Ascertaining neuron importance by information theoretical analysis in motor Brain–Machine Interfaces

Yiwen Wang a, *, Jose C. Principe a, Justin C. Sanchez b

a Department of Electrical & Computer Engineering, University of Florida, Gainesville, FL, USA
b Department of Pediatrics, Neuroscience and Biomedical Engineering, University of Florida, Gainesville, FL, USA

A R T I C L E   I N F O

Article history:
Received 7 June 2009
Accepted 25 June 2009

Keywords:
Neuron importance
Kinematics decoding
Information theoretical analysis
Brain–Machine Interfaces

A B S T R A C T

Point process modeling of neural spike recordings has the potential to capture with high specificity the information contained in spike time occurrence. In Brain–Machine Interfaces (BMIs) the neural tuning characteristic assessed from neural spike recordings can distinguish neuron importance in terms of its modulation with the movement task. Consequently, it improves generalization and reduces significantly computation in previous decoding algorithms, where models reconstruct the kinematics from recorded activities of hundreds of neurons. We propose to apply information theoretical analysis based on an instantaneous tuning model to extract the important neuron subsets for point process decoding on BMI. The cortical distribution of extracted neuron subsets is analyzed and the statistical decoding performance using subset selection is studied with respect to different number of neurons and compared to the one by the full neuron ensemble. With much less computation, the extracted importance neurons provide comparable kinematic reconstructions compared to the full neuron ensemble. The performance of the extracted subset is compared to the random selected subset with same number of neurons to further validate the effectiveness of the subset-extraction approach.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

In Brain–Machine Interfaces (BMI), signal processing techniques exploit the spatial and temporal structure of neural activity to reconstruct kinematic trajectories with the purpose of bypassing spinal cord lesions (Brockwell, Rojas, & Kass, 2004; Kim et al., 2003; Sanchez et al., 2002; Wessberg et al., 2000; Wu, Gao, Bienenstock, Donoghue, & Black, 2006). While an animal performs 3D food-reaching tasks or 2D target-tracking tasks, microelectrode arrays implanted into multiple cortical areas synchronously collect neuronal activity to decode the kinematic signals (Serruya, Hatsopoulos, Paninski, Fellows, & Donoghue, 2002; Wessberg et al., 2000). These recordings contain single unit activity (spike trains), which are effectively realizations of a point process (PP) because the information is contained in the spike time occurrence. When traditional signal processing models are applied to PPs, they require time windows (or bins) to estimate the firing rate at the model input, with a loss of precision that may not model appropriately their rich neural dynamics. Alternatively, point process methods (Brown, Frank, Tang, Quirk, & Wilson, 1998; Gabbiani & Koch, 1998; Rieke, Warland, Steveninck, & Bialek, 1997; Reich, Victor, & Knight, 1998; Tuckwell, 1988; Truccolo, Eden, Fellows, Donoghue, & Brown, 2005; Zhang, Ginzburg, McNaughton, & Sejnowski, 1998) are appealing because they operate directly with spike timing, however they also require deeper neurophysiology modeling and are more computational involved (Brown, Nguyen, Frank, Wilson, & Solo, 2001; Eden, Frank, Barbieri, Solo, & Brown, 2004; Frank, Eden, Solo, Wilson, & Brown, 2002; Frank, Stanley, & Brown, 2004; Suzuki & Brown, 2005; Wang, Paiva, Principe, & Sanchez, in press).

In motor BMIs there are no precise techniques to target modulated neurons during the surgical phase. As many neurons as possible are sampled from multiple cortical areas with known motor association. These recorded neurons contribute unevenly to the movement task, with some neurons related weakly or not at all to the task. Deriving models trained on all the recorded neural spikes brings problems into computation of the decoding model. The extra degrees of freedom introduced by the redundant neuronal information would possibly lead to poor generalization. Moreover, computation grows with the number of neurons either linearly or quadratically depending upon the complexity of the decoding algorithm, which brings significant computational burden to implement BMI in low-power, portable hardware. Therefore,
quantifying neural importance from neural firings is at the core of efficient motor BMI design.

Different techniques have been proposed to evaluate neuron importance in an adaptive decoding modeling framework (Sanchez et al., 2003). One assessment is in principle dependent upon the type of decoding model chosen to predict the kinematic variables and its performance, such as sensitivity analysis and single neuron correlation analysis through a linear Wiener Filter. This analysis depends on the topology of the decoding model, therefore they are hard to interpret from neurophysiologic point of view. A contrasting approach attempts to assess neural importance independently from the decoding approach. When working with the microstructure of the neural data (spike timing) to implement Bayesian filters, the BMI observation model relates single neuron firings to kinematic variables, normally called the tuning of each neuron.

