34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Transductive Neural Decoding for Unsorted Neuronal Spikes of Rat Hippocampus

Zhe Chen, Fabian Kloosterman, Stuart Layton, and Matthew A. Wilson

Abstract— **Neural decoding is an important approach for extracting information from population codes. We previously proposed a novel transductive neural decoding paradigm and applied it to reconstruct the rat's position during navigation based on unsorted rat hippocampal ensemble spiking activity. Here, we investigate several important technical issues of this new paradigm using one data set of one animal. Several extensions of our decoding method are discussed.**

I. Introduction

Neural decoding, as an inverse problem to neural encoding analysis, aims to infer sensory stimuli or motor kinematics based on recorded ensemble neuronal spiking activity. Neural decoding is important not only for understanding neural codes (i.e., neural response features capable of representing all information that neurons carry about the stimuli of interest), but also for extracting maximal information from population neurons in engineering applications, such as brain-machine interfaces [13]. Traditional neural decoding methods based on spiking activity [3], [21], [20], [12] rely on spike sorting, a process that is computationally expensive, time-consuming, and prone to errors [8], [18], [19]. To overcome this drawback, we have proposed a novel *transductive* neural decoding paradigm and applied it to unsorted rat hippocampal population codes [10].[1] Unlike traditional neural encoding/decoding methods, the proposed paradigm does not require estimating tuning curves for individual sorted single units. Our paradigm is also different from other spike-sorting-free decoding methods in the literature [6], [17] in that spike waveform features are used in decoding analysis.

In this paper, we first briefly review the transductive, spike sorting-free decoding method [10], before discussing in greater detail several technical issues related to application of the method to neural data. Next, we discuss extensions to the proposed method. From an information-theoretic perspective, we also propose a practical way to assess the mutual information between sensory stimuli and neural spiking responses, which are mathematically characterized by a spatio-temporal Poisson process (STPP).

II. An Overview of Transductive Neural Decoding

The basic idea of transductive neural decoding described in [10] is to model ensemble neuronal spiking activity as a spatio-temporal point process [15], in which the timing of spike events is defined with a random measure in time, and the "mark" associated with the spike events is defined with another random measure in real space.

A. Spatio-temporal Poisson Process (STPP)

Let us consider a STPP, which is the simplest spatio-temporal point process in which events are independent in time. Let $\lambda(t, \mathbf{a})$ denote the rate function, and $\mathbf{a} \in S$ (where S is a vector space). For any subset $S_j \in S$ in the space, the number of the events occurring inside the region is also a temporal Poisson process with associate rate function $\lambda_{S_j}(t)$:

$$\lambda_{S_j}(t) = \int_{S_j} \lambda(t, \mathbf{a}) \mathrm{d}\mathbf{a}, \quad \text{and} \quad \lambda_S(t) = \int_S \lambda(t, \mathbf{a}) \mathrm{d}\mathbf{a}. \tag{1}$$

The expected number of events in any spatio-temporal region is also Poisson distributed with the mean rate given by

$$\mu = \frac{1}{T} \int_0^T \int_S \lambda(t, \mathbf{a}) \mathrm{d}\mathbf{a} \mathrm{d}t = \int_0^T \lambda_S(t) \mathrm{d}t. \tag{2}$$

In the special case where the generalized rate function is a *separable* function of *space* and *time* such that

$$\lambda(t, \mathbf{a}) = \lambda_S(t) p(\mathbf{a}), \tag{3}$$

where $p(\mathbf{a})$ represents the spatial probability density function (pdf) of the random variable $\mathbf{a}$, and $\int_S p(\mathbf{a}) \mathrm{d}\mathbf{a} = 1$. The interpretation of the separable STPP is as follows: To generate random Poisson events in space-time, the first step is to sample a Poisson process with a rate function $\lambda_S(t)$, and the second step, is to draw a random vector $\mathbf{a}$ (associated with each event) from $p(\mathbf{a})$. Therefore, the spatio-temporal point process may be viewed as a purely *temporal marked point process*, with spatial marks at each time point of event occurrence from the ground process, and the marked space is defined by a random probability measure [15]. Detailed technical backgrounds are referred to [10].

