LETTER **Communicated by Emery Brown**

A Computationally Efficient Method for Incorporating Spike Waveform Information into Decoding Algorithms

Valérie Ventura
vventura@stat.cmu.edu
Sonia Todorova
Department of Statistics, Carnegie Mellon University, and Center for the Neural Basis of Cognition, Pittsburgh, PA 15213, U.S.A.

Spike-based brain-computer interfaces (BCIs) have the potential to restore motor ability to people with paralysis and amputation, and have shown impressive performance in the lab. To transition BCI devices from the lab to the clinic, decoding must proceed automatically and in real time, which prohibits the use of algorithms that are computationally intensive or require manual tweaking. A common choice is to avoid spike sorting and treat the signal on each electrode as if it came from a single neuron, which is fast, easy, and therefore desirable for clinical use. But this approach ignores the kinematic information provided by individual neurons recorded on the same electrode. The contribution of this letter is a linear decoding model that extracts kinematic information from individual neurons without spike-sorting the electrode signals. The method relies on modeling sample averages of waveform features as functions of kinematics, which is automatic and requires minimal data storage and computation. In offline reconstruction of arm trajectories of a nonhuman primate performing reaching tasks, the proposed method performs as well as decoders based on expertly manually and automatically sorted spikes.

1 Introduction

Motor brain-computer interfaces (BCIs) map neural activity to the movement of a neuroprosthetic device (Schwartz, 2007) and have the potential to restore motor ability to people with paralysis and amputation. In this letter, we focus on decoding motor intention from the activity of neurons recorded with a microelectrode array.

Decoding consists of predicting kinematic variables from spike trains. The spike trains of individual neurons are not usually observed because the electrodes on an array record the combined signals of multiple

S. Todorova is currently at Google Research, New York.

Neural Computation **27, 1033–1050** (2015)
doi:10.1162/NECO_a_00731

neurons, but they can be estimated by sorting the spike waveforms (Lewicki, 1998). Another popular choice now is to avoid spike sorting and treat each electrode as a single putative neuron, which is fast, easy, and therefore desirable for clinical use (Fraser, Chase, Whitford, & Schwartz, 2009). But this approach ignores the kinematic information provided by individual neurons recorded on the same electrode. Ventura (2009b) proposed decoding kinematics from a joint model for the electrodes' spike trains and the spike waveforms, which also avoids spike sorting but does not sacrifice kinematic information. Based on a nonparametric and a parametric implementation of that decoder, respectively, Kloosterman, Layton, Chen, and Wilson (2014) and Todorova, Sadtler, Batista, Chase, and Ventura (2014) report better accuracies to decode 1D location of rats from hippocampal place cells and 3D arm velocity from rhesus monkey M1 and PvM data, compared to decoding from unsorted electrodes and from units carefully sorted by human experts.

In this letter, we first argue that decoding from the true neurons' spike trains is the statistically most efficient approach and that decoding from the joint model for the electrodes' spike trains and the spike waveforms is the next best option. But accurate spike sorting to retrieve the neurons' spike trains requires computationally intensive algorithms, large amounts of data, and expert tuning (Harris, Henze, Csicsvari, Hirase, & Buzsáki, 2000; Gibson, Judy, & Markovi, 2012), and predictions from the joint model are not obtained in closed form but require computationally expensive numerical or stochastic approximations (Todorova et al., 2014). To transition BCI devices from the lab to the clinic, decoding must proceed automatically and in real time, which prohibits the use of the two most statistically efficient methods.

We then propose a computationally efficient implementation of the decoder based on the joint model for the electrodes' spike trains and the spike waveforms, which is automatic and yields closed-form kinematic predictions. We evaluate its performance on offline reconstruction of arm trajectories for a nonhuman primate performing reaching tasks and show that it is as efficient as decoding from spikes sorted using the state-of-the-art focused mixture model of Carlson et al. (2014).

2 Methods

Consider a neuron i whose firing rate $\lambda_i(\mathbf{x}_t)$ spikes per msec, say, is modulated by kinematic variables $\mathbf{x}_t$. The spike-generating process for this neuron is

$$Y_{it} \mid \mathbf{x}_t \sim \text{Bernoulli}(\lambda_i(\mathbf{x}_t)), \tag{2.1}$$

$$\mathbf{W}_t \mid Y_{it} = 1 \sim f_i(\mathbf{w}), \tag{2.2}$$

where $Y_{it} = 1$ if the neuron spikes at time t and $Y_{it} = 0$ otherwise, and $\mathbf{W}_t$ is a vector of its waveform measurements, with distribution f_i. An electrode records the activity of several neurons and noise. Its voltage is thresholded, and the voltage threshold crossings are recorded together with their waveforms. The electrode spike-generating process is

$$Z_t \mid \mathbf{x}_t \sim \text{Bernoulli}(\tau(\mathbf{x}_t)), \tag{2.3}$$

$$\mathbf{W}_t \mid Z_t = 1, \mathbf{x}_t \sim \sum_{j=1}^{K} \pi_j(\mathbf{x}_t) f_j(\mathbf{w}), \tag{2.4}$$

$$\mathbf{W}_t \mid Z_t = 0, \mathbf{x}_t \sim \mathbf{w}_0, \tag{2.5}$$

where $Z_t = 1$ denotes a threshold crossing at t and equation 2.4 is the distribution of the corresponding waveforms; equation 2.4 specifies that the probability that the spike was produced by unit j is $\pi_j(\mathbf{x}_t)$, with waveform features distribution f_j. The K units recorded by the electrode include a noise unit, which gathers the waveforms not attributable to any neurons. We ignore coincident spikes to simplify the exposition and the notation, in which case the threshold crossing rate in equation 2.3 is the sum of the firing rates of the K units: $\tau(\mathbf{x}_t) = \sum_j \lambda_j(\mathbf{x}_t)$, and $\pi_j(\mathbf{x}_t) = \lambda_j(\mathbf{x}_t)/\tau(\mathbf{x}_t)$. Ventura (2009a) provides the exact formula for $\tau(\mathbf{x}_t)$. Equation 2.5 is the distribution of the electrode voltage below the threshold ($Z_t = 0$), which is not typically recorded. Without loss of generality, we let it be a degenerate distribution with all its mass at $\mathbf{w}_0 = 0$.