The evaluation of neural tuning importance requires an understanding of two interconnected aspects: (1) How to fully describe the neuron spikes encoding behavior; (2) What is the criterion for quantifying the quality of such neural response. For neurons located in the motor cortex, researchers developed a variety of tuning analysis approaches to describe the motor representation in neural firing (Georgopoulos, Kalaska, Caminiti, & Massey, 1982; Moran & Schwartz, 1999; Roitman, Paspalar, Johnson, & Ebner, 2005; Simoncelli, Paninski, Pillow, & Schwartz, 2004). The tuning curve was measured initially as a cosine curve between the movement direction and cortical firing in a center-out task (Georgopoulos, Lu-rito, Petrides, Schwartz, & Massey, 1989). The peak discharge rate of a cell is called preferred direction. To quantify the degree of tuning, the tuning depth has been proposed as a metric and is defined as the difference between the maximum and minimum values of the firing rates, normalized by the standard deviation of the firing rate (Carmena et al., 2003; Sanchez et al., 2003). As a scalar, the tuning depth summarizes the statistical information contained in the tuning curve to evaluate the neural representation, which indicates how modulated the cell’s firing rate is to the kinematic parameter of interest. However, this metric has some shortcomings. First, it is based on a very simple tuning model only involving the direction and may not capture other information that the neuron encodes. Secondly, the depth definition on the tuning curve is not sufficient because it can exaggerate the value when the neuron firing rate standard deviation is close to 0. Thirdly, the depth criterion is evaluated on spike rates not by point process modeling, which may exclude information in firing time and makes the depth value depend on the binning window size.

These issues with the tuning depth indicate the need to find a more principled metric to quantify tuning for neuron importance evaluation. Since the tuning depth is quantifying the information transferred from neural representations to the kinematics domain, it is relevant to utilize descriptors from information theory to estimate neural tuning. We propose mutual information to apply (Reza, 1994) as a metric for evaluating instantaneously neural receptive properties. We first test if the criterion would evaluate consistently and accurately neural tuning in a synthetic dataset. For in vivo BMI recordings, an accurate spike encoding model is necessary before applying the criterion. Researchers have developed many parametric linear/nonlinear models to describe tuning curves by a few parameters. However, there is no systematic method to completely characterize how a specific stimulus parameter governs the subsequent response of a given neuron because the tuned cells could have very different tuning properties. A Linear–Nonlinear-Poisson (LNP) model (Simoncelli et al., 2004) was proposed to capture the preferred kinematic direction and the nonlinearity of neural tuning directly from spike data. We recently proposed an extended LNP model to describe the instantaneous relationship between neuron spiking and the movement response with a causal time delay estimation (Wang, Sanchez, & Principe, 2007, submitted for publication), which allows direct implementation in a PP decoding framework (Wang et al., in press) and enables online kinematic reconstruction. Based on such a tuning model on spike train, mutual information can be applied to in vivo BMI data to extract the relevant neuron subset. The quantification and comparison of the decoding performance (kinematics estimation) as a function of number of neurons in PP sequential estimation has been reported in (Wang, Sanchez, & Principe, 2009a). However, these results are incomplete because they do not show directly the advantage of the neuron subset selection since the associated decoding models have different model orders.

It is rather difficult to quantify generalization performance across models of different sizes and a straight comparison of decoding performance amongst models of different orders is misleading. The best strategy is to match the decoding model order and select the neuron subsets with different criteria. In this paper, we further extend the comparison of decoding performance to equal neuron subset sizes, but where the neurons are chosen by mutual information or randomly from the neural ensemble pool. Both tuning model and decoding algorithm are described in Section 2. The simulation and detailed analysis on vivo BMI data to extract neuron subset with cortical distribution is presented in Section 3, followed by the evaluation and comparison in terms of the decoding performance by different subset neurons selections.

2. Data collection and methods

2.1. Data collection

The Brain–Machine Interface paradigm was designed and implemented in Dr. Miguel Nicolelis laboratory at Duke University. Chronic, neural ensemble recordings were collected from the brain of an adult female Rhesus monkey, and synchronized with task behavior. Microelectrode arrays were chronically implanted in five cortices: right dorsolateral premotor area (PMA), right primary motor cortex (MI), right primary somatosensory cortex (S1), right supplementary motor area (SMA), and the left primary motor cortex (MI). Table 1 shows the assignment of the sorted neural activity to the electrodes for different motor cortical areas.

Table 1
Assignment of the sorted neural activity to the electrode.

<table>
<thead>
<tr>
<th>Monkey1 (Left handed)</th>
<th>Right PMA</th>
<th>Right MI</th>
<th>Right S1</th>
<th>Right SMA</th>
<th>Left MI</th>
</tr>
</thead>
</table>
10 ms was selected to construct the point process observation sequence. With this interval, 94.1% of the intervals with spikes had only a single spike. For each time interval and in each channel, 1 was assigned when there were one or more spikes, otherwise 0 was assigned. 185 multi-channel spike trains were generated with a length of 1750 seconds.

The monkey performed a 2D target-reaching task to move the cursor on a computer screen with a joystick to reach the target. The monkey was rewarded when the cursor intersected the target. The corresponding position of the joystick was recorded continuously for an initial 30-min period at a 50 Hz sampling rate, referred to as the “pole control” period (Carmena et al., 2003). The velocity is estimated as the difference between the current and previous recorded positions, and the acceleration is estimated by first differences from the velocity. For fine temporal resolution, all of the kinematics are interpolated and time synchronized with the neural spike trains.