B. Bayesian Decoding

In the context of neural decoding, let random variable $\mathbf{a} \equiv \{a_1, \dots a_d\} \in \mathbb{R}^d$ denote the measured d-dimensional feature extracted from the spike waveform (e.g., peak amplitude, waveform derivative, principal components, or any features that are used in spike sorting process), let $\mathbf{x} \in \mathbb{R}^q$ denote the sensory stimulus or motor covariate (such as the animal's position, head direction, velocity, etc.) that is being decoded. Furthermore, let n denote the number of spike events in the complete d-dimensional space within a time interval $[t, t + \Delta t)$, and let $\mathbf{a}_{1:n}$ denote the associated n d-dimensional spike waveform features. The d-dimensional feature space is divided evenly into J non-overlapping regions $S \equiv (S_1 \cup S_2 \cdots \cup S_J)$, and $\mathbf{a}_{1:n} \in S$.

To infer the probability of the unknown variable of interest $\mathbf{x}_t$ at time t, we resort to the Bayes rule

$$P(\mathbf{x}_t | \mathbf{a}_{1:n}) = \frac{P(\mathbf{a}_{1:n} | \mathbf{x}_t) P(\mathbf{x}_t)}{P(\mathbf{a}_{1:n})} \tag{4}$$

where $P(\mathbf{x}_t)$ denotes the prior probability, $P(\mathbf{a}_{1:n} | \mathbf{x}_t)$ denotes the likelihood, and the denominator denotes a normalizing constant. Provided that a non-informative *temporal prior* for $P(\mathbf{x}_t)$ is used (for this reason, from now on we will drop the subscript t on $\mathbf{x}_t$; the extension of using a temporal prior is discussed later), then Bayesian decoding is aimed to maximize the product of the likelihood and *spatial prior* $P(\mathbf{x})$:

$$P(\mathbf{x} | \mathbf{a}_{1:n}) \propto P(\mathbf{a}_{1:n} | \mathbf{x}) P(\mathbf{x}) \tag{5}$$

Supported by NIH Grants DP1-OD003646 and MH061976. The authors are with the Department of Brain and Cognitive Sciences, MIT, Cambridge, MA 02139, USA. F. Kloosterman is now with Neuro-Electronics Research Flanders, Leuven, Belgium. Z. Chen is also with the Neuroscience Statistics Research Lab, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114. (Email: zhechen@mit.edu)

[1] The term "transductive" is motivated by "transductive inference" initiated in the machine learning literature [16], which aims to avoid solving a more general problem. In the context of neural decoding, we aim to infer input stimuli from population codes without resorting to spike sorting.

To compute the likelihood, we assume that the spike events follow a *time-homogeneous* STPP with a generalized rate function $\lambda(\mathbf{a},\mathbf{x})$. It follows that the number of events occurring within a time window $[t, t+\Delta t)$ and subregion S_j in the d-dimensional spike feature space also follows a Poisson distribution with the rate function $\lambda_{\Delta t,S_j}(\mathbf{x}) = \Delta t \int_{S_j} \lambda(\mathbf{a},\mathbf{x})\mathrm{d}\mathbf{a}$, which can be viewed as a spatial tuning curve (TC) with respect to the covariate space $\mathbf{x}$. By dividing the spike feature space into J non-overlapping spatial subregions $S \equiv \cup_{j=1}^{J} S_j$, we can factorize the likelihood function into a product of Poisson likelihoods of all J subregions

$$\begin{aligned} P(\mathbf{a}_{1:n}|\mathbf{x}) &= \prod_{j=1}^{J} Poisson\Big(n(S_j); \lambda_{\Delta t,S_j}(\mathbf{x})\Big) \\ &= \frac{\Big[\prod_{j=1}^{J}\Big(\Delta t \int_{S_j}\lambda(\mathbf{a},\mathbf{x})\mathrm{d}\mathbf{a}\Big)^{n(S_j)}\Big]\Big[e^{-\Delta t\sum_{j=1}^{J}\int_{S_j}\lambda(\mathbf{a},\mathbf{x})\mathrm{d}\mathbf{a}}\Big]}{\prod_{j=1}^{J} n(S_j)!} \end{aligned} \quad (6)$$