The neuron and electrode models above assume that (i) neurons are Poisson and mutually independent, so their spiking rates depend only on the kinematics $\mathbf{x}_t$, and (ii) waveforms are stationary, so their distributions do not depend on covariates such as time t and spike trains statistics (e.g. spiking rate and interspike intervals). Distributional assumptions are addressed in Sections 2.2 and 2.3.

2.1 Statistically Efficient Decoding. The diagram in Figure 1 represents the processing pipeline for the data. The true neurons' spike trains Y provide the most information about the kinematics $\mathbf{x}$, since Y and $\mathbf{x}$ are most closely connected in the graph. But Y is unobserved and thus cannot be used for decoding. Two alternative decoding routes are commonly implemented: one uses the waveform measurements $\mathbf{W}$ to sort the signal into individual units' spike trains, $\hat{Y}$, the other ignores the waveforms altogether and treats the electrode threshold crossings Z as the spike trains of single putative neurons. A model describing the dependence between the available spike trains ($\hat{Y}$ or Z) and the kinematics is then used to decode $\mathbf{x}$ (see section 2.2). The latter route is popular in the lab because it is fast and easy to implement and performs well in practice (Fraser et al., 2009),

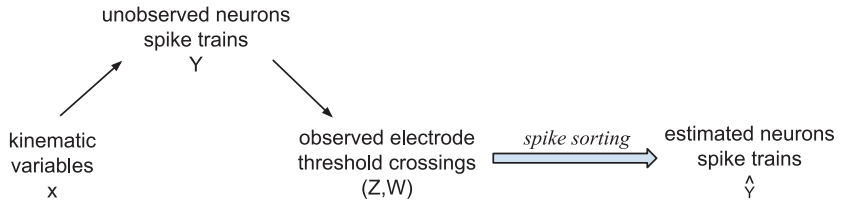

Figure 1: Observed data, $(Z, \mathbf{W})$, unobserved neurons spike trains, Y, and spike trains of sorted units, $\hat{Y}$. The data processing inequality states that $I(\mathbf{x}; \hat{Y}) \leq I(\mathbf{x}; Z, \mathbf{W})$: spike sorting the observed data $(Z, \mathbf{W})$ decreases the information about $\mathbf{x}$.

but it is statistically inefficient because it ignores the information about $\mathbf{x}$ in equation 2.4. The former route is not fully efficient either: applying the data processing inequality (Kullback, 1997) to Figure 1 suggests that $\hat{Y}$ provide less information about $\mathbf{x}$ than the unprocessed data $(Z, \mathbf{W})$, unless $\hat{Y} = Y$, that is, unless spike sorting retrieves the true neuron spike trains. Processing $(Z, \mathbf{W})$ to obtain $\hat{Y}$ discards information about $\mathbf{x}$ because the waveforms alone are used to sort spikes, which ignores that the kinematics $\mathbf{x}$ also contain information about spike identities when they modulate the neurons' firing rates (Ventura, 2009b; Ventura & Gerkin, 2012). Todorova et al. (2014) observe that decoding jointly from $(Z, \mathbf{W})$ is particularly superior to decoding from sorted spike trains $\hat{Y}$ when sorting is poor, for example when spike waveform clusters overlap and when neurons on an electrode have very different tuning curves so that the $\pi_j(\mathbf{x})$ in equation 2.4 are highly modulated. Decoding from $\hat{Y}$ or from $(Z, \mathbf{W})$ is equivalent when $\pi_j(\mathbf{x}) = \pi_j$ for all j and when the waveform model in equation 2.4 clusters spikes perfectly (see appendix A).

To summarize, decoding from the true neurons' spikes trains Y is most efficient, and decoding from the observed data $(Z, \mathbf{W})$ is the next best option. However, both options are problematic for real-time decoding: the most advanced spike sorters are computationally demanding and often require manual tuning, yet they are unlikely to retrieve exactly the true neurons' spike trains, and decoding from the joint models for $(Z, \mathbf{W})$ of Todorova et al. (2014) and Kloosterman et al. (2014) is computationally very intensive. In section 2.3, we propose a reformulation of the joint model from which decoding kinematics is computationally trivial. But first, we review the most common linear methods to decode from spike trains.

2.2 Computationally Efficient Decoding from Spike Trains. Let $\mathbf{s}_t = (s_t^1, s_t^2, \ldots, s_t^N)^T$ denote the vector of spike counts in time bin of size δ centered at t for N putative neurons, which can be either spike-sorted units

or electrodes treated as single neurons. Closed-form predictions are desirable for real-time implementations of BCIs and are obtained by using gaussian linear models for the kinematics' and neurons' spike counts. One such model is the forward filter, or reverse regression (RR), which predicts the kinematics $\mathbf{x}_t$ as linear functions of spike counts (Warland, Reinagel, & Meister, 1997):

$$x_{kt} = \delta_{k0} + \mathbf{d}_k^T \mathbf{s}_{t-\tau} + \zeta_{kt}, \quad k = 1, 2, \ldots, \tag{2.6}$$

where x_{kt} is the kth component of $\mathbf{x}_t$, $\mathbf{s}_{t-\tau}$ is the vector of spike counts lagged by τ compared to the kinematics, δ_{k0} and $\mathbf{d}_k$ are a scalar and a vector of regression coefficients, and ζ_{kt} is gaussian noise. Alternatively, optimal linear estimation (OLE; Salinas & Abbott, 1994) consists of modeling the lagged spike counts as gaussian variables with firing rates linear in the kinematics:

$$\mathbf{s}_{t-\tau} = \beta_0 + \mathbf{B}\,\mathbf{x}_t + \eta_t, \tag{2.7}$$

where β_0 and $\mathbf{B}$ are a scalar and a matrix of regression coefficients and η_t is a vector of gaussian noise, and predicting the kinematics with the maximum likelihood of $\mathbf{x}_t$ in equation 2.7. Dynamic Bayesian decoding supplements equation 2.7 with a state equation that models the smoothness of kinematic trajectories; here we use an autoregressive process of order one:

$$\mathbf{x}_t = \mathbf{A}\,\mathbf{x}_{t-1} + \epsilon_t, \tag{2.8}$$

where $\mathbf{A}$ is a matrix of coefficients and ϵ_t are gaussian perturbations. Equations 2.7 and 2.8 constitute a state-space model from which predictions for $\mathbf{x}_t$ are obtained in closed form using Kalman recursive equations (Brown, Frank, Tang, Quirk, & Wilson, 1998).