2.2. Ascertainment of neural importance based on tuning model

Based on the work by Simoncelli and colleagues (Simoncelli et al., 2004) on Linear–Nonlinear–Poisson (LNP) model, we developed an instantaneous encoding modeling for the neuron tuning function instead of performing the estimation on windows of data. Compared to the windowed approach (Simoncelli et al., 2004), our method builds a one-to-one mapping between the instantaneous kinematics vectors to the corresponding neural spiking time, which enables online decoding from spike train. Here, we briefly review the instantaneous tuning model in Fig. 1, but a full explanation can be found in (Wang et al., submitted for publication) including the model reasoning and testing.

\[
\lambda_t = f(k \cdot \mathbf{x}_{t+lag})
\]

\[
\text{spike}_t = \text{Poisson}(\lambda_t)
\]

where \(\mathbf{x}_{t+lag}\) is the instantaneous kinematics vector defined as \([p_x, v_x, a_x, p_y, v_y, a_y]\) \(1^{T+lag}\) where each vector is 2 dimensional for position, velocity, acceleration plus a bias. The setting of the state with the full kinematic vector include the information that each kinematic variable conveys for tuning, which ultimately is what is needed in BMI decoding. The lag refers to the causal time delay between motor cortical neuron activity and kinematics due to the propagation effects of signals through the motor and peripheral nervous systems, which is obtained by the approach described in (Wang et al., 2007). The linear filter projects the kinematics vector \(\mathbf{x}\) into its linear vector \(k\) (representing a preferred direction in space), which produces a scalar value that is converted by a nonlinear function \(f\) and applied to the Poisson model as the instantaneous conditional firing probability \(\lambda_t\). The Poisson model, which encodes the randomness of neural behavior, generates spike trains with an instantaneous firing probability defined by the nonlinear output. This modeling method assumes that the generation of spikes depends only on the recent stimulus and is historically independent of previous spike times.

The filter weights are obtained optimally is based on the standard technique of spike-triggered regression

\[
k = (E[\mathbf{x}^T_{t+lag} \mathbf{x}_{t+lag}] + \alpha)^{-1}E[\mathbf{x}^T_{t+lag}|\text{spike}_t][\mathbf{x}_{t+lag}]
\]

where \(E[\mathbf{x}^T_{t+lag}|\text{spike}_t][\mathbf{x}_{t+lag}]\) is the conditional expectation of the kinematic data given the spikes that mimics the role of the cross-correlation vector \(P\) between the input and the binary spike train considering a causal time delay. \(E[\mathbf{x}^T_{t+lag} \mathbf{x}_{t+lag}]\) gives the autocorrelation matrix \(R\) of the input vector considering causal time delay. The parameter \(\alpha\) is a regularization parameter to properly condition the inverse (In the experiment, \(\alpha\) is chosen to maximize the linear filter performance, experimentally set \(10^{-3}\)). Therefore, Eq. (3) reduces to a conditional expectation of the binary spike train, i.e. this linear filter gives the spike-triggered average instantaneous kinematic vector \((E_{k+lag}[\mathbf{x}] + \alpha)^{-1}\) scaled by the decorrelated kinematic data \((E[\mathbf{x}^T_{t+lag} \mathbf{x}_{t+lag}] + \alpha)^{-1}\). The optimal linear filter actually projects the multi-dimensional kinematic vectors along the direction where they differ the most from the spike-triggered kinematic vectors.

The nonlinear encoding function \(f\) for each neuron was estimated using an intuitive nonparametric technique (Chichilnisky, 2001; Simoncelli et al., 2004). Given the linear vector \(k\), the kinematics vectors after filtering become a scalar, which is represented by \(y\). The histogram of all the filtered kinematics is smoothed with a Gaussian kernel. The same procedure was repeated to draw the smoothed histogram for the outputs of the spike-triggered vectors filtered by \(k\). The nonlinear function \(f\), which gives the conditional instantaneous firing rate to the Poisson spike-generating model, was then estimated as the ratio of the two smoothed histograms. In practice, it can be implemented as a look up table in the testing phase as

\[
p(\text{spike}|y_{\text{test}}) = \frac{\sum_{i} G(y_{\text{test}} - y_{\text{spike,train}})}{\sum_{i} G(y_{\text{test}} - y_{\text{train}})}
\]

where \(G\) is the Gaussian kernel and the kernel size obeys Silverman’s rule (Silverman, 1981), \(y_{\text{test}}\) is a possible sample we generate at time \(r\) in range of the filtered test data, \(y_{\text{train}}\) is one sample in the filtered training data, and \(y_{\text{spike,train}}\) is corresponding spike-triggered sample.

Here, we are interested in ranking the neuron importance in terms of kinematics tuning, which can be defined as the information amount that the motor cortex extracts from the corresponding neural spike event. The well-established concept of mutual information (Reza, 1994) as a metric for evaluating neuron instantaneous receptive properties is based on information theory and would capture as much as possible information from the neuronal response. Define a tuned neuron as a cell that extracts more information between the linear filtered kinematics and its spiking output. The mutual information between the spike and the delayed linear filter kinematics vector is drawn simply for each neuron as

\[
I^{(j)}_{\text{spike}}(y_{\text{lag}}) = \int_{j} f_{j}(y_{\text{lag}}) \sum_{\text{spike}=0,1} p(\text{spike}^{(j)}|y_{\text{lag}}) \\
\times \log_{2} \left( \frac{p(\text{spike}^{(j)}|y_{\text{lag}})}{p(\text{spike}^{(j)})} \right) dy_{\text{lag}}
\]

where \(j\) is the neuron index, \(f_{j}(y_{\text{lag}})\) is the pdf of the linear filtered kinematics evaluated as a function of time lag. In our experiment, Eq. (5) is estimated as in