where $n(S_j)$ denotes the number of spike events within the region S_j. In the limiting case when the subregion becomes sufficiently small such that $n(S_j)$ is equal to 0 or 1 within the time interval Δt, simplifying (6) and replacing it into (5) yields the posterior

$$P(\mathbf{x}|\mathbf{a}_{1:n}) \propto \prod_{i=1}^{n} \lambda(\mathbf{a}_i,\mathbf{x}) e^{-\Delta t\lambda(\mathbf{x})} P(\mathbf{x}) \quad (7)$$

where $\lambda(\mathbf{x})$ denotes the rate of spike events occurring in the covariate space $\mathbf{x}$.

To compute (7), we need to compute or establish a representation for the generalized rate function $\lambda(\mathbf{a},\mathbf{x})$ and its marginal rate function $\lambda(\mathbf{x})$. In practice, these rate functions are estimated *a priori* by recording spike events and their associated features while sampling over the covariate space. Note that the generalized rate function used in (6) can be written as

$$\lambda(\mathbf{a},\mathbf{x}) = \frac{\#spikes(\mathbf{a},\mathbf{x})}{occupancy(\mathbf{x})} = \frac{N}{T}\frac{p(\mathbf{a},\mathbf{x})}{\pi(\mathbf{x})} = \mu\frac{p(\mathbf{a},\mathbf{x})}{\pi(\mathbf{x})} \quad (8)$$

where N denotes the total number of spike events recorded within time interval $(0,T]$, μ is the mean spiking rate defined in (2), $\pi(\mathbf{x})$ denotes the occupancy probability of $\mathbf{x}$ during the complete time interval, and $p(\mathbf{a},\mathbf{x})$ denotes the joint pdf of $\mathbf{a}$ and $\mathbf{x}$. Furthermore, we have $\lambda(\mathbf{x}) = \mu\frac{p(\mathbf{x})}{\pi(\mathbf{x})}$, and $\lambda(\mathbf{a},\mathbf{x}) = \lambda(\mathbf{x})\frac{p(\mathbf{a},\mathbf{x})}{p(\mathbf{x})} = \lambda(\mathbf{x})p(\mathbf{a}|\mathbf{x})$, where $p(\mathbf{a}|\mathbf{x})$ denotes the conditional pdf.

In the decoding phase, in order to compute the likelihood (6), we would need to evaluate the target point in the functions $\lambda(\mathbf{a},\mathbf{x})$ and $\lambda(\mathbf{x})$, or equivalently in $p(\mathbf{a},\mathbf{x})$ and $p(\mathbf{x})$. Finally, to choose the *maximum a posteriori* (MAP) estimate of $\mathbf{x}$, denoted by $\mathbf{x}_{MAP}$, we simply evaluate the log-posterior (7) among all candidates in the $\mathbf{x}$ space, and choose the one that has the highest value.

C. Density Estimation: Parametric vs. Nonparametric Methods

In our decoding paradigm, the essential task is to estimate the joint pdf $p(\mathbf{a},\mathbf{x})$ and its marginal $p(\mathbf{x})$. Multivariate density estimation has been well studied in statistics [14]. Common methods include (i) parametric approaches, which model the data by a finite mixture model (a process similar to spike sorting at the first place); and (ii) nonparametric approaches, such as the histogram or kernel density estimation (KDE). Parametric representation is compact but less flexible; nonparametric approaches are model-free but more computationally expensive.