2.3 Computationally Efficient Decoding from Spike Waveforms. Decoding from the observed data $(Z, \mathbf{W})$ involves a likelihood function in two parts: the likelihood of the electrodes' spike trains in equation 2.3 can be approximated by a gaussian linear likelihood for their spike counts (see equation 2.6 or 2.7), but the likelihood of the waveform features $\mathbf{W}$ based on their distribution in equation 2.4 is neither linear nor gaussian, so kinematic predictions cannot be obtained in closed form; fitting mixture distributions like equation 2.4 and estimating K can also be difficult (Lewicki, 1998). In this section, we show how to approximate the likelihood of $\mathbf{W}$ by a set of linear equations, which are easy to fit and decode from.

We start with calculating the moments of the waveform features in the binned electrodes' spike trains:

$$\begin{aligned} E(\mathbf{W}^p \mid \mathbf{x}) &= \sum_{z=0,1} P(Z=z \mid \mathbf{x})\, E(\mathbf{W}^p \mid Z=z, \mathbf{x}) \\ &= [1-\tau(\mathbf{x})]\ \mathbf{w}_0^p + \tau(\mathbf{x})\ \left[\sum_{j=1}^{K} \pi_j(\mathbf{x})\, \mu_j^p\right] \\ &= \mathbf{w}_0^p + \sum_{j=1}^{K} \lambda_j(\mathbf{x})(\mu_j^p - \mathbf{w}_0^p), \end{aligned}$$

where $\mathbf{w}_0$ is defined in equation 2.5, and $\mu_j^p = \int \mathbf{w}^p f_j(\mathbf{w})\, d\mathbf{w}$ is the pth moment of f_j in equation 2.4, that is, the pth moment of the waveform features of the jth neuron recorded by the electrode, $p \in \mathbb{N}$. The moments $E(\mathbf{W}^p \mid \mathbf{x})$ contain information about $\mathbf{x}$ since they depend on $\mathbf{x}$. Moreover, they are linear functions of $\mathbf{x}$ when the neurons' tuning curves $\lambda_j(\mathbf{x})$ are linear in $\mathbf{x}$, a common assumption. We cannot calculate them because they are functions of K, $\lambda_j(\mathbf{x})$, and μ_j^p, which are unknown since we did not spike-sort the electrode signals, but we can estimate them by regressing the corresponding sample moments on the kinematics:

$$\overline{\mathbf{w}_{t-\tau}^p} = \gamma_0^p + \mathbf{G}^p\, \mathbf{x}_t + \delta_t^p, \tag{2.9}$$

where $\overline{\mathbf{w}_t^p}$ is the vector of the sample average of the exponentiated waveform features in time bin t,[1] γ_0^p and $\mathbf{G}^p$ are regression coefficients, τ is a lag between neural activity and kinematics, and δ_t^p is a vector of errors. The sample moments in equation 2.9 are approximately gaussian by the central limit theorem and their expectations are linear in $\mathbf{x}$, so equation 2.9 can be used as linear gaussian observation equations in addition to the spike count observation equations (see equation 2.7) when decoding using OLE or Kalman filtering, and $\overline{\mathbf{w}_{t-\tau}^p}$ can be used as additional predictors (see equation 2.6) when decoding using reverse regression. Kinematic predictions from these augmented models are obtained in closed form because they are linear and gaussian. Other waveform summaries also have linear expectations and can thus be similarly used for decoding. In particular,

[1] Beware: for the linear relationship to hold, the sample average must include $\mathbf{w}_0^p$ when no spike is observed; that is, if the spike count in the bin of size δ centered at t is s_t, then $\overline{\mathbf{w}_t^p} = \delta^{-1}\{\sum_{i=1}^{s_t} \mathbf{w}_{t,i}^p + (\delta - s_t)\mathbf{w}_0^p\}$.

letting W_i denote the ith component of the vector of waveform features $\mathbf{W}$, we can prove as above that the cross-moments $E(\prod_i W_i^{p_i} \mid \mathbf{x})$ are linear in $\mathbf{x}$ for all $p_i \in \mathbb{N}$ when neurons have linear tuning curves.

2.3.1 Illustrative Toy Example. Assume that an electrode records two neurons tuned to a 1D kinematic $x \in [0, 1]$, with $\lambda_1(x) = x$ and $\lambda_2(x) = 1 - x$. Ignoring coincident spikes, the firing rate of the electrode is $\tau(x) = 1$, a constant of x, so the unsorted electrode spike train provides no information to decode x. Assume that 1D waveform features W have gaussian distributions with mean 1 and variance 0.1 for one neuron, and mean 2 and variance 1 for the other. The regression of $\overline{w}$ on x estimates the first moment $E(W \mid x) = \lambda_1(x) \cdot 1 + \lambda_2(x) \cdot 2 = 2 - x$: it depends on x, so it provides information to decode x. The second sample moment also provides information about x since $E(W^2 \mid x) = \lambda_1(x)(1^2 + 0.1) + \lambda_2(x)(2^2 + 1) = 5 - 3.9x$ depends on x. And so on. If the 1D waveform features have equal means 1 and variances 0.1 and 1, respectively, then $E(W \mid x) = 0$ and $E(W^2 \mid x) = \lambda_1(x)(1^2 + 0.1) + \lambda_2(x)(1^2 + 1) = 2 - 0.9x$: the first moment no longer depends on x but the second moment does and thus provides information about x.

