on waveforms are biased and inconsistent, unless spikes can be classified with perfect accuracy. Our second contribution is the formulation of a clustering approach that incorporates tuning information, and which yields unbiased tuning function estimates.

1. Background and Results

Consider an electrode that records I waveform generators. Generators are either neurons, or sources of noise such as static discharges, fluctuations in the local field potential, etc. When the bandpassed voltage of the electrode exceeds a chosen threshold at time t , we record a snippet of measurements a_t , which may correspond to a real spike, to noise, or to some combination of spikes and noise. Note that for simplicity and without loss of generality, all generators and their waveforms may be referred to as neurons and spikes in the rest of the article, unless noted otherwise. The

Author contributions: V.V. designed research, performed research, and wrote the paper.

The author declares no conflict of interest.

¹E-mail: vventura@stat.cmu.edu.

This article contains supporting information online at www.pnas.org/cgi/content/full/0901771106/DCSupplemental.

methods described here apply in general so the recorded waveforms a_t can also be reduced to sets of features such as principal components (PCs) (6) or wavelet coefficients (7). We denote by \mathcal{X} the set of I -dimensional binary vectors $x = (x_1, \dots, x_I)$ that give all $(2^I - 1)$ possible subsets of the I generators being active alone or approximately together to produce a suprathreshold event at the electrode. For example, when $I = 2$, x can take $2^I - 1 = 3$ values, $(1, 0) \equiv 10$, $(0, 1) \equiv 01$ or $(1, 1) \equiv 11$, which corresponds to generators 1 and 2 being active alone and together. Finally, we denote by c_t the value at time t of the covariates thought to modulate the neurons' firing rates, $\lambda_i(c_t)$, $i = 1, \dots, I$.

1.1. Background. Traditional spike sorting using waveforms. Neurons fire spikes that have characteristic waveforms, but, because the voltage of an electrode is noisy, recorded waveforms do not match their true waveforms exactly, but rather arise from distributions f_x centered around them. Either implicitly or explicitly, all spike-sorting methods assume that suprathreshold measurements a_t originate from a mixture distribution

$$f(a) = \sum_{x \in \mathcal{X}} \pi_x f_x(a), \quad [1]$$

where π_x is the proportion of events produced by generator combination x so that $\sum_{x \in \mathcal{X}} \pi_x = 1$. Eq. 1 makes no assumption. It simply states that, given a suprathreshold event at the electrode, the probability that it was produced by x is π_x , and if so, its measurement a arises from f_x . Eq. 1 can be visualized by plots of the data. For example, overlaying raw voltage measurements or plotting their PCs against one another will reveal, more or less clearly, clusters that are each a random sample of a component distribution f_x . Spike sorting effectively consists of separating these clusters, so that all spikes within each domain can be assumed to have originated from the same f_x .

Many methods exist to find cluster boundaries (4), the simplest being to draw them by hand. Another is to make explicit use of Eq. 1. First, Bayes rule yields the probability that a suprathreshold event with measurement a was produced by combination $x \in \mathcal{X}$,

$$P(x | a) = \frac{\pi_x f_x(a)}{f(a)}, \quad [2]$$

where the denominator is Eq. 1 and the numerator are its summands. The event is then assigned to the combination x^* that maximizes $P(x | a)$,

$$x^*(a) = \arg \max_{x \in \mathcal{X}} P(x | a), \quad [3]$$

with corresponding allocation for generator $i = 1, \dots, I$ the i th component of $x^*(a)$. Although it is not immediately obvious, this procedure also consists of drawing cluster boundaries. For example, the boundaries implied by Eq. 3 are linear or quadratic when the f_x are Gaussians with equal or unequal covariance matrices (8). This is illustrated in Fig. 1B.

Eq. 3 is known as the optimal classification rule, because it yields the lowest spike misclassification rate (8). The catch is that π_x and f_x must be known. How close to optimal this classification rule is in practice depends, therefore, on the validity of the models we select for f_x and on how well we can estimate them from data. At present, f_x are most commonly assumed to be Gaussians (3), although t distributions might be more suitable (9, 10).

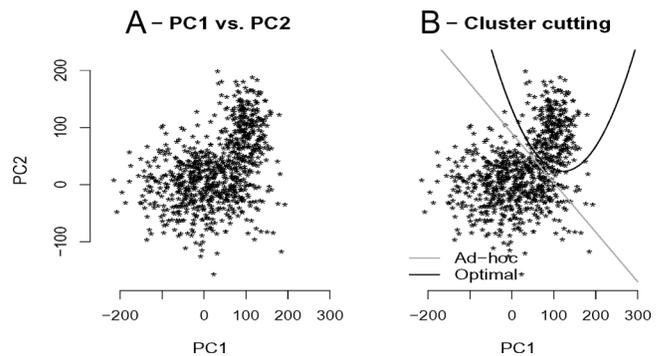


Fig. 1. Spike sorting, toy example. Two neurons are recorded by an electrode. Neuron 1 spikes only when $0 < c < 3$, neuron 2 only when $3 < c < 6$. The first two PCs of their waveforms are simulated from bivariate Gaussians: $f_{10} = N((0, 0), \begin{pmatrix} 70^2 & 0 \\ 0 & 40^2 \end{pmatrix})$, $f_{01} = N(\begin{pmatrix} 100 \\ 100 \end{pmatrix}, \begin{pmatrix} 30^2 & 0 \\ 0 & 40^2 \end{pmatrix})$, and $f_{11} \equiv 0$ since neurons do not spike together. (A) Plot of the two PCs. Each dot represents a spike. We see what could be two overlapping elliptical shaped clusters, or a single peanut-shaped cluster. (B) Cluster boundaries drawn by hand assuming there are two clusters (straight line), and by applying Eq. 3 assuming that f_x are Gaussians.

Soft or probabilistic spike assignments are seldom used for spike sorting, but we consider them because they are crucial to our results. The portion of an event with measurement a we allocate to generator i is

$$y_i^{\text{soft}} = \sum_{x: x_i=1} P(x | a), \quad [4]$$

where the sum is over all combinations x that code for generator i being active ($x_i = 1$). For example, when $I = 2$, if $P(x | a) = 0.1, 0.3$ and 0.6 for $x = 10, 01$, and 11 , respectively, soft assignments allocate $0.1 + 0.6 = 0.7$ and $0.3 + 0.6 = 0.9$ spike to generator 1 and 2, respectively, whereas hard assignments (Eq. 3) allocate one full spike to both.