\[
I^{(j)}_{\text{spike}}(y_{\text{lag}}) = \sum_{j} p(y_{\text{lag}}) \sum_{\text{spike}=0,1} p(\text{spike}^{(j)}|y_{\text{lag}}) \\
\times \log_{2} \left( \frac{p(\text{spike}^{(j)}|y_{\text{lag}})}{p(\text{spike}^{(j)})} \right)
\]

where the pmf \(p(y)\) is estimated on 1000 points uniformly distributed within the range of filtered kinematics \(k \cdot x\) by Parzen window (Parzen, 1962) with the condition to satisfy that the summation is 1. \(p(\text{spike})\) can be calculated simply as the percentage of the spike count during the entire spike train. \(p(\text{spike}|k \cdot x)\) is exactly the
nonlinear function $f$ in LNP model. Note that vector $k$ and all the \( t \) pmfs in Eq. (6) are re-estimated for each lag to obtain the optimal time lag.

\[
\text{lag}^{*}(i) = \arg \max_{\text{lag}} T^{(i)}(\text{spk}(i); \ y_{\text{lag}}).
\]  

(7)

The corresponding mutual information maximum assigns the lag to evaluate the neuron importance in terms of the tuning information amount as

\[
\text{MIV}^{(i)} = T^{(i)}(\text{spk}(i); \ y_{\text{lag}}^{*}(i)).
\]  

(8)

We evaluate all 185 neurons recorded from different cortical area, and rank their importance according to the value of mutual information estimated by Eq. (8). Once we set a threshold for MIV, a neuron subset can be obtained for which MIV is greater than the threshold.

2.3. Kinematics decoding using the neuron subset

In the decoding process, we estimate the kinematics vector from neuron spike trains by sequential Monte Carlo (SMC) estimation on point process (Wang, Paiva, & Principe, 2007; Wang et al., in press). Sequential Monte Carlo estimation (SMCE) (Doucet, de Freitas, & Gordon, 2001) is a well-known technique for implementing a recursive Bayesian filter by Monte Carlo simulations. The key idea is to construct the posterior density function required for Bayesian estimation from a set of random samples with associated weights and to calculate the estimations based on these weighted samples. As the number of samples increases, this Monte Carlo characterization becomes an equivalent representation of the posterior density without assumptions. Here we apply this approach directly on discrete observations in the spike domain to reconstruct the kinematics states.

Given an observation interval \( (0, T] \), the neural spike train can be modeled as an inhomogeneous Poisson process characterized by its conditional intensity function \( \lambda(t|x) \), i.e. the instantaneous rate of events, defined as

\[
\lambda(t|x) = \lim_{\Delta t \to 0} \frac{\text{Pr}(N(t + \Delta t) - N(t) = 1|x)}{\Delta t}
\]  

(9)

where \( N(t) \) is the counting process of events giving the total number of neuron spikes in the interval \( (0, t] \), for \( t \in (0, T] \). \( x_t \) is the kinematic state we want to reconstruct for BMI that evolves according to the linear relation

\[
x_t = Fx_{t-1} + \eta_t
\]  

(10)

where \( F \) is the state evolution model represented by a \( D \) by \( D \) matrix, where \( D \) is the dimension of the state vector \( x_t \). It establishes the dependence on the previous state and is estimated by least square from 10000 points of training data. \( \eta_t \) is zero-mean white noise with covariance \( Q \), which is estimated from the difference between the predicted kinematics and desired ones by least square in training. The probability of observing spikes

\[
p(\Delta N_t | \lambda_t) = (\lambda_t \Delta t)^{\Delta N_t} \exp(-\lambda_t \Delta t)
\]  

(11)

where the conditional firing rate is related to the kinematics state \( x_t \) by instantaneous tuning model in (1).

Let \( \{X_{i,0}, w_i \}_{i=1}^{N_s} \) denote a random measure (Arulampalam, Maskell, Gordon, & Clapp, 2002) on the posterior density \( p(\theta|x_0; N_s) \), where \( \{X_{i,0}, i = 1, \ldots, N_s\} \) is the set of all state samples up to time \( t \) with associated normalized weights \( \{w_i, i = 1, \ldots, N_s\} \), is the sample index and \( N_s \) is the number of samples generated at each time. The posterior density of the kinematics at time \( t \) given spike observation \( N_{1:t} \) up to time \( t \) can be approximated by a weighted convolution of the samples with a Gaussian kernel as

\[
p(x_{0:t} | N_{1:t}) \approx \sum_{i=1}^{N_s} w_i \cdot G(x_{0:t} - x_{0:t}, \sigma)
\]  

(12)

where \( G(x - \bar{x}, \sigma) \) is the Gaussian kernel with mean \( \bar{x} \) and covariance \( \sigma \). According to the principle of Importance Sampling (Bergman, 1999; Doucet, 1998; Doucet et al., 2001), new samples are generated from \( p(x_0 | x_{t-1}) \) by (10) as a prediction stage. Sequential Importance Resampling (Gordon, Salmond, & Smith, 1993) is applied at every time index to avoid degeneration, so that the weight changes proportionally given by

\[
w_i(t) \propto p(\Delta N_t | \lambda_t).
\]  

(13)

Without a closed form solution for state estimation, we estimate all the information available through the posterior density of the state given the observed spike event \( p(x_t | N_{1:t}) \) at every step. In this way, any interesting moments of the pdf can be obtained, for example, Maximum A Posteriori (MAP) can be applied to get the state estimation \( \hat{x}_t \), which picks out the sample \( x_t^* \) with maximum posterior density. We can also alternatively obtain the mean value of the posterior and the error covariance by a technique called collapse (Wu, Black, Mumford, Gao, Bienenstock, & Donoghue, 2004). Here we assume that the firing rates of all the neurons are conditionally independent. We test the decoding performance using the neural subset extracted in Section 2.2.