Equation 2.9 is valid for all p, but different waveform moments contribute different amounts of information about the kinematics, as illustrated in the toy example. In principle, a distribution can be summarized by all its moments and cross-moments because there exists a dual mapping between distributions and moment-generating functions (Feller, 1968). Therefore, the full set of waveform sample moments and cross-moments contains all the information in equation 2.4 about $\mathbf{x}$. But they also contain noise since they are estimated from data. Using several of them for decoding may contribute additional kinematic information, but using too many will contribute more noise than signal. How many and which to include depends on many factors, including the number of neurons recorded by each electrode, the amount of noise, and the adequacy of the models. For example, if an electrode records only one neuron, then all its waveform moments (see equation 2.9) are proportional to its spike counts (see equation 2.7), on expectation, so none provides additional information, and using any of them in the decoding model in addition to the spike counts is bound to increase the variance of the kinematic predictions. We need to select which sample moments and cross-moments to use for decoding.

2.3.2 Model Selection for Reverse Regression. To capture the kinematic information in the waveform moments, we replace the decoding model based on unsorted spike counts (see equation 2.6) by

$$x_{kt} = \delta_{k0} + \mathbf{d}_k^T \mathbf{u}_{t-\tau} + \zeta_{kt}, \quad k = 1, 2, \ldots, \tag{2.10}$$

where $\mathbf{u}_{t-\tau}$ is the vector containing the lagged spike counts $\mathbf{s}_{t-\tau}$ and a number of waveform moments and cross-moments $\overline{\mathbf{w}_{t-\tau}}$, $\overline{\mathbf{w}^2_{t-\tau}}$, etc. To balance the bias and variance of the kinematic predictions to achieve the minimum expected prediction error, we fit equation 2.10 to training data using Lasso (Tibshirani, 1996), with the prediction error estimated by 10-fold cross-validated MSE.

2.3.3 Model Selection for OLE and Kalman Filtering. Ideally, we would score all models composed of any subset of the available spike count and moment equations (equations 2.7 and 2.9), to identify the minimum prediction error model. But such an exhaustive search is prohibitively time-consuming so we apply instead a greedier added variable test (AVT) procedure. We first include in the model all unsorted spike count equations (see equation 2.7), because it is the current practice and it helps evaluate the benefit of adding waveform moment observation equations. We then add waveform moment equations sequentially only if they contain kinematic information that is not already explained by the current model, as determined by an added variable test. For example, if the spike counts $\mathbf{s}$ and first waveform moments $\overline{\mathbf{w}}$ of all electrodes are already in the model and we consider adding the second moment $\overline{w^2}$ of one of the electrode's waveform features, we regress $\overline{w^2}$ on $\mathbf{s}$ and $\overline{\mathbf{w}}$, and we test if adding the kinematics as regressors explains additional variability in $\overline{w^2}$. If it does, we add the moment equation for $\overline{w^2}$ to the decoding model. We proceed similarly for each waveform moment in turn. The size of the resulting decoding model increases with the significance level α of the added variable test. The decoding accuracy for the data analyzed in section 3 was stable for $\alpha \in [1, 10]\%$, so we used $\alpha = 1\%$ because smaller models allow faster decoding.

The AVT model selection prevents highly correlated observation equations from entering the model. The equations in the model are nevertheless correlated so it is important to estimate jointly the variance-covariance matrix of the errors (η_t, δ_t^p) in equations 2.7 and 2.9. This algorithm does not explicitly build decoding models that minimize the prediction error: better models exist. In particular, we enter the unsorted spike count equations (see equation 2.7) in the model by default, although they do not necessarily contain more kinematic information than waveform moments, as we illustrated in the toy example.

2.4 Decoding Methods Summary. We predict kinematics using the three linear decoding paradigms described in section 2.2: reverse regression (RR), OLE, and Kalman filtering (KF), which use as inputs:

- The unsorted spike counts (SC, see section 2.2).

- The unsorted spike counts and a set of waveform features sample moments $\overline{\mathbf{w}^p}$ (see equation 2.9; see section 2.3). We consider two models (plus another in appendix B):
  M1 includes all spike counts and the first three moments ($p = 1, 2, 3$) of only one feature: the waveform amplitude. No model selection is performed.
  M2 includes all spike counts, AVT selected waveform moments of order up to $p = 5$ of the four waveform features depicted in Figure 2B, and AVT selected cross-moments of order two.
- Sorted spikes counts (see section 2.2). Two expert sorters are applied:
  Manual: units are carefully sorted by a human expert using template matching.
  FMM: units are sorted using the focused mixture model of Carlson et al. (2014).

We also consider decoding from sorted spike counts together with the first moment of waveform amplitudes to assess spike sorting quality rather than decoding accuracy (see the gray box plots in Figure 2).

3 Results

We evaluate the performances of the linear models listed in section 2.4 to decode the arm velocity of a rhesus macaque in an experiment performed in Andrew Schwartz's MotorLab (Fraser & Schwartz, 2012; Todorova et al., 2014), using the neural signal recorded in ventral premotor cortex on the 71 active electrodes of a 96-electrode Utah array. Specifically, we decode velocity from unsorted spike counts alone and together with waveform moments, from units carefully sorted by a human expert and from units sorted using the state-of the-art focused mixture model (FMM) sorter of Carlson et al. (2014), which has achieved remarkable accuracies in several data sets. We use the default settings for FMM to keep the procedure automatic and thus ignore customizable options such as adjusting manually the number of clusters and aligning the waveforms.