Traditional estimation of tuning curves. Estimating the neurons' tuning curves $\lambda_i(c)$ involves choosing a model for $\lambda_i(c)$, and regressing the neurons' spike trains $\mathbf{y}_i = (y_{it}, t = 1, \dots, T)$ on $\mathbf{c} = (c_t, t = 1, \dots, T)$. Models for $\lambda_i(c)$ can be parametric, such as cosine functions for motor cortex data, or non-parametric, such as spline smoothers or step functions, which are commonly used to obtain peristimulus time histograms (PSTHs). Different types of regressions are appropriate for different situations. Hard assignments spike trains are binary, so a binary, e.g., logistic, regression should be used. If spike trains are first binned and resulting spike counts regressed on the covariate, as is often done to obtain a PSTH, a Poisson regression should be used. Soft spike trains $\mathbf{y}_i = \mathbf{y}_i^{\text{soft}}$ take values in $[0, 1]$, so binary regression is no longer appropriate. In that case we apply a transformation to $\mathbf{y}_i^{\text{soft}}$, which maps $[0, 1]$ onto $[-\infty, +\infty]$, e.g., logit or probit transformations, so that the transformed $\mathbf{y}_i^{\text{soft}}$ becomes amenable to ordinary regression.

Spike sorting incorporating tuning information. The inputs to any clustering procedure are vectors of features that characterize the data, e.g., waveform features in the context of spike sorting. The simplest such procedure relies on a plot of these inputs to cut clusters by hand. The same inputs yield estimates of π_x and f_x when clustering is based on Eq. 3. This was illustrated in Fig. 1B.

The information provided by the modulation of neurons by covariates c can be incorporated for spike sorting by supplementing the feature vectors with the values of c concurrent with suprathreshold events. These augmented vectors can then be used as inputs to any clustering procedure. The simplest case is illustrated in Fig. 2: features are plotted against each other and clusters can be cut by the naked eye. Just as different waveforms help identify neurons, so do the dimensions of c that modulate neurons

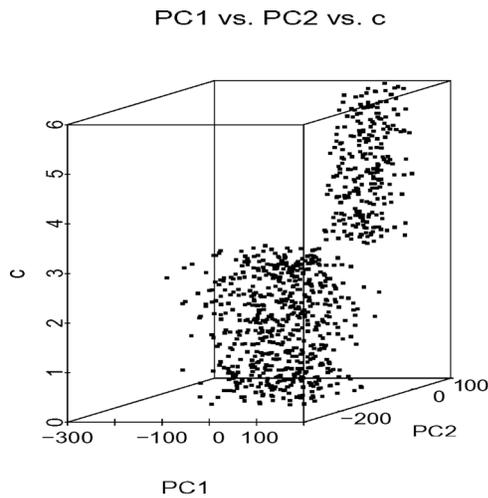


Fig. 2. Same data as in Fig. 1, but the PCs are now plotted against the rate-modulating covariate c . The two clusters that previously overlapped (Fig. 1) are now separated along the c axis. Note that better cluster separation can sometimes be achieved by using more PCs. However, the information in covariates about spikes' identities is independent of waveform information, and can be useful for spike sorting however much information waveforms provide.

most differently. In the extreme case when tuning curves share no common support, as in Fig. 2, clusters are separated perfectly. In practice, tuning curves often have common support, so clusters will overlap. In that case misclassified spikes are unavoidable, so we prefer a model-based spike sorter that minimizes misclassifications, as follows. We begin by writing the distribution of the waveforms, as in Eq. 1, but this time we condition on c . This gives

$$f(a | c) = \sum_{x \in \mathcal{X}} \pi_x(c) f_x(a | c), \quad [5]$$

where $\pi_x(c)$ is the probability that, given a suprathreshold event at the electrode when the covariates take value c , generator combination x gave rise to that event, and $f_x(a | c)$ is the distribution of its measurement a . Waveforms are characteristic of the neurons who produce them and do not depend on covariates, so we can reduce $f_x(a | c)$ to the same $f_x(a)$ used in Eq. 1.[†] We then use Bayes rule to calculate the probability that a suprathreshold event with waveform a , detected at the electrode when the covariate has value c , was generated by $x \in \mathcal{X}$,

$$P(x | a, c) = \frac{\pi_x(c) f_x(a)}{f(a | c)}. \quad [6]$$

A hard assignment allocates this spike to combination

$$x^*(a | c) = \arg \max_{x \in \mathcal{X}} P(x | a, c), \quad [7]$$

with corresponding allocation for generator i the i th component, $y_i^{\text{hard}} = x_i^*(a | c)$, while a soft assignment allocates

$$y_i^{\text{soft}} = \sum_{x: x_i=1} P(x | a, c) \quad [8]$$

to generator i , $i = 1, \dots, I$.

This approach is optimal, but requires that Eq. 5 be estimated. This involves choosing models for $f_x(a)$ and $\lambda_i(c)$, and a model

[†] Note that in Eqs. 1 and 5, we could let $f_x(a)$ depend on spiking history, to account for waveform nonstationarities such as spike amplitude decays after short interspike intervals.

for neuron dependencies, which dictates how the $\pi_x(c)$'s relate to the $\lambda_i(c)$'s. To see this, assume that composite, substantially different, waveforms are recorded whenever two generators spike within γ ms of one another. Then the probabilities that generator j contaminates a spike from generator i , $j \neq i$, are $2\gamma\lambda_j(c)$ and $1 - 2\gamma\lambda_j(c)$, respectively, with λ_j expressed in spikes per millisecond. Therefore, if an electrode records $I = 2$ independent generators, the probabilities $\pi_{10}(c)$, $\pi_{01}(c)$, and $\pi_{11}(c)$ that generators spike alone or together are proportional to $\lambda_1(c)[1 - 2\gamma\lambda_2(c)]$, $\lambda_2(c)[1 - 2\gamma\lambda_1(c)]$, and $2\gamma\lambda_1(c)\lambda_2(c)$, respectively. Similar expressions can be derived for larger I (11). If generators are dependent, $\pi_x(c) = \pi_x(c | \mathcal{H})$ are expressed as above, but the λ_i 's now depend on some aspects of spike train histories \mathcal{H} (ref. 12, and references therein). For example, if $I = 2$ neurons cannot spike within s_0 ms of each other, we could reduce spiking history to $\mathcal{H} = \{s_1, s_2\}$, with s_i the time elapsed since neuron i last spiked, and set $\lambda_i(c | s_1, s_2) = \lambda_i^*(c)$ if $s_j > s_0$ and 0 otherwise, $i = 1, 2$, $j \neq i$. Then given a suprathreshold electrode event at t , this would imply $\pi_{10}(c_t | 0, s_2) \propto \lambda_1^*(c_t)$, $\pi_{01}(c_t | s_1, 0) \propto \lambda_2^*(c_t)$, and $\pi_{11}(c_t | 0, 0) = 0$, which matches the intuition that, if neurons cannot spike together, the probability that the spike at t was fired by neuron i is proportional to its rate.