3. Results and analysis

3.1. Simulation

We apply our information theoretical criterion on synthetic spike trains to test if it could accurately represent neuron tuning importance with different behavior/stimulus data and stochastic spike generation. Three different sets of 2D movement kinematics (velocity) are generated. The magnitude and the direction of first dataset are both uniformly distributed within the range \([0, 1] \). The second dataset has magnitude uniformly distributed while the direction is Gaussian distributed, centered at \( 2/3 \pi \) with standard deviation 0.1\pi. The third dataset has Gaussian distributed magnitude centered at 0.7 with standard deviation 0.1, and Gaussian distributed direction centered at \( 2/3 \pi \) with standard deviation 0.1\pi. The velocity signals are passed through an LNP model with the assumed nonlinear tuning function in Eq. (14):

\[
\lambda_i = \exp(\mu + \beta \cdot \vec{v} \cdot \vec{D}_{\text{prefer}})
\]  

(14)

where \( \lambda_i \) is the instantaneous firing probability, \( \mu \) is the background firing rate, \( \beta \) represents the modulation factor to a certain preferred direction, which is represented by a unit vector \( \vec{D}_{\text{prefer}} \). The spike train is generated by an inhomogeneous Poisson spike generator, once we have the knowledge of \( \lambda_i \). For a time interval, a number is randomly drawn from a normalized uniform distribution (i.e. 0 to 1) and compared with the instantaneous conditional firing probability. If the number is smaller than the probability, then a spike is generated in this time interval.

We generate each velocity dataset with 100 Hz sampling frequency and 100 second duration (10 000 samples totally) or 10 s duration (1000 samples totally) to test the reliability of the tuning criterion when there is fewer data. The background firing rate \( \mu \) is set to 0. The preferred direction is set as \( 1/3 \pi \). We implemented 10 synthetic neurons distinguished by their modulation factor \( \beta \) varying from 1 to 10, which hints at a monotonically increasing tuning. The first uniformly distributed dataset is supposed to give full perspective of the tuning curve, since it explores all possible
The average tuning information across Monte Carlo trials for different neurons:

- **Dataset 1** by tuning depth
- **Dataset 2** by tuned depth
- **Dataset 3** by information analysis

### Table 2

<table>
<thead>
<tr>
<th>Sample#</th>
<th>Method</th>
<th>Dataset 1 (%)</th>
<th>Dataset 2 (%)</th>
<th>Dataset 3 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10³</td>
<td>Traditional tuning depth</td>
<td>0.9705 ± 0.0186</td>
<td>0.9775 ± 0.0133</td>
<td>0.9911 ± 0.058</td>
</tr>
<tr>
<td></td>
<td>Information theoretical analysis</td>
<td>0.9960 ± 0.0024</td>
<td>0.9964 ± 0.0021</td>
<td>0.9988 ± 0.0008</td>
</tr>
<tr>
<td></td>
<td>t-test (p value)</td>
<td>1(0)</td>
<td>1(0)</td>
<td>1(0)</td>
</tr>
<tr>
<td>10⁴</td>
<td>Traditional tuning depth</td>
<td>0.9976 ± 0.0013</td>
<td>0.9977 ± 0.0014</td>
<td>0.9991 ± 0.0005</td>
</tr>
<tr>
<td></td>
<td>Information theoretical analysis</td>
<td>0.9997 ± 0.0002</td>
<td>0.9996 ± 0.0002</td>
<td>0.9999 ± 0.0001</td>
</tr>
<tr>
<td></td>
<td>t-test (p value)</td>
<td>1(0)</td>
<td>1(0)</td>
<td>1(0)</td>
</tr>
</tbody>
</table>

### Table 3

<table>
<thead>
<tr>
<th>Sample#</th>
<th>Method</th>
<th>Dataset 1 (%)</th>
<th>Dataset 2 (%)</th>
<th>Dataset 3 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10³</td>
<td>Traditional tuning depth</td>
<td>7</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Information theoretical analysis</td>
<td>62</td>
<td>57</td>
<td>76</td>
</tr>
<tr>
<td>10⁴</td>
<td>Traditional tuning depth</td>
<td>76</td>
<td>84</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Information theoretical analysis</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

The statistical similarity results comparison.

The comparison of percentage of Monte Carlo results in monotonically increasing.

---

Fig. 2. The average tuning information across trials by different evaluation. The line group in upper plot is the tuning information estimated by traditional tuning depth for all 3 datasets. The line group in bottom plot is the tuning information estimated by information theoretical analysis for all 3 datasets.

Direction angles. The Gaussian distributed direction in the second dataset favors samples at a certain direction. It will not change the information about the tuning curves in terms of direction angle when compared to the first dataset. The third dataset have the Gaussian distribution magnitude with center at 0.7, which means for given direction angle the instantaneous firing probability is higher than the uniformly distributed magnitude with mean at 0.5. Since randomness is involved in the generation of the velocity and spike trains, we will evaluate the tuning depth criterion for 100 Monte Carlo trials.