The four waveform features we consider are the peak-to-trough amplitude, the time elapsed from peak to trough, the size of the trough, and its width at half minimum; they are depicted in Figure 2B. We standardize the features on each electrode to reduce the influence of extreme values on the fit of the moment equations (see equation 2.9) when p is large. We consider two models based on spikes and waveforms. Model M1 includes the unsorted spike counts of the 71 active electrodes and the 3×71 first three moments of a single feature, the waveform amplitude; the variables in this model are not selected in any optimal way. Model M2 aims to include more predictive waveform moments among the moments of order up to $p = 5$ for the four features, and the cross-moments of order two. We use LASSO to select these moments for RR decoding and the AVT sequential selection

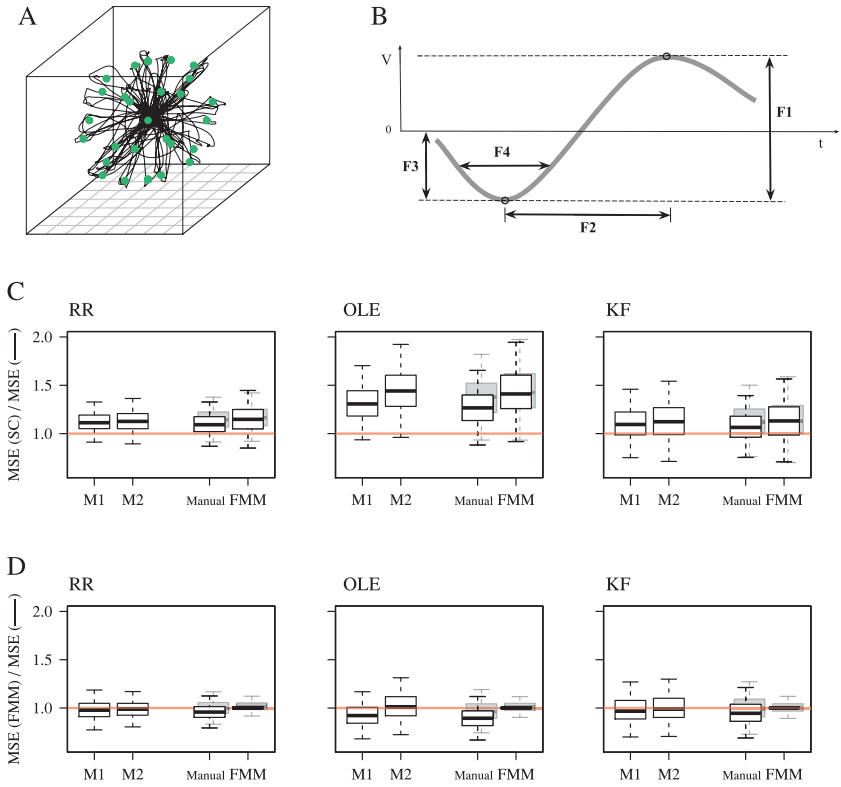

Figure 2: (A) Experimental data: Arm position over several center-out and out-center reaches to 26 targets in 3D. (B) Waveform features whose moments we consider: F1: amplitude; F2: time elapsed from peak to trough voltage; F3: size of trough; F4: width of trough at half minimum. (C) Efficiencies of decoding from unsorted spike counts and waveform moments (M1, M2) and from sorted spike counts (manual, FMM) relative to decoding from unsorted spike counts alone (SC). Adding waveform moments to unsorted spike counts improves decoding across paradigms. The shadow boxes show the efficiencies of decoding from sorted spike counts together with the waveform amplitude first moment: improvements suggest that spike sorting was not perfect. (D) Efficiencies of the same methods as in C but relative to decoding from FMM sorted neurons. Model M2 is as efficient as decoding from FMM sorted data.

procedure for OLE and KF (see section 2.3), first considering the $4 \times 71 = 284$ first moments, then the 284 second moments, the $\binom{4}{2} \times 71 = 426$ cross-moments, and finally the third, fourth, and fifth moments in sequence. The numbers of moments of each order that entered model M2 on average over all training sets are summarized in Table 1.

Table 1: Number of Waveform Moments Selected for Model M2.

	Moment type					
	1st	2nd	Cross	3rd	4th	5th
Total number	284		426	284		
Number selected for RR	136	109	162	67	24	59
Number selected for OLE and KF	124	75	46	40	17	10

We have four days of data with five sessions per day, one session containing 52 reaches to and from 26 targets arranged evenly on a virtual 3D sphere (see Figure 2A). We analyze only the portion of the reaches between movement onset and target acquisition, which sums to 8 minutes of data. We bin the spikes in 16 ms time windows, lagged $\tau = 8$ bins (128 ms) compared to arm movement, where $\tau = 8$ is the integer value in [0, 12] that maximizes the average R^2 of the electrodes' tuning curve models in equation 2.7. We use four sessions recorded on the same day (208 trials) to train all decoding models, and we decode the 52 reaches of the remaining session of that day to evaluate the various methods, initializing the decoders at the observed initial velocity for each trial. We repeat this for all combinations of four sessions in each of the five days, thereby creating $5 \times 4 = 20$ training sets and $20 \times 52 = 1040$ test trials.

We measure the accuracy of a decoded reach by its mean squared error (MSE) and the relative efficiency of two decoders by their MSE ratio. The median absolute accuracy over the 1040 test trials decoded from unsorted spike counts is 0.015, 0.031, and 0.011 m^2/s^2 for RR, OLE, and KF, respectively. The last is superior because it uses a prior kinematic model to smooth the decoded trajectories (see equation 2.8) in addition to the observation model. The white box plots in Figures 2C and 2D summarize the efficiencies of the 1040 test trials decoded from unsorted spike counts and waveform moments and from sorted spike counts, relative to decoding from unsorted counts alone (see Figure 2C), the current standard in some labs, and from state-of-the-art FMM-sorted units (see Figure 2D). Because relative efficiencies above one mean that the reference decoding method is less efficient, we conclude that (1) decoding from unsorted spike counts is least efficient; (2) decoding from model M2 is comparable to decoding from FMM-sorted units and these two methods are more efficient than all other methods; (3) model M1, which uses three moments of just one feature, with no model selection, achieves over 90% of the efficiency of the best methods; and (4) all conclusions listed here hold across all three decoding paradigms.