Models for f_x , λ_i , and joint spiking are the same assumptions needed to first spike sort based on Eq. 1, then estimate tuning curves, although they must now be specified all at once rather than sequentially. Note that neurons are typically, or implicitly, assumed to be independent for traditional spike sorting. Similarly, relationships between neurons are typically ignored to estimate tuning curves, unless they are of primary interest (13). A default assumption of independence is still an assumption.

The next step is to estimate Eq. 5. This might first seem impossible, because the $\pi_x(c)$'s depend on the yet unknown $\lambda_i(c)$'s. But, just as the information in waveforms can be harnessed to estimate f_x and π_x in Eq. 1 (3), so the information in the times of suprathreshold events can be harnessed to estimate $\pi_x(c)$, and therefore $\lambda_i(c)$, in Eq. 5 (11). The algorithm in ref. 11 does just that, under the assumptions that f_x are Gaussians, neurons are independent, and spike independently of the past; it accommodates parametric and nonparametric models for λ_i , and can be easily extended to allow other choices for f_x , such as t distributions. This algorithm is an exact expectation-maximization (EM) algorithm of the same type as the algorithm in ref. 3, which outputs the maximum likelihood estimates of f_x and $\pi_x(c)$. Because this algorithm is maximum likelihood based, tools for model and variable selection are readily available: the number of neurons recorded by an electrode can be determined by penalized likelihood (AIC, BIC), and models for f_x , λ_i , and the variables c that significantly modulate spiking rates can be chosen via likelihood ratio tests (12). Several of these issues are illustrated in supporting information (SI) Appendix, and more details are in ref. 11.

With Eq. 5 estimated, suprathreshold events are sorted and tuning curves estimated, as described in the previous section. Note that the estimates of λ_i , $i = 1, \dots, I$, obtained as part of the estimation of Eq. 5 correspond to the estimates obtained by regressing the soft spike trains in Eq. 8 on the covariates c . The proposed spike sorter effectively performs spike sorting and tuning function estimation simultaneously rather than sequentially.

1.2. Results. However basic or sophisticated, regressing a response variable y on covariates c always achieves the same goal: it provides an estimate of $E(Y | c)$, that is an estimate of how y varies as a function of c on average. In our context, where y_i is the spike train of neuron i after spike sorting, we regress y_i on c to estimate its tuning curve $\lambda_i(c)$. This regression therefore makes sense only if $E(Y_i | c) = \lambda_i(c)$.

In that case many suprathreshold electrode events will be noise. To illustrate this, assume that the noise on the previous electrode exceeds the threshold at a constant rate of 158 Hz, which corresponds to normally distributed noise with mean 0, SD equal to the threshold, and voltage sampled every millisecond. The electrode now records $I = 3$ generators, the third being a noise source. We simulated noise events and combined them with the previous electrode spike train. The proportion of pure noise events versus real spikes is $\pi_{001} = 66\%$ to 34% , and a significant proportion of real spikes are corrupted by noise. We simulated the PCs of pure-noise events from f_{001} , a normal distribution with mean 0 and SD 5, and assumed for simplicity that the waveform PCs of clean and noise-corrupted spikes arise from the same distributions, that is we assumed $f_{100} = f_{101}$, $f_{010} = f_{011}$, and $f_{110} = f_{111}$. This choice is not particularly realistic but does not affect our results. We spike sorted the data and estimated tuning curves; they are shown in Fig. 3D and E. The presence of a noise cluster overlapping with the clusters of real spikes results in further bias in the tuning-curve estimates when spike sorting ignores tuning information, whereas soft assignments in Eq. 8 still yield unbiased estimates.

2. Discussion

The standard paradigm in neuroscience is to perform spike sorting first, and then analyze the relationship between the spikes and the putative stimuli. We proved that when spike sorting is conducted without considering the covariates that modulate neurons spiking, estimates of their tuning functions are not consistent. We further argued that the correct conceptual way to view the problem out is to note that spike sorting provides information about rate estimation and vice versa. As a consequence, spike sorting and tuning curve estimates should be performed simultaneously rather than sequentially.

Spike sorting is a clustering problem. The traditional approach is to cluster vectors of waveform features that characterize suprathreshold events. We suggest to supplement these vectors with the covariates thought to modulate neurons spiking. Any clustering method can then be applied to these augmented vectors, although some adjustments might be needed since covariates are not physical measurements like waveforms. In particular, we showed how to adjust model-based automatic spike sorting, and proved that the resulting tuning curve estimates were unbiased. From a statistical view point, the proposed method consists of modeling the waveform measurements as a covariate varying mixture of distributions, whose mixture proportions are themselves mixtures of the unknown rates of point processes. In practice, this approach requires that models be chosen for the distributions of waveforms and joint firing rate model, and that an algorithm be available to estimate them. Such an algorithm was developed in ref. 11, under the assumption that neurons are independent of the past and of other neurons, and the waveform distributions f_x are Gaussian. The more general case is under construction, and will incorporate non-Poisson spiking behavior and nonstationarity of waveforms due to, for example, refractory periods and neurons bursting, as in ref. 14. As for modeling assumptions, they are the very same ones needed for sequential model-based spike sorting and tuning-curve estimation, no more, no less. But one legitimate concern is that the covariate-dependent and soft sorting method suggested as an alternative to pure waveform hard sorting represents a significant shift from current practices. Furthermore, waveform measurements must be saved so they can be included in the estimation of tuning properties, which is far more cumbersome than the current practice of spike sorting just once and being done with it.

Is it worth changing current practices? Tuning-curve bias can produce erroneous scientific conclusions, as illustrated by the examples in *SI Appendix*. But there will be situations where the size of the bias is not large enough, in an absolute sense, to cause concern. Additionally, there are other sources of bias and variability

that might be of greater magnitude. Such sources include the quality of chosen models for waveforms and tuning functions, and the size of the sample available for estimation of classification rules, which not only determines their variances, but also impacts how well the estimating algorithm will converge. In decoding experiments, we showed that tuning-curve estimates may be biased, but they will be consistently so if the same spike-sorting method is used for encoding and decoding. Hence, decoded trajectories will not themselves be biased. However, Fig. 3B and D suggests that tuning-curve estimates are less modulated than the true curves, which should translate into loss of efficiency. Because decoding uses many electrodes at once, the resulting aggregate effect might be substantial. More generally, the size of the bias might be of concern in studies that report results aggregated across neurons.

Determining when to implement the suggested method will require an extensive study, and the development of diagnostic tools, which is beyond the scope of this article. Our main intention here was to bring awareness to the conceptual flaw of waveform-based spike sorting, and to propose a solution.