Fig. 2 shows the average tuning information with standard deviation across 100 Monte Carlo trials evaluated for 10 neurons with 100 s duration. The line group in upper plot is the tuning information estimated by traditional tuning depth for all 3 datasets. In order to get the statistical evaluation between Monte Carlo runs, the traditional tuning depth were not normalized to [0, 1] for each realization as normally done in real data. The line group in bottom plot is the tuning information estimated by information theoretical analysis for all 3 datasets. Both groups show higher information amount evaluated for each neuron from dataset 3 than the other 2 datasets as expected. However, the 2 lines evaluated from dataset 1 and dataset 2 are grouped much closer, which means less bias introduced by prior distribution is present in the information theoretical analysis than the traditional tuning depth. Since more samples on a certain direction angle should not affect the information amount, the information theoretical analysis provides the estimation that makes more sense.

The tuning criterion is expected to represent consistently the tuning information across different Monte Carlo trials. However, directly comparing the standard deviation for each neuron through Monte Carlo trials between 2 methods is not fair, since their scales are quite different. We use correlation coefficient to measure the similarity of the tuning information curves along 10 neurons between each trial and the average performance. The statistical results through 100 trials for 3 datasets evaluated by 2 methods with both durations are shown in Tables 2 and 3. For each dataset, pair-wise student t-test was performed to see if the results are statistically different from the traditional tuning depth. The test is performed against the alternative specified by the left tail test $C_{tuning depth} < C_{information theoretical Analysis}$. All the tests are performed on the null hypothesis at $\alpha = 0.01$ significance level. Under the null hypothesis, the probability of observing a value as extreme or more extreme of the test statistic, as indicated by the $p$-value, is also shown in Table 2.

For each dataset, the tuning information criterion by information theoretical analysis shows consistent information representation with higher correlation and less standard deviation in terms of similarity to average tuning information. All the $t$-test confirms the statistical performance improvement. In the real data analysis, there is no guarantee that we always have sufficient data to estimate neural tuning. However, note that with less duration (1000 samples), the information theoretical criterion performs better than the traditional one.

To distinguish the 10 neurons, we expect the criterion will be able to accurately rank the neurons monotonically related to the modulation factor $\beta$ from 1 to 10, even for the single realization of the spike train. Throughout the 100 Monte Carlo trials, the monotonicity of the tuning depth along 10 neurons for 3 datasets by both methods for the two durations is shown in Table 3. For example, among 100 Monte Carlo trials on 1000 sample simulation, only 7 trials show monotonicity by traditional tuning depth, while 62 trials show monotonicity by information theoretical analysis.

Note that the traditional tuning depth shows much poorer monotonicity for all the dataset compared to information theoretical analysis. It even fails on monotonicity test in dataset 3. This is because the normalization term in the traditional tuning depth (the standard deviation of the firing rate) is exponentially increasing when both modulation factor $\beta$ and mean of speed increases. When there are enough data, all the datasets show 100% monotonicity of tuning information across the 10 neurons evaluated by the information theoretical analysis. Even with insufficient data, the information theoretic tuning again shows much greater monotonicity percentage than the traditional tuning depth. Thus the information theoretical tuning depth is more reliable to rank neurons.

3.2. In vivo neural recordings

For in vivo neural recoding of BMI data, we rank all 185 neurons according to the mutual information value (herein MIV) in (8)
on 10,000 training data samples in Fig. 3. The plot shows that the amount of information carried by neurons with respect to kinematics decreases steadily across the neural ensemble. Notice that there is an inflection point in this curve, separating neurons with much higher MIV than the others, and this is a reasonable point to set our threshold. For this example, the first 30 most tuned neurons could be selected as candidates to decode the movements in the BMI model. The cortical distribution of the selected neuron is shown in Fig. 4. Here the 5 different cortical areas are shown in different color bar with the corresponding mutual information estimated by (5)–(8). The selected 30 neurons are labeled as red stars. This threshold selects 1 neuron in PMA, 21 neurons in M1, 6 neurons in S1, and 2 neurons in SMA. The most tuned neurons are in M1 as expected.

We test the neuron subset extraction by comparing the decoding performance with the full neuron ensemble. Fig. 5 shows the reconstructed kinematics using sequential estimation on point process from all 185 neurons and selected 30 neurons for 1000 testing samples. The left and right panels depict respectively the reconstructed kinematics for x-axis and y-axis. The 3 rows of plots from top to bottom display respectively the reconstructed position, the velocity and the acceleration. In each subplot, the red dotted line indicates the desired signal, the blue solid line indicates the kinematics estimation by selected 30 neurons, and green dashed line indicates the estimation by 185 neurons. We can see that both approaches could follow the desired signal and provide comparable decoding performances.