We investigate two methods here: one is based on an ℓ-mixtures of Gaussians (MoG) model, another based on Gaussian KDE, using a *non-isotropic* multivariate Gaussian kernel K

$$p(\mathbf{a},\mathbf{x}) = \sum_{r=1}^{\ell} \pi_r K\big(\mathbf{m}_r;\mathbf{H}_r\big) \quad (9)$$

$$p(\mathbf{a},\mathbf{x}) = \frac{1}{M\sigma_1\ldots\sigma_m h_1\ldots h_q}\sum_{m=1}^{M}\prod_{i=1}^{d} K\Big(\frac{a_i-\tilde{a}_{i,m}}{\sigma_i}\Big) \times \prod_{j=1}^{q} K\Big(\frac{x_j-\tilde{x}_{j,m}}{h_j}\Big) \quad (10)$$

where $\sum_{r=1}^{\ell}\pi_r = 1$, $(\mathbf{m}_r,\mathbf{H}_r)$ denote the r-th mean vector and diagonal (yet non-isotropic) covariance matrix, respectively, in the mixture model for the augmented vector $\mathbf{z} = (\mathbf{a},\mathbf{x})$; $\{\tilde{\mathbf{a}}_m,\tilde{\mathbf{x}}_m\}$ denotes the m-th source data point from the training set, and σ_i and h_j denote the kernel bandwidth (BW) parameters for the i-th and j-th element in $\mathbf{a}$ and $\mathbf{x}$, respectively.

For the MoG model, the unknown parameters $\{\pi_r,\mathbf{m}_r,\mathbf{H}_r\}_{r=1}^{\ell}$ are estimated from the expectation-maximization (EM) algorithm. For the KDE, the BW parameters are estimated from [2].

III. Mutual Information Between Stimulus and Neural Responses

If we denote X as the sensory stimuli, and R as the raw neuronal responses (spike waveform). Any feature extraction from raw data (such as spike count, PCA) can be modeled as a generic nonlinear function f. According to the *Data Processing Inequality*, post-processing of R never increases the mutual information between X and R [13]

$$I(X;f(R)) \le I(X;R) \quad (11)$$

This inequality is also applicable to spike sorting. In comparison to spike sorting-based decoding, spike sorting-free decoding sidesteps the sorting (clustering) process and reduces information loss, and prevents accumulating sorting error into decoding analysis.

The mutual information between the sensory input $\mathbf{x}$ and spike waveform features $\mathbf{a}$ is written as [4]

$$I(\mathbf{x};\mathbf{a}) = H(\mathbf{x}) + H(\mathbf{a}) - H(\mathbf{a},\mathbf{x}) \quad (12)$$

where the marginal and joint entropy functions can be estimated from KDE. For instance, given the multivariate kernel density estimator of an unknown pdf $p(\mathbf{x})$, a simple plug-in resubstitution estimator for differential entropy can be written as [1]: $\hat{H}(\mathbf{x}) = -\frac{1}{M}\sum_{m=1}^{M}\log\hat{p}_M(\mathbf{x}_m)$, where $\hat{p}_M(\mathbf{x})$ denotes a kernel density estimator based on M data samples.

In the presence of multi-electrode recording, given the spike waveform feature $(\mathbf{a}_1,\ldots,\mathbf{a}_\ell)$ from ℓ electrodes, the mutual information is given by [4]

$$\begin{aligned} I(\mathbf{x};\mathbf{a}_1,\ldots,\mathbf{a}_\ell) &= H(\mathbf{a}_1,\ldots,\mathbf{a}_\ell) - H(\mathbf{a}_1,\ldots,\mathbf{a}_\ell|\mathbf{x}) \\ &= H(\mathbf{a}_1,\ldots,\mathbf{a}_\ell) - \sum_{r=1}^{\ell} H(\mathbf{a}_r|\mathbf{x}) \quad (13) \\ &= H(\mathbf{a}_1,\ldots,\mathbf{a}_\ell) + \sum_{r=1}^{\ell}\Big[H(\mathbf{x}) - H(\mathbf{a}_r,\mathbf{x})\Big] \end{aligned}$$

where the second step follows from the conditional independence assumption between the covariate $\mathbf{x}$ and the neural response $\mathbf{a}_r$ from each electrode. The conditional entropy $H(\mathbf{a}_r|\mathbf{x}) = H(\mathbf{x}) - H(\mathbf{a}_r,\mathbf{x})$ can be estimated directly from KDE, and the joint entropy $H(\mathbf{a}_1,\ldots,\mathbf{a}_\ell)$ may be estimated with a resampling method.