3. Methods

Proof of Theorem 1.2: For time bins where a spike is recorded at the electrode, which we denote by $Z = 1$, soft spike assignments are such that

$$E(Y_i^{\text{soft}} | Z = 1, c) = E\left(\frac{\sum_{x:x_i=1} \pi_x(c) f_x(A)}{f(A|c)}\right)$$

with expectation with respect to the true waveform distribution. Given c , that distribution is Eq. 5, not Eq. 1. Hence,

$$\begin{aligned} E(Y_i^{\text{soft}} | Z = 1, c) &= \int \left(\frac{\sum_{x:x_i=1} \pi_x(c) f_x(a)}{f(a|c)}\right) f(a|c) da \\ &= \int \sum_{x:x_i=1} \pi_x(c) f_x(a) da = \sum_{x:x_i=1} \pi_x(c) \end{aligned}$$

since densities integrate to one. This summation is over neuron combinations x that have $x_i = 1$, hence, it is the probability that neuron i spiked, given $Z = 1$. Letting Y_i denote the true neuron spike train, we therefore have $E(Y_i^{\text{soft}} | Z = 1, c) = P(Y_i = 1 | Z = 1, c) = \frac{P(Y_i=1, Z=1|c)}{P(Z=1|c)} = \frac{P(Y_i=1|c)}{P(Z=1|c)}$, since $Y_i = 1$ implies $Z = 1$. When no spike is detected at the electrode ($Z = 0$), we set $Y_i^{\text{soft}} = 0$, so that trivially $E(Y_i^{\text{soft}} | Z = 0, c) = 0$ for all c . Then, unconditionally, $E(Y_i^{\text{soft}} | c) = E(Y_i^{\text{soft}} | Z = 1, c)P(Z = 1 | c) + E(Y_i^{\text{soft}} | Z = 0, c)P(Z = 0 | c) = P(Y_i = 1 | c)$, which is the firing rate $\lambda_i(c)$ of neuron i , expressed in units of spikes per the duration of time bins used to discretize the EST. Q.E.D.

Proof of Theorem 1.1: As above we have $E(Y_i^{\text{hard}} | c) = E(Y_i^{\text{hard}} | Z = 1, c)P(Z = 1 | c)$, which will reduce to $\lambda_i(c)$ iff $E(Y_i^{\text{hard}} | c, Z = 1) = P(Y_i = 1 | c)/P(Z = 1 | c)$. Without loss of generality we set $i = 1$, and for simplicity we work with scalar waveform measurements a , and treat the case of $I = 2$ neurons. The proof does extend generally but becomes very cumbersome. Then $E(Y_1^{\text{hard}} | c, Z = 1) = P(Y_1^{\text{hard}} = 1 | c, Z = 1) = 1 - P(Y_1^{\text{hard}} = 0 | c, Z = 1)$, which, for hard spike assignments Eq. 3, simplifies to

$$1 - P[\pi_{01} f_{01}(A) > \pi_{10} f_{10}(A) \ \& \ \pi_{01} f_{01}(A) > \pi_{11} f_{11}(A)] = 1 - \int_{a \in \mathcal{A}} f(a|c) da, \quad [9]$$

where $\mathcal{A} = \{a : \pi_{01} f_{01}(a) > \pi_{10} f_{10}(a) \ \& \ \pi_{01} f_{01}(a) > \pi_{11} f_{11}(a)\}$. The form of \mathcal{A} depends on the configuration of the f_x , $x \in \mathcal{X}$. The six possibilities are shown schematically in Fig. 4. For configuration (Fig. 4A), we have

$$E(Y_1^{\text{hard}} | c, Z = 1) = 1 - \int_{a_1}^{a_2} f(a|c) da = 1 - \sum_{x \in \mathcal{X}} \pi_x(c) [F_x(a_2) - F_x(a_1)], \quad [10]$$

where F_x is the cumulative distribution function of f_x , a_1 is such that $\pi_{01} f_{01}(a_1) = \pi_{10} f_{10}(a_1)$, and a_2 is such that $\pi_{01} f_{01}(a_2) = \pi_{11} f_{11}(a_2)$. Eq. 10 reduces to the required $\sum_{x:x_1=1} \pi_x(c)$ for all c iff $F_{01}(a_2) - F_{01}(a_1) \rightarrow 1$, $F_{10}(a_2) - F_{10}(a_1) \rightarrow 0$, and $F_{11}(a_2) - F_{11}(a_1) \rightarrow 0$. Because a_1 and a_2 are constrained by $\pi_{01} f_{01}(a_1) = \pi_{10} f_{10}(a_1)$ and $\pi_{01} f_{01}(a_2) = \pi_{11} f_{11}(a_2)$, the three limits are obtained only as all the f_x pull away from one another, so that none shares a common support. Conversely, f_x disjoint implies that a_1 and a_2 are such that $F_{10}(a_1) = 1$, $F_{01}(a_1) = F_{11}(a_1) = 0$, $F_{01}(a_2) = F_{10}(a_2) = 1$, and $F_{11}(a_2) = 0$ (this can be seen from Fig. 4A by letting the distributions spread apart), so that $F_{01}(a_2) - F_{01}(a_1) = 1$ and $F_{10}(a_2) - F_{10}(a_1) = F_{11}(a_2) - F_{11}(a_1) = 0$. Then $E(Y_1^{\text{hard}} | c, Z = 1) = 1 - \pi_{01}(c)$, which is the probability that neuron 2 does not spike, which therefore equals $\pi_{10}(c) + \pi_{11}(c) = \sum_{x:x_1=1} \pi_x(c)$, as required,

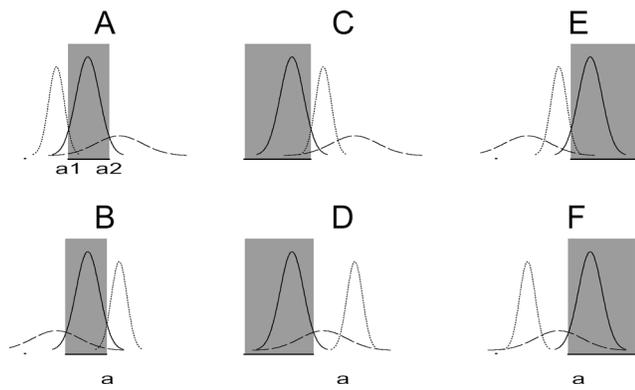


Fig. 4. Various configurations for $\pi_x f_x(a)$, $x \in \mathcal{X}$ [dotted is for neuron 1 ($x = 10$), solid for neuron 2 ($x = 01$), and large dashed for joint spikes ($x = 11$)]. The shaded areas indicate the set \mathcal{A} of values of a such that $\pi_{01}f_{01}(a) > \pi_x f_x(a)$, $x \neq 01$.

since $\sum_{x \in \mathcal{X}} \pi_x(c) = 1$. The other 5 configurations in Fig. 4 work out similarly. Therefore the common procedure of regressing Y_i^{hard} on c_t yields inconsistent firing rate estimates, unless the f_x are disjoint.