We then explore the reconstruction performance measured by NMSE from different neural subset sizes, which have 60, 40, 30, 20, 10 most important neurons associated with the movement, comparing to the decoding results by the full ensemble of 185 neurons. We assume in this analysis that each model is equally well trained, which may not be true since we are using the same training dataset with models of different number of parameters. However,
Table 4

<table>
<thead>
<tr>
<th>Neuron subset</th>
<th>Position x,y</th>
<th>Velocity Ax,Ay</th>
<th>Acceleration x,y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Ensemble</td>
<td>0.5628 ± 0.1861, 0.9643 ± 0.3222</td>
<td>0.5145 ± 0.1259, 0.8022 ± 0.2555</td>
<td>0.5097 ± 0.1261, 0.4985 ± 0.0940</td>
</tr>
<tr>
<td>60</td>
<td>0.5339 ± 0.1908, 0.8925 ± 0.2256</td>
<td>0.5031 ± 0.1345, 0.4985 ± 0.1440</td>
<td>0.5098 ± 0.1098, 0.5044 ± 0.1039</td>
</tr>
<tr>
<td>40</td>
<td>0.5339 ± 0.1818, 0.8003 ± 0.1585</td>
<td>0.5173 ± 0.1385, 0.4985 ± 0.1440</td>
<td>0.5044 ± 0.1039, 0.5044 ± 0.1039</td>
</tr>
<tr>
<td>30</td>
<td>0.5339 ± 0.2047, 0.8022 ± 0.2555</td>
<td>0.5145 ± 0.1259, 0.8022 ± 0.2555</td>
<td>0.5097 ± 0.1261, 0.5097 ± 0.1261</td>
</tr>
<tr>
<td>20</td>
<td>0.5828 ± 0.1858, 0.7273 ± 0.2674</td>
<td>0.5334 ± 0.1538, 0.7273 ± 0.2674</td>
<td>0.4915 ± 0.1124, 0.7273 ± 0.2674</td>
</tr>
<tr>
<td>10</td>
<td>0.5770 ± 0.2167, 0.7304 ± 0.3408</td>
<td>0.5697 ± 0.1550, 0.7304 ± 0.3408</td>
<td>0.5348 ± 0.1502, 0.7304 ± 0.3408</td>
</tr>
</tbody>
</table>

it tells about their relative performances and may elucidate how important the selection of the threshold is to select the number of most informative neurons. The statistical performance evaluated by NMSE with an 8 s window with 50% overlap on 8000 test data are shown in Table 2 with the corresponding standard deviation. The results are also plotted in Fig. 6.

With only 30 selected neurons (bolded row in Table 4) out of the full ensemble 185 (italic row in Table 4), we could achieve similar or even better performance in terms of NMSE. This means that not all the neurons in motor cortex are closely related to the movement task. Some of the neurons’ activities might contribute as noise for the task, which reduces the decoding performance. At the same time, computation with only 30 neurons saves 84% running time compared to the full 185 neural set. Moreover, the threshold does not seem to be critical for the x direction where the fitting performance is better since after 20 neurons and until 60 neurons, the performance is basically the same. The interpretation for the y direction is more difficult, perhaps because the model fitting is much poorer. In this direction smaller models tend to perform better, as we can expect from learning theory.

However, these results do not show clearly the advantage of the MIV selection because we are not comparing models of the same size. Therefore, we further validate the MIV selection criterion by comparing the decoding performance with random selected subsets of 30 neurons (“random subset” for short). Fig. 7 shows the distribution of 30 neurons randomly chosen for 50 trials (y-axis) across all the neuron ensembles. For each trial, the neurons that also appear in the MIV subset are labeled by red crosses. We can see from the figure that the random chosen neuron subsets are evenly distributed across all the cortical areas.

In Table 4, the statistical performance of the MIV subset is evaluated in test data built from 19 segments (50% overlap) within a total 8000 samples. First we use the random neuron subset generated from each trial to explore the reconstruction performance in each segment for each of the kinematic variables (Fig. 8). In each subplot, the red star indicates the performance with the MIV subset. The blue cross indicates the average performance by random subsets across trials for each segment, and the error bar size indicates the standard deviation. We can see that for the x direction, the performance of the MIV subset is smaller than the random subset for position, velocity and acceleration. Most of the time (13 out of 19) it is smaller than one standard deviation for position and acceleration. For the y direction where fitting is worse, the results are not as clear, with some segments showing better performance in position for the random subset. For velocity, the MIV subset selection is better and for acceleration the two methods are equivalent. This means that our neural tuning approach which incorporates information from the x and y coordinates to derive neural tuning may not be warranted, i.e. neural tuning for x and y coordinates may be necessary for better results. Alternatively, it
may simply mean that neural tuning for the y direction is much shallower and so any combination is equally valid.

Fig. 9 shows the distribution of NMSE decoded by the MIV subset and by random subsets across trials and segments. The left and right panels depict respectively the reconstructed kinematics for x-axis and y-axis. The 3 rows of plots from top to bottom display respectively for the position, the velocity and the acceleration. In each subplot, the red bar indicates the kernel smoothed histogram of the NMSE reconstructed by MIV subset, and the mean value is marked by a red star. The cyan bar indicates the kernel smoothed histogram of the NMSE reconstructed by 50 random subsets, and the mean value is marked by a blue circle. Note that the NMSE by MIV subset has a peakier distribution than the one from random subsets for most kinematics. It is unclear why the MIV varies so much from segment to segment, but it may indicate that the tuning is changing. In terms of averaged performance across segments, which is shown together with the standard deviation in Table 5, NMSE reduces dramatically 32% for position along x, 30% for velocity along x and y, and 15% for acceleration along x, but increases a little for position along y (4%) and along acceleration along y (2%). This was not expected and may reflect the fact that the instantaneous tuning considers all the kinematic variables from x and y directions. In case one of the directions overpowers the other, the tuning computation may not be equally accurate in each direction.