Experimental data are often collected carefully, with the electrode voltage thresholds chosen to collect clear units. Using more permissive thresholds, or larger electrodes, might result in recording more neurons on each

electrode, which may render extracting kinematic information from the waveforms more difficult. We investigated this hypothesis in a data set of 35 synthetic channels, each containing the combined signals of a pair of randomly selected electrodes. We reached the same conclusions as in the original data set; in particular, decoding from model M2 and from FMM-sorted spikes was comparably efficient and more efficient than decoding from unsorted spike trains. Details can be found in appendix B.

The waveform moments can also be used to judge the quality of spike sorting for the purpose of decoding. Indeed, if spike sorting retrieved the true neurons, then adding waveform moments that are modulated by the kinematics should not contribute additional kinematic information to decoders based on the sorted spike counts (see section 2.1 and appendix A). The shadow gray box plots in Figures 2C and 2D summarize the relative efficiencies of the 1040 test trials when decoding from sorted spike counts together with the first moment of the waveform amplitudes. Decoding from manually sorted units is substantially improved by adding that one waveform moment, which suggests that manual sorting was deficient. Adding the same waveform moment to FMM-sorted units increases the relative efficiency by less than 2% in median across the 1040 test trials, which suggests that FMM sorting retrieved all or almost all the available kinematic information. However, this does not necessarily imply that FMM sorting isolated the true neurons; for example, no kinematic information is lost if a well-isolated cluster of waveforms contains the spikes of several neurons that have the same tuning curves or if the spikes of a true neuron get sorted into several separate units.

Thus far, we have used waveform moments together with unsorted spike counts to improve decoding from unsorted electrodes. However, if sorted spikes are available, we could also add a selection of moments to decoding models based on sorted spikes to retrieve the kinematic information that might have been lost due to imperfect sorting.

4 Discussion

Spike-based BCI have the potential to restore motor ability to people with paralysis and amputation. We argued that it is statistically most efficient to decode from the true neurons' spike trains, and that the next best option is to decode from a joint model for the observed electrode spike trains and the waveforms (Ventura, 2009b). But accurate spike sorters are computationally demanding and are unlikely to retrieve perfectly the true neurons' spike trains from the electrode signals. And while both the parametric and nonparametric joint model implementations of Kloosterman et al. (2014) and Todorova et al. (2014) have proven valuable for decoding, they are too computer intensive to implement for real-time applications. The main contribution of this letter is a linear implementation of the decoding joint

model for the electrode spike counts and waveform features, which is fully automatic, fast to fit to training data, yields closed-form predictions, and is therefore well suited for real-time decoding. It has low storage and computational requirements: we need only collect the electrode spike trains and a few low-dimensional waveform features, and calculate spike counts and sample averages of these features. The method implicitly extracts kinematic information from individual neurons without sorting the electrode signals and avoids the explicit definition and estimation of the waveform feature distributions and the number of neurons recorded by the electrodes. We show that to reconstruct arm trajectories offline of a nonhuman primate, the proposed linear decoder outperforms decoding from unsorted spike counts alone and performs comparably to decoding from spike counts sorted using a recent state-of-the-art method.

Our method is general and should apply offline and online whenever the neural data are recorded by electrodes that capture the activities of several neurons. Its promise lies in its low computational and storage overhaul, both very important for online decoding where bandwidth may be limited and the decoder parameters must be updated often.

4.1 Model Selection. We reduced the waveforms to four shift-invariant features; better choices may exist. We considered moments of these features of order up to $p = 5$ and cross-moments of order two, and moments of split electrodes in appendix C. Other summary statistics of the waveforms could be similarly considered provided they are linearly associated with the kinematics, for example, moments of order p, with p a real number instead of an integer. We recommend including in the decoding model either a small number of low-order moments of these features (using too many will increase the variance of the predictions) or a selection of moments using Lasso for reverse regression and the sequential added variable test procedure for OLE and Kalman filtering. That procedure does not explicitly minimize the prediction error of the model, so developing one that does would be useful. Similarly, there might be more computationally efficient alternatives to Lasso with better behavior on highly correlated predictors such as the stagewise algorithm of Tibshirani (2014).

4.2 Robustness to Model Assumptions. We assumed normally distributed spike counts, and linear observation and state equations. General point process models (Barbieri et al., 2004) would be more appropriate, but we did not consider them because they do not yield kinematic predictions in closed form. To capture nonlinearities between firing rates and kinematics, the forward decoding model could be extended to an additive model of nonparametric transformations of the spike counts and waveform moments (Wagenaar, Ventura, & Weber, 2009; Warland et al., 1997), and nonparametric response transformation models could be used for the spike count and

moment equations in equations 2.7 and 2.9. To decode, we fixed every unit's temporal lag with respect to the kinematic variables to the mean estimated lag of the unsorted electrodes' tuning curves models (see equation 2.7). Using different lags can improve decoding (Wu, Gao, Bienenstock, Donoghue, & Black, 2006). Our model does not account for possible data nonstationarities, but it could be updated regularly at minimal computational cost to track potential waveform changes. We expect that more modeling refinements would not affect much the relative statistical efficiencies of the decoding methods presented here.

Appendix A

We prove that decoding from perfectly sorted neurons and from the joint model for the electrode spike trains and waveforms are equivalent either when the model in equation 2.4 isolates neurons perfectly or when $\pi_j(\mathbf{x}) = \pi_j$ for all j, because then the two models have proportional likelihoods and thus yield the same predictions.