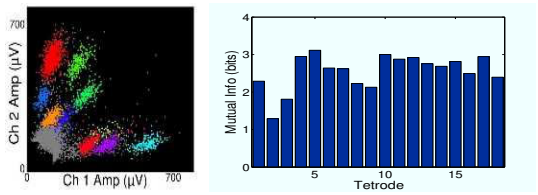

Fig. 1. *Left:* Raw spike amplitudes from one tetrode (shown in 2 channels). *Right:* Estimated mutual information (bits) between the position ($\mathbf{x}$) and spike amplitude ($\mathbf{a}$) in each tetrode (computed from Eq. 12); note that they are all bounded by the entropy of stimulus (3.36 bits).

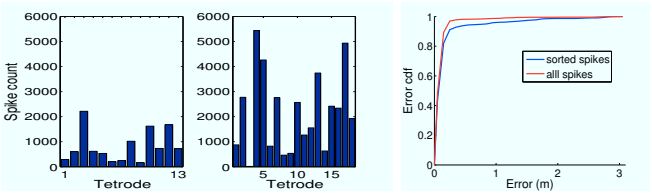

Fig. 2. *Left:* Numbers of spike counts per tetrode for sorted vs. all spikes. *Right:* Decoding error cdf curves from using sorted and all spikes.

IV. Application: Position Reconstruction with Unsorted Rat Hippocampal Ensemble Spiking Activity

A. Data

For experimental protocol and details, the reader is referred to [10]. Animals were traveling in a 3.1-m linear track environment, which was binned with 0.1-m bin size resulting in 31 position bins. Simultaneous tetrode recordings were collected from the CA1 area of rat hippocampus. In each recording recording session, the waveforms of all unsorted spikes were re-thresholded at 75 μV. Next, for unsorted spike events, the spikes with a peak to trough width of greater than 150 μs are considered as originating from pyramidal cells and are included in the decoding analysis. For each tetrode, the peak amplitudes from 4 channels are used to construct $\mathbf{a} \in \mathbb{R}^4$ (see the left panel of Fig. 1 for illustration). In one selected data set studied here, we collect 48 putative pyramidal cells from 18 tetrodes within about 30-min recordings. The first half of the data is used as the training set. The temporal bin size is chosen as $\Delta t = 250$ ms, and only run periods (velocity filter 0.15 m/s) are chosen in encoding and decoding analyses. The decoding error is defined as $|\mathbf{x}_{true} - \mathbf{x}_{MAP}|$ ($\mathbf{x} \in \mathbb{R}$) for each temporal bin.

B. Results

1) Transductive Decoding with Sorted vs. All Spikes: For the selected data set, we show the number of spikes per tetrode based on sorted (10664) spikes or all recorded (39383) spikes (Fig. 2, left panel). As seen, nearly 73% recorded spikes are discarded in spike sorting. Potentially, many non-clusterable spikes contain tuning information; and traditional spike sorting-based decoding methods may suffer an information loss by discarding those spikes. The decoding error cumulative distribution function (cdf) curve (Fig. 2, right panel) indicates a statistically significant improvement in decoding accuracy (two-sample KS test: $P < 0.001$). The median (mean) statistics of decoding error are 0.1111 (0.1920) m for using all spikes, and 0.1172 (0.2051) m for using only sorted spikes. This result also confirms our previous finding [10].

2) Parametric vs. Nonparametric Density Estimation: Next, we compare two density estimators (Eqs. 9 and 10) in our proposed transductive neural decoding paradigm. For the nonparmetric method, we use Gaussian KDE with non-isotropic BW parameters.

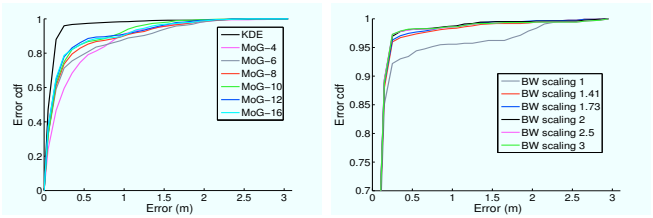

Fig. 3. Decoding error cdf curves from both mixtures of Gaussians (MoG) and KDE methods (*left*), and from using various BW scalings (*right*).