Therefore, we calculated the averaged NMSE (aNMSE) of all six kinematic variables as a scalar across segments of test data (an 8 s window with 50% overlap on the same test data) reconstructed by each neuron subset. Fig. 10 plots the statistical performance of the aNMSE across 50 trials of random subset selections, where the mean of aNMSE is marked by a blue cross with error bar indicating the standard derivation, comparing to the one by the MIV subset, marked by red star, across different segments of test data. We can see that for most segments, the MIV subset reconstructs better the trajectory than the random subsets. But for segment 4, the MIV subset performs almost equivalently as the average performance by random subsets, and even a little bit worse for segment 8. This may be due to the fact that our approach assumes that neural importance for decoding does not change over the task execution. However, it is known that neurons vary their involvements in a given task over time (Carmena et al., 2003; Fetz, 1992). The worse performance by MIV subset on some segments indicates that some important neurons for those segments may be excluded.

Fig. 8. Statistical performance (NMSE) by random subsets across 50 trials vs. MIV subset performance along different segment of data. The left and right panels depict respectively the NMSE of reconstructions for x-axis and y-axis. The 3 rows of plots from top to bottom display respectively the position, the velocity and the acceleration. The x-axis refers different segments of data. The mean of the performance by random subsets across trials is marked by blue cross, and the error bar size indicates the standard derivation. The performance by MIV subset is marked by red stars. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
Fig. 9. The distribution of NMSE decoded by MIV subset and random subsets. The left and right panels depict respectively the reconstructed kinematics for $x$-axis and $y$-axis. The 3 rows of plots from top to bottom display respectively for the position, the velocity and the acceleration. In each subplot, the red bars indicate the kernel smoothed histogram of the NMSE reconstructed by MIV subset, and the mean value is marked by a red star. The cyan bars indicate the kernel smoothed histogram of the NMSE reconstructed by random subsets, and the mean value is marked by a blue circle. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 5
Statistical performance of the kinematics reconstruction error (NMSE) by MIV subset vs. random subsets.

<table>
<thead>
<tr>
<th>Neuron subset</th>
<th>Position x</th>
<th>Position y</th>
<th>Velocity x</th>
<th>Velocity y</th>
<th>Acceleration x</th>
<th>Acceleration y</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIV subset (30 neurons)</td>
<td>0.5339 ± 0.2047</td>
<td>0.8022 ± 0.2555</td>
<td>0.4985 ± 0.1440</td>
<td>0.4985 ± 0.0940</td>
<td>0.7536 ± 0.1697</td>
<td>0.9993 ± 0.3562</td>
</tr>
<tr>
<td>Random subsets (30 neurons)</td>
<td>0.7887 ± 0.2683</td>
<td>0.7706 ± 0.3872</td>
<td>0.7107 ± 0.2037</td>
<td>0.7470 ± 0.2358</td>
<td>0.8824 ± 0.1920</td>
<td>0.9766 ± 0.3898</td>
</tr>
</tbody>
</table>

Fig. 10. Statistical performance (aNMSE) by random subsets across 50 trials vs. MIV subset performance along different segment of data. The mean performance by random subsets is marked by blue cross with the error bar size indicating the standard derivation. The performance by MIV subset is marked by red stars.

4. Conclusion

Here we propose mutual information as a criterion to evaluate neuron importance based on its tuning properties. We compare the mutual information criterion with the traditional tuning depth analysis and show that it is more accurate in synthetic data because it takes into consideration the data distribution while the conventional algorithms do not. Although the estimation of mutual information is more involved, simple Parzen estimation seems to provide reasonable accuracy. For BMI data, mutual information is applied to the instantaneous tuning model on point processes to ascertain neuron importance associated with the kinematics. The information theoretic criterion statistically quantifies the amount of information between the kinematic vectors triggered off by the spike train, which may be considered all that matters for motor BMI signal translation. By setting a single threshold, it helps determine a subset of neurons most tuned to the task. The selected neurons in the subset exist mainly in M1. Selecting subsets decreases the computation complexity, and provide comparable performance on kinematic estimation compared to the full ensemble.

We further compare the decoding performance with the same number of neurons selected by our method with random chosen subsets. This is important because it is misleading just to compare decoding performance across several model orders due to the different generalization across model orders. Our selected neuron subsets reduce NMSE dramatically for most of the kinematic variables, which demonstrates the effectiveness of our subset-extraction approach. However, our assumption of including both
x and y directions to construct the instantaneous kinematic vector to estimate tuning may not be warranted. In fact, due to the task specifics, the movements in y direction have less amplitude than the movements along x direction, and this impacts the quality of the reconstruction, as well as our assessment of neural tuning. Future work should quantify neural tuning for each direction independently. Moreover, as we investigated the decoding performance along different segments of data, our results suggest that the importance of the neurons vary over time. Therefore, it is necessary to explore the time-variant neuronal importance for BMI applications in the future.

Acknowledgments

This work was supported by NSF grants ECS-0422718, CISE-0541241 and CNS-0540304. The authors would like to thank Dr. Miguel Nicolelis for the use of the monkey data collected in his laboratory at Duke University.

References