The likelihood of a spike under the joint model is the product of the conditional and marginal distributions in equations 2.4 and 2.3; the likelihood of a sorted spike is the product of equation 2.1 over the K units. When no spike is recorded at time t, these likelihoods are proportional: $[1 - \tau(\mathbf{x})] f_0$ and $\prod_{j=1}^{K}[1 - \lambda_j(\mathbf{x})] \approx 1 - \sum_{j=1}^{K} \lambda_j(\mathbf{x}) = 1 - \tau(\mathbf{x})$, ignoring once again the coincident spikes for the sake of notational simplicity. When a spike with waveform w is observed:

- Perfect waveform separation implies $f_k(w) \neq 0$ when unit k spiked, and $f_j(w) = 0$ for all $j \neq k$. Then the likelihood of the sorted spike is $\lambda_k(\mathbf{x})(\prod_{j \neq k}[1 - \lambda_j(\mathbf{x})]) \approx \lambda_k(\mathbf{x})$, which is proportional to the joint model likelihood, $\tau(\mathbf{x})(\sum_{j=1}^{K} \pi_j(\mathbf{x}) f_j(w)) = \tau(\mathbf{x})(\pi_k(\mathbf{x}) f_k(w)) = \lambda_k(\mathbf{x}) f_k(w)$.
- $\pi_j(\mathbf{x}_t) = \pi_j$ implies $\lambda_j(\mathbf{x}) = \pi_j\, \tau(\mathbf{x})$. Then the likelihood of the sorted spike is $\lambda_k(\mathbf{x})(\prod_{j \neq k}[1 - \lambda_j(\mathbf{x})]) \approx \lambda_k(\mathbf{x})$ if it is assigned fully to neuron k; or $\prod_{j=1}^{K}[\lambda_j(\mathbf{x})]^{\alpha_j}$ with $\alpha_j = \pi_j\ f_j(w) / \sum_{k=1}^{K} \pi_k f_k(w)$ if probabilistic assignments are used (Ventura, 2009b), which reduces to $(\prod_{j=1}^{K} \pi_j^{\alpha_j})\tau(\mathbf{x})$, since $\lambda_j(\mathbf{x}) = \pi_j\, \tau(\mathbf{x})$ and $\sum_j \alpha_j = 1$. The likelihood of the joint model is $\tau(\mathbf{x}_t)(\sum_{j=1}^{K} \pi_j f_j(w))$, which is proportional to the likelihood of the sorted spike since $\tau(\mathbf{x}) \propto \lambda_k(\mathbf{x})$.
- Note that $\lambda_k(\mathbf{x}) = \pi_k\, \tau(\mathbf{x})$ means that the neurons have firing rates proportional to the electrode's, in which case sorting its spikes does not provide additional information compared to using the electrode as a putative neuron.

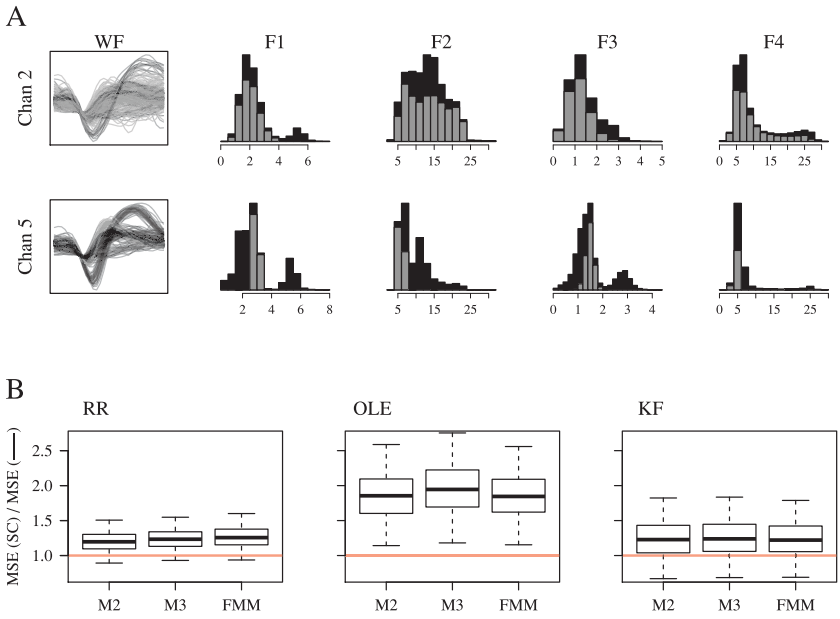

Figure 3: (A) Waveforms and features of two representative combined electrodes, each composed of the signals of two original electrodes, shown in gray and black. (B) Efficiencies of decoding from spike counts alone and together with waveform moments relative to decoding from unsorted electrodes. Adding waveform moments to unsorted spike counts improves decoding across paradigms. Models M2 and M3 are at least as efficient as decoding from expertly sorted data; M3 is slightly superior.

Appendix B: Results of Combined Electrodes

We decode from pairs of aggregated electrodes to investigate if extracting kinematic information from waveform moments is more difficult when electrodes record more units than in the original data. Because electrodes that are closer to neurons record larger waveforms and their thresholds tend to be larger, we scale the waveforms by their respective thresholds before combining electrodes. Figure 3A shows the waveforms and their four features for two representative combined electrodes. Figure 3B summarizes the efficiencies of the 1040 test trials decoded from the joint linear model M2 and from FMM-sorted spikes (we do not have manually sorted units for the combined electrodes data) relative to decoding from unsorted spike counts: the two decoders are comparably efficient and more efficient than decoding from unsorted electrodes. Figure 3B also shows the relative efficiency of the joint linear model M3, which is somewhat more efficient than M2. This model is described in appendix C.

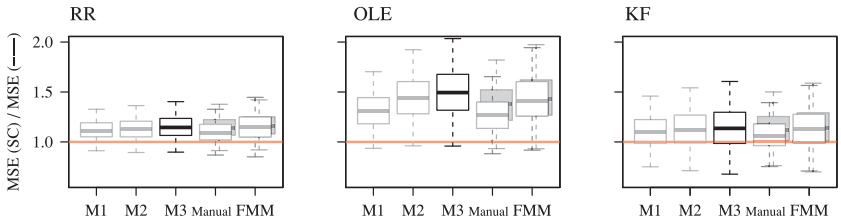

Figure 4: Efficiencies of decoding from unsorted spike counts and waveform moments (M1, M2, M3) and from sorted spike counts (Manual, FMM) relative to decoding from unsorted spike counts alone (SC). This is the same figure as Figure 2C, with the efficiencies of model M3 added: M3 is the most efficient of all decoders we investigated.