- Salganicoff M, Sarna M, Sax L, Gerstein GL (1988) Unsupervised waveform classification for multi-neuron recordings: A real-time, software-based system I Algorithms and implementation. *J Neurosci Methods* 25:181–7.
- Ohberg F, Johansson H, Bergenheim M, Pedersen J, Djupsjobacka M (1996) A neural network approach to real-time spike discrimination during simultaneous recording from several multi-unit nerve filaments. *J Neurosci Methods* 64:181–7.
- Lewicki MS (1994) Bayesian modeling and classification of neural signals. *Neural Comput* 6:1005–1030.
- Lewicki MS (1998) A review of methods for spike sorting: The detection and classification of neural action potential. *Network: Comput Neural Syst* 9:R53–R78.
- Quiroga RQ (2007) Spike sorting. *Scholarpedia* 2:3583.
- Glaser EM, Marks WB (1968) On-line separation of interleaved neuronal pulse sequences. *Data Acquisition Process Biol Med* 5:137–156.
- Letellier JC, Weber PP (2000) Spike sorting based on discrete wavelet transform coefficients. *J Neurosci Methods* 101:93–106.
- Wasserman L (2004) *All of Statistics* (Springer, New York), Chap 22.
- Harris KD, Henze DA, Csicsvari J, Hirase H, Buzsaki G (2000) Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *J Neurophysiol* 84:401–414.
- Shoham S, Fellows MR, Normann RA (2003) Robust, automatic spike sorting using mixtures of multivariate t-distributions. *J Neurosci Methods* 127:111–122.
- Ventura V (2009) Automatic spike sorting using tuning information. *Neural Comput*, in press.
- Kass RE, Ventura V, Brown EN (2005) Statistical issues in the analysis of neuronal data. *J Neurophysiol* 94:8–25.
- Okatan M, Wilson MA, Brown EN (2005) Analyzing functional connectivity using a network likelihood model of ensemble neural spiking activity. *Neural Comput* 17:1927–1961.
- Pouzat C, Delescluse M, Voit P, Diebolt J (2004) Improved spike-sorting by modeling firing statistics and burst-dependent spike amplitude attenuation: A Markov chain Monte Carlo approach. *J Neurophysiol* 91:2910–2928.

To prove that hard assignments Eq. 7 yield inconsistent estimates, all we do is rewrite the proof above with π_x replaced by $\pi_x(c)$. Then a_1 and a_2 also depend on c , which leaves the outcome of the proof unchanged.

Because all spike sorters assume, at least implicitly, that the data arise from the mixture distribution in Eq. 1, the same proof can be used to show that they yield inconsistent tuning-curve estimates, with the only change the domain of integration \mathcal{A} in Eq. 9, which must now describe formally the spike assignment rules of these spike sorters. For example, consider the spike sorter that consists of cutting clusters with the naked eye. A realistic way to do that is to assign a spike with waveform a to the neuron combination x whose mean waveform distribution is closest to a . If f_{01} and f_{10} are normal distributions with equal variance–covariance matrices, this means assigning to neuron 1 all waveforms in the set $\mathcal{A} = \{a : f_{10}(a) > f_{01}(a)\}$. Substituting this \mathcal{A} in Eq. 9 does not simplify it to the desired $\sum_{x \in \mathcal{X}} \pi_x(c)$, and thus yields the same conclusion that resulting tuning curves are inconsistent. One can proceed similarly for other spike sorters, although it might be difficult to formalize \mathcal{A} . However, despite lacking a general formal proof, we believe that it is highly unlikely for any waveform-based spike sorter to produce consistent tuning curves when waveform clusters overlap.

ACKNOWLEDGMENTS. I thank three reviewers for detailed and constructive comments, and the associate editor for support. This work was supported by National Institutes of Health Grants 2R01MH064537 and 1R01EB005847.

Supplement for : Traditional waveform based spike sorting yields biased rate code estimates

Valérie Ventura

S1 Additional theorem

Theorem S1.1. *Soft spike trains obtained from waveform based spike sorting in Eq.4 are such that*

$$E(Y_i^{soft} | c) \neq \lambda_i(c),$$

unless (1) covariates c do not modulate neurons spiking, or (2) tuning curves are proportional and neurons never spike jointly, or (3) spikes can be classified with no errors. Hence tuning curves estimated from such spike trains cannot be consistent for the true $\lambda_i(c)$, unless (1), (2), or (3) holds.

Proof: For spike assignments in Eq.4 we still have $E(Y_i^{soft} | Z = 0, c) = 0$ for all c , but

$$E(Y_i^{soft} | Z = 1, c) = \int \frac{\sum_{x: x_i=1} \pi_x f_x(a)}{f(a)} f(a | c) da \tag{S1}$$

no longer simplifies to $P(Y_i = 1 | c)/P(Z = 1 | c)$, unless $f(a | c) = f(a)$ for all a and c (Eqs.1 and 5 are equal), or spike assignments are identical under traditional and new spike sorters, that is $\forall(a, c)$ and $\forall x \in \mathcal{X}$, $\frac{\pi_x f_x(a)}{f(a)} = \frac{\pi_x(c) f_x(a)}{f(a|c)}$ (Eqs.2 and 6 are equal).

Eqs.1 and 5 are equal for all a only when (1) neurons' firing rates are not modulated by c . If (2) all neurons' tuning curves are proportional, non-constant, functions of c , i.e. $\lambda_i(c) = a_i \lambda(c)$, it may seem that tuning contains no discriminating information about neurons. However the probabilities $\pi_x(c)$ still contain information about neurons spiking alone or jointly as c varies. Hence Eqs.2 and 6 are equal only if there is no joint spiking. Indeed in that case $\pi_x(c) = 0$ for all $x \in \mathcal{X}$ such that $\sum_{i=1}^I x_i > 1$, and

$\pi_x(c) = \lambda_{x \cdot (1, \dots, I)^T}(c) / \sum_{i=1}^I \lambda_i(c) = a_{x \cdot (1, \dots, I)^T} / \sum_{i=1}^I a_i$ when $\sum_{i=1}^I x_i = 1$. Therefore $\pi_x(c)$ is a constant independent of c that is necessarily equal to π_x , so that Eqs.2 and 6 are equal. In practice, it is likely that $E(Y_i^{soft} | c)$ will be close to $\lambda_i(c)$ provided the probabilities of joint spikes are small. When (3) the distributions f_x are non-overlapping, then spikes are classified with no error, and tuning curves are unbiased since they are obtained from the true neurons' spike trains. It is also obtained from Eq.S1: when f_x have non-overlapping supports, Eqs.2 and 6 are equal for all a and c , with value 0 or 1.