Note that the FMM sorter did not retrieve the units it identified in the original data, so the quality of spike sorting is questionable. It is likely that sorting could be improved by tweaking the parameters of the algorithm, but we did not attempt this so that the procedure would remain automatic.

Appendix C: Splitting Electrodes

We define "splitting" an electrode along the 1D waveform feature W as forming K equal-sized disjoint clusters $(\omega_{j-1}, \omega_j]$, where w_j is the $(100\,j/K)$th observed percentile of the feature in a training set. Todorova et al. (2014) used four-way splitting as a crude spike sorter for the purpose of decoding kinematics. Here we consider split sorting for this and another purpose: when electrodes capture the signals of many neurons, many waveform moments may be needed to capture the kinematic information they provide. A split can also be viewed as an electrode that records a restricted range of waveforms likely produced by a subset of these neurons and whose waveform features distribution may be well summarized by fewer moments.

To investigate this possibility, we decode our data using model M3, which includes the unsorted/unsplit spike counts by default, and selected split spike counts, unsplit and split first moments of the four features, unsplit and split second moments of the four features, unsplit cross-moments, and, finally, selected third, fourth, and fifth unsplit moments of the four features. We consider only the first two moments of the electrodes' splits to control the size of the model. We use $K = 4$ splits for each feature on all electrodes and drop one split per feature because the sum of the spike counts and moments over the four splits is proportional to the corresponding unsorted/unsplit electrode spike count and moment. Selection is carried out using Lasso for RR and AVT for OLE/KF. Figures 4 and 3 show that model M3 provides some improvement over model M2. Using $K = 8$ splits instead of 4 provided yet slightly better results, suggesting that K could be chosen to lower the prediction risk of the decoder.

References

Barbieri, R., Frank, L. M., Nguyen, D. P., Quirk, M. C., Solo, V. C., Wilson, M. A., & Brown, E. N. (2004). Dynamic analyses of information encoding in neural ensembles. *Neural Comput.*, *16*(2), 277–307.

Brown, E. N., Frank, L. M., Tang, D., Quirk, M. C., & Wilson, M. A. (1998). A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *J. Neurosci.*, *18*(18), 7411–7425.

Carlson, D., Vogelstein, J., Wu, Q., Lian, W., Zhou, M., Stoetzner, C., . . . Carin, L. (2014). Multichannel electrophysiological spike sorting via joint dictionary learning and mixture modeling. *IEEE Transactions on Biomedical Engineering*, *61*(1), 41–54.

Feller, W. (1968). *An introduction to probability theory and its applications*. New York: Wiley.

Fraser, G. W., Chase, S. M., Whitford, A., & Schwartz, A. B. (2009). Control of a brain-computer interface without spike sorting. *J. Neural Eng.*, *6*(5), 055004.

Fraser, G. W., & Schwartz, A. B. (2012). Recording from the same neurons chronically in motor cortex. *J. Neurophysiology, 107*(7), 1970–1978.

Gibson, S., Judy, J. W., & Markovi, D. (2012). Spike sorting: The first step in decoding the brain. *IEEE Signal Process. Mag.*, *29*(1), 124–143.

Harris, K. D., Henze, D. A., Csicsvari, J., Hirase, H., & Buzsáki, G. (2000). Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements. *J. Neurophysiology, 84*(1), 401–414.

Kloosterman, F., Layton, S. P., Chen, Z., & Wilson, M. A. (2014). Bayesian decoding using unsorted spikes in the rat hippocampus. *J. Neurophysiology, 111*, 217–227.

Kullback, S. (1997). *Information theory and statistics*. Mineola, NY: Courier Dover.

Lewicki, M. S. (1998). A review of methods for spike sorting: The detection and classification of neural action potentials. *Network, 9*(4), R53–R78.

Salinas, E., & Abbott, L. F. (1994). Vector reconstruction from firing rates. *J. Comp. Neuroci.*, *1*(1), 89–104.

Schwartz, A. B. (2007). Useful signals from motor cortex. *J. Physiology., 579*(Pt. 3), 581–601.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B*, *58*(1), 267–288.

Tibshirani, R. J. (2014). A general framework for fast stagewise algorithms. arXiv preprint arXiv:1408.5801.

Todorova, S., Sadtler, P., Batista, A., Chase, S., & Ventura, V. (2014). To sort or not to sort: The impact of spike-sorting on neural decoding performance. *J. Neural. Eng.*, *11*(5), 056005.

Ventura, V. (2009a). Automatic spike sorting using tuning information. *Neural Computation, 21*(9), 2466–2501.

Ventura, V. (2009b). Traditional waveform based spike sorting yields biased rate code estimates. *PNAS, 106*(17), 6921–6926.

Ventura, V., & Gerkin, R. C. (2012). Accurately estimating neuronal correlation requires a new spike-sorting paradigm. *PNAS, 109*(19), 7230–7235.

Wagenaar, J., Ventura, V., & Weber, D. (2009). Improved decoding of limb-state feedback from natural sensors. In *Proceedings of the IEEE Engineering in Medicine and Biology Society Conference* (pp. 4206–4209). Piscataway, NJ: IEEE.

Warland, D. K., Reinagel, P., & Meister, M. (1997). Decoding visual information from a population of retinal ganglion cells. *J. Neurophysiology*, *78*(5), 2336–2350.

Wu, W., Gao, Y., Bienenstock, E., Donoghue, J. P., & Black, M. J. (2006). Bayesian population decoding of motor cortical activity using a Kalman filter. *Neural Computation*, *18*(1), 80–118.

Received August 13, 2014; accepted January 5, 2015.