S2 Additional example 1

We consider a common type of designed experiment, whose aim is to compare the firing rates of neurons under several experimental conditions. Here, we consider two conditions $c = 1$ and $c = 2$ and we assume that $I = 2$ Poisson neurons spike independently at constant rates within each condition. Neuron 1 always spikes at $\lambda_1(c) = 30\text{Hz}$, while neuron 2 does not spike when $c = 1$, and spikes at 75Hz when $c = 2$; see Fig.S1A. Without loss of generality we use only one waveform PC for spike sorting, which we simulate from normal distributions with means and variances $(6, 1)$, $(8, 1)$, and $(10.5, 3)$, for $x = 10, 01$, and 11 , respectively. The f_x overlap partially, so spike misclassifications are unavoidable. Using this model, we simulated the neurons' spike trains for 10 seconds in each condition, and concatenated them to form the electrode's train of suprathreshold events, reducing to just one event any group of events occurring within 1 msec of one another. We then simulated the PCs of these events, spike sorted the data, and estimated the neurons' firing rates. Spike classification rules and tuning curves were estimated using the algorithms in [1] for Eq.1 and in [2] for Eq.5, assuming normal distributions for the waveform PCs and a within condition constant rate model for $\lambda_i(c)$. The number of neurons I recorded by the electrode was determined by AIC. We found $I = 2$ across simulated datasets, the correct number, which is not so surprising given that we work in 1-D, the observed sample is

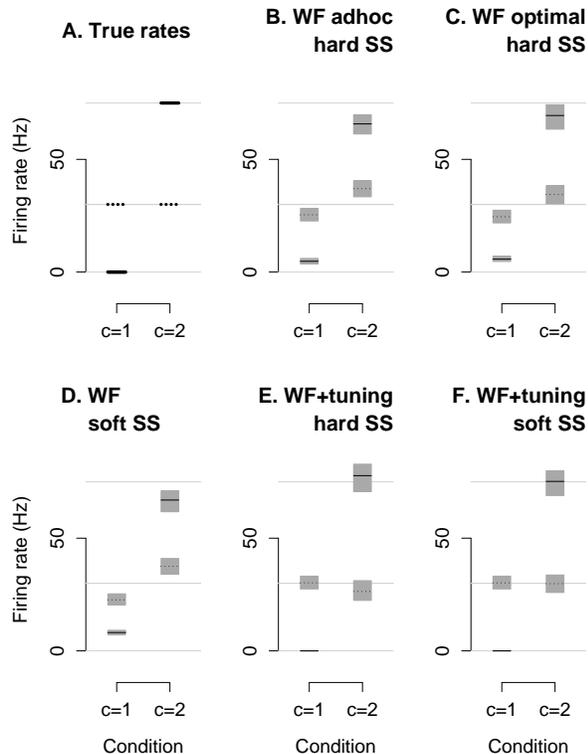


Figure S1: *True firing rates $\lambda_i(c)$ of two neurons under experimental conditions $c = 1$ and $c = 2$, and average firing rate estimates with 95% simulation bands (grey shadings), within which fall 95% of estimates in repeated simulations. The spike trains used for estimation were obtained by (B) ad-hoc waveform based spike sorting, (C) Eq.3, (D) Eq.4, (E) Eq.7, and (F) Eq.8. The estimates in (BCDE) are biased; moreover both neurons are found to be significantly modulated across conditions (p -value $\ll 0.0001$). (F) Estimated rates match the true rates on average, and only neuron 2 is correctly found to be significantly modulated across conditions.*

large (over 1000 spikes), there are only two clusters, which are sufficiently separated and of similar sizes, and we fit the correct WF distribution family to the data. To reproduce results more easily, the true classification rules could be used instead.

We repeated the simulation 100 times to calculate the mean estimated firing rates and 95% simulation bands. They are displayed in Fig.S1. Only soft spike trains in Eq.8 yield estimates that are equal to the true rates within error (Fig.S1F). Likelihood ratio tests (they reduce to the usual two-sample t-tests in this application) also confirmed that neuron 2 had significantly different rates under the two conditions ($P \ll 0.0001$) whereas neuron 1 did not ($P > 0.5\%$). In contrast, for all other spike sorting methods, t-tests determined that both neurons had significantly different rates in the two conditions ($P \ll 0.0001$).

Erroneous conclusions could matter. For example, extracellular electrodes in dorsal gan-

glia typically record the activities of several types of neurons. Muscle spindles are modulated by muscle length, but only primary afferents also encode for muscle stretch velocity. If the two conditions in Fig.S1 were to correspond to two different stretch velocities with constant muscle length, we would conclude from Fig.S1BCDE that the two neurons were primary afferents, when in fact one might be a type 2 spindle or even a cutaneous neuron. Misclassified neurons might in turn corrupt subsequent analyses.

S3 Additional example 2

Another common situation involves estimating the temporal evolution of neurons firing rates in response to a stimulus, by estimating how firing rates vary with $c_t = t$, the experimental time. Fig.S2A shows the true firing rates of $I = 2$ simulated neurons recorded by the same electrode. The experiment lasts 1 second, and the stimulus is presented at time $t = 200$ msec. One neuron has an excitatory response to the stimulus, while the other does not respond. We simulated the data for 25 repeats of the experiment, assuming the same waveform first PC distributions specified in the previous example. We estimated the spike classification rules, sorted the data, and estimated the firing rates, assuming normal distributions for the PCs and 50 milliseconds bins step functions for the firing rates.

Fig.S2 shows the average 50 milliseconds bins PSTHs we would obtain from perfectly isolated neurons and from spike trains obtained by various spike sorting options. Only soft assignments in Eq.8 (Fig.S2E) yield firing rate estimates that are similar to those obtained from isolated neurons (Fig.S2B), while traditional procedures yield estimates that are biased, and suggest that both neurons have an excitatory response to the stimulus. Estimates from Eq.7 (Fig.S2G) suggest that one neuron has an inhibitory response.

Inconsistent estimates are more than a statistical inconvenience. Indeed, a preliminary

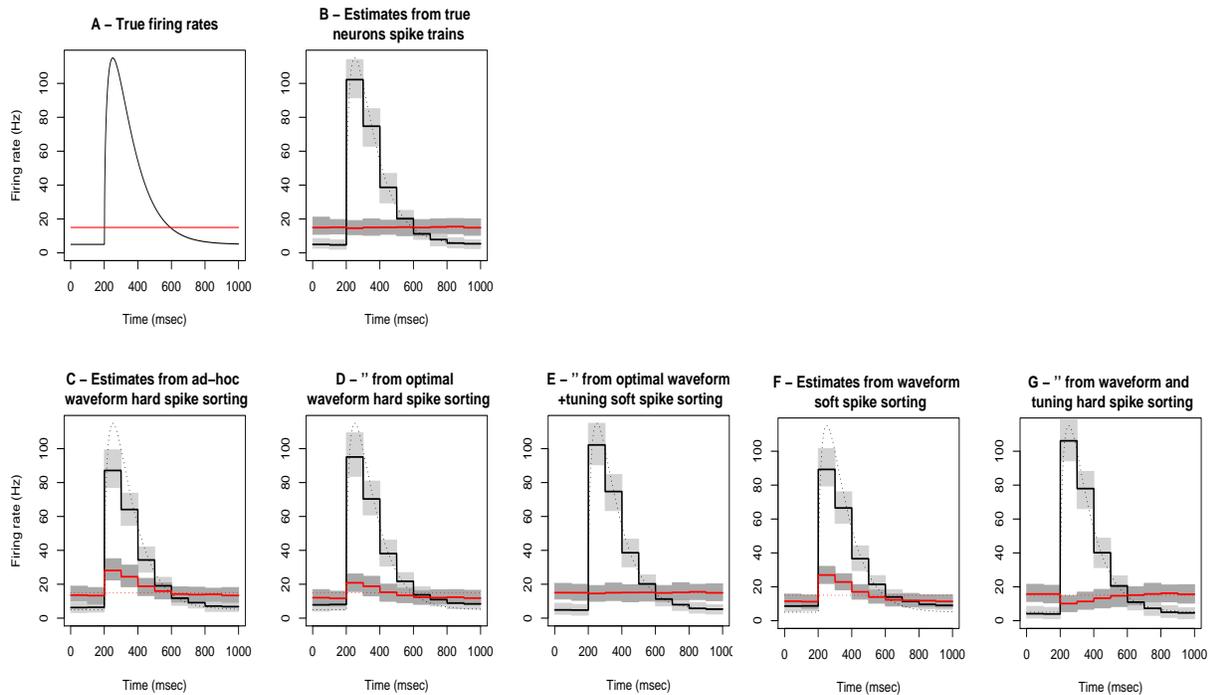


Figure S2: True firing rates $\lambda_i(t)$ of two neurons recorded by an electrode, and average PSTHs with 95% simulation bands (grey shadings), within which fall 95% of estimates in repeated simulations. True rates are overlaid in dotted. PSTHs obtained from (B) perfectly isolated neurons, (C) ad-hoc spike sorting, spike sorting according to (D) Eq.3, (E) Eq.8, (F) Eq.4, and (G) Eq.7. Only the estimates in (E) are unbiased and match the firing rate estimates obtained from the true spike trains in (B).

step in analyses of neural data often consists of grouping neurons in various functional groups, for example excitatory, inhibitory, or unresponsive neurons. Traditional waveform based spike sorting can produce mistakes in such groupings.

Note that the family of 50 msec PSTH models we fitted to the data does not include the true firing rate models in Fig.S2A. In that case firing rate estimates cannot be unbiased for the true rates, even when they are fitted to the true spike trains, as is evident in Fig.S2B. They can only be unbiased for the chosen model. This does not conflict with the theorems, whose proofs involve only true models. Had we fitted to the data a model

flexible enough to include the true model, firing rates estimated from isolated neurons or from data spike sorted according to Eq.8 would be unbiased for the true rates. Estimates from waveform based spike sorted data would not.

S4 A spike train decoding simulation

Assume we have the spike trains of motor cortical neurons, each tuned to hand velocity, and that our goal is to predict movement. Decoding of this population signal has been accomplished successfully with population vectors, linear, likelihood, Bayesian and dynamic methods. See [3] for a review of these methods. All methods involve encoding and decoding steps. Encoding consists of estimating the modulation of neurons in response to neural inputs, here velocity, while decoding uses this knowledge to predict velocity given neural activity. All decoding methods but the population vector summarize neurons modulations by their tuning curves, which, as we showed in this paper, may be biased. However tuning curves are consistently biased if we apply the same spike sorting algorithm to the spike trains used for encoding and decoding. Hence decoded trajectories will not themselves be biased. However, we remarked that biased tuning curve estimates appeared less sharply tuned than the true tuning curves, and speculated that this could degrade decoding performance. The following simulation is a first stab at investigating this speculation.

Generative model: We simulated spike trains for 80 neurons, assuming the exponentiated cosine tuning curves

$$\lambda_i(\vec{v}_t) = (k_i + m_i \vec{v}_t \cdot \vec{D}_i)^{1.5}, \quad i = 1, \dots, 80, \quad (\text{S2})$$

where k_i and m_i are positive constants determining base firing rate and directional sensitivity, \vec{D}_i is the unit-length preferred direction, and the two-dimensional velocity \vec{v}_t traces out the smooth bold path in Fig.S4 over the course of 12 seconds, with path defined

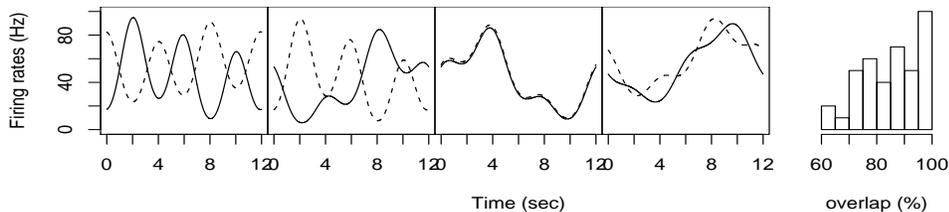


Figure S3: True firing rates $\lambda_i(\vec{v})$ for 4 pairs of neurons recorded by 4 electrodes, plotted as functions of time along the 12 second velocity path shown in Fig.S4, and histogram of rate overlap percentages for all 40 electrodes. Many pairs of neurons have similar firing rates, so there is limited information in the firing rates for spike sorting.

by $x_t = 6 \cos(\pi t/6)$, $y_t = 2 \sin(\pi t/2)$, for $t \in [0, 12]$, and velocity defined by the respective derivatives. Each neuron is assigned random values of k_i , m_i , and \vec{D}_i , so that its maximum firing rate is between 80 and 100 Hertz. Given the velocities, the spike counts are taken to have Poisson distributions with mean in Eq.S2. To create electrode spike trains, we assigned two neurons to each of 40 electrodes. Fig.S3 shows the true firing rates of the neuron pairs on 4 electrodes, plotted as functions of time along the velocity path, and a histogram of the overlap percentages between the pairs of firing rates for all 40 electrodes, where the overlap between two rates $\lambda_1(t)$ and $\lambda_2(t)$ is defined as $\left(\int_{t=0}^{t=12 \text{ sec}} \min[\lambda_1(t), \lambda_2(t)] dt \right) / \left(\int_{t=0}^{t=12 \text{ sec}} \max[\lambda_1(t), \lambda_2(t)] dt \right)$. The average overlap is 85%, and many electrodes have almost identical firing rates with overlaps close to 100%. This means that there is much less information for spike sorting in neurons tuning compared to the example of Fig.3 in the main text, for which the firing rates overlap was uncharacteristically low at 21%. For simplicity, we used the same distributions for the waveform first PCs of all neuron pairs recorded by each of the 40 electrodes. We took $f_x(a)$ to be normal distributions with means and variances (6, 1), (8, 1), and (10.5, 3), for $x = 10, 01$, and 11, respectively.

Fitting model: In implementing encoding and decoding, we assumed Poisson spike trains with exponential cosine firing rates

$$\lambda_i(\vec{v}; \theta_i) = \exp(\theta_{0i} + \theta_{1i}v_x + \theta_{2i}v_y). \quad (S3)$$

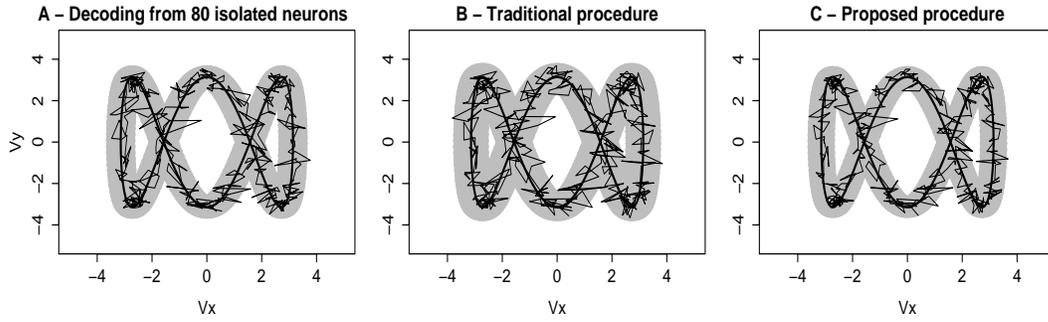


Figure S4: Trajectories decoded by maximum likelihood using (A) 80 isolated neurons and (B,C) 80 neurons recorded by 40 electrodes. (B) Traditional approach: tuning curves are estimated from data spike sorted based on waveform information only. (C) Proposed approach: spike sorting uses waveform and tuning information. The true trajectory is the smooth bold line. The gray shadings are 95% confidence bands for the true mean decoded trajectories; tighter bands mean more accurate predictions.

Generative and fitting models in Eqs.S2 and S3 are different, a realistic scenario since in real applications we are unlikely to use the *true* model. However we matched generative and fitting models for the spikes first PC distributions because our main concern here is to illustrate the effect that the addition of neurons' tuning for spike sorting has on decoding efficiency.

Spike sorting and tuning curve estimation were carried out with data simulated during 4 loops of the velocity trajectory, using the algorithm in [2]. The data were discarded, and the estimated tuning curves used for movement predictions based on fresh electrode spike trains (ESTs). These ESTs must be spike sorted prior to decoding, which cannot be done with the proposed method, since tuning information depends on the very covariate \vec{v} we aim to predict. To resolve this we used the recursive scheme of [4], which consists of spike sorting using for \vec{v} the velocity predicted at the previous time point, before a new prediction for \vec{v} can be made.

Results: Fig.S4 shows actual and decoded trajectories. Fig.S4A was obtained by using isolated neurons. In that case spike sorting is not needed and tuning curves are fitted

directly to the true neurons' spike trains. To help compare methods we plotted 95% confidence intervals for the mean decoded trajectory based on 100 repeats of the same encoding/decoding experiment. Tighter bands correspond to better decoding. The integrated squared error (ISE) also provides a more quantitative assessment of efficiency. For a particular data set, the ISE is the squared difference between the decoded and actual velocities, averaged over all time bins. Typically the ISE tends to decrease proportionally to the inverse of the number of neurons. Therefore, based on this measure, the accuracy of a method based on N_1 neurons will be comparable to the accuracy of another method based on N_2 neurons when $N_2 = N_1 \times R$, where $R = ISE_1/ISE_2$ is the ratio of the ISEs of the two methods. We thus calculated that the traditional method in Fig.S4B would require 50% more neurons to be as efficient as decoding from isolated neurons, whereas the proposed method would require 10% fewer. The gain in efficiency of the proposed method happens because the estimating algorithm alleviates some discrepancies between generative tuning curve model in Eq.S2 and the fitting model in Eq.S3, as was illustrated in [4]. Running the same simulation with the mean of f_{01} equal to 7 rather than 8, which increases the overlap between f_{10} and f_{01} from 19 to 40%, the traditional method would require twice as many neurons to be as efficient as decoding from isolated neurons. Letting the mean of f_{01} take uniform values between 6 and 11, so that f_{10} and f_{01} on each electrode can vary randomly between complete overlap and perfect separation, the traditional method was 1.5 times less efficient than decoding from isolated neurons. In contrast, the proposed method remained 10% more efficient than decoding from isolated neurons in all simulation configurations

This simulation gives evidence that including covariate information for spike sorting might improve spike train decoding. However, our simulation is not entirely fair, so we must be cautious with these results. Indeed, we compared traditional maximum likelihood decoding (Fig.S4B), with a decoding scheme that induces bias and smoothness in the decoded velocity path, because the predicted velocity at time t is used for spike sorting at time $t + 1$. Bias degrades decoding performance, while smoothness reduces variability

and so improves it. This idea was fully explored in [4]. The correct way to compare the two schemes would be to inject the same amount of smoothness in both, which can be achieved by modeling the velocity path as a random walk, and working out the full Bayesian solution for decoding the velocity. This solution is usually referred to as dynamic (Bayesian) decoding. It has been worked out in the traditional approach [3], but remains to be resolved for the proposed approach. This solution is well beyond the scope of the current paper, and is the topic of a separate upcoming manuscript.

S5 References

1. Lewicki MS (1994) Bayesian modeling and classification of neural signals. *Neural Computation* 6(5): 1005-1030.
2. Ventura V (2009) Automatic Spike Sorting using Tuning Information. *Neural Computation*, 21: 2466-2501.
3. Brockwell AE, Kass RE, Schwartz AB (2007) Statistical Signal Processing and the Motor Cortex. *Proceedings of the IEEE*, 95: 881-898.
4. Ventura V (2008) Spike Train Decoding without Spike Sorting. *Neural Computation*, 20: 923-963.