TABLE I

Statistics of the mean/median decoding error (unit: meter) using various thresholds or various compressed sample size M for our transductive amplitude-based decoding.

threshold (μV)	sorted spikes	all spikes
100	0.2056/0.1159	0.1928/0.1009
125	0.2174/0.1176	0.2057/0.1116
150	0.2347/0.1197	0.2132/0.1154
M (per tetrode)	sorted spikes	all spikes
500	0.4536/0.1876	N/A
1000	0.3283/0.1307	N/A
2000	0.2444/0.1194	0.5642/0.1598
3000	0.2051/0.1172	0.3960/0.1279
ALL data	0.2051/0.1172	0.1920/0.1111

For the parametric method, we use various numbers (4, 6, 8, 10, 12) of MoG for each tetrode, resulting in a maximum of 216 multivariate Gaussians for 18 tetrodes. Note that using a Gaussian mixture for density estimation is in spirit similar to the clustering process during spike sorting, except that we estimate $(\mathbf{a}, \mathbf{x})$ jointly (instead of $\mathbf{a}$ alone in spike sorting), and that the spread of the kernels is allowed to be overlapping (without making hard decisions). The decoding results are shown in Fig. 3 (left panel). As seen, the decoding accuracy also improves as the number of the Gaussian mixtures increases. The nonparametric method has a better decoding performance due to its more accurate representation of the density. Besides, for the decoding purpose, a *local* density representation is more preferable to a *global* characterization.

3) Reduction of Source Samples: In KDE representation, the density is represented by M source data points (at one electrode). Obviously, the storage requirement and computational complexity of decoding is linearly proportional to M. To reduce the computational burden, we attempt to reduce the source samples by two methods. The first method uses a higher threshold (in our case, greater than 75 μV) to exclude low-amplitude spike events. Generally, the low-amplitude spikes have less recoverable information of the stimulus. The second method aims to compress source samples using some computational methods [11], [7], [9]. Here we use a computationally efficient KD-tree method (http://www.ics.uci.edu/∼ihler/code/kde.html).

The results of the decoding error statistics are summarized in Table I. As seen from the mean/median error statistics, the decoding accuracy degrades while using a very high threshold; however, better performance can also be expected using a slightly higher threshold (e.g., 100 μV). On the other hand, reducing source sample size using a computational method always degrades decoding accuracy, regardless of the data source (sorted spikes or all spikes).

4) Scaling the BW Parameters: In the presence of noisy spikes (in the low-amplitude space), it is common to use a larger kernel BW to smooth the noise-contaminated samples. To test this idea, we fix the position BW (0.05 m) and scale the initial amplitude BW

(estimated from [2]) by different scalars ($\sqrt{2}$, $\sqrt{3}$, 2, 2.5, 3) in all four dimensions. The decoding error cdf curves are shown in Fig. 3 (right panel). For this data set, the optimal scaling parameter is 2, achieving the median (mean) decoding error of 0.1043 (0.1346) m. Note that the mean decoding error is greatly reduced.

V. Extensions and Discussion

A. Curse of Dimensionality

In a general setting, the dimensionality of the covariate space can be very large: either q is large, or the range for individual univariate dimension is large (with a relatively small bin size). For MAP estimation, a naive even binning of the covariate space can be extremely inefficient, since the occupancy density $\pi(\mathbf{x})$ may be very sparse. To tackle the "curse of dimensionality" problem, we may use a *divide-and-conquer* approach by constructing q independent decoders, each one equipped with its own density estimator. Another way is to use informative cues to draw candidate samples from an informative covariate space (an idea similar to importance sampling) [5]. Here we discuss two sampling approaches.

1) Sampling from a Temporal Prior: Within the state-space framework, we can sample the current covariate $\mathbf{x}$ from a transition prior of the covariate at the previous discrete time step [3], [21]:

$$P(\mathbf{x}_t) = \int p(\mathbf{x}_t, \mathbf{x}_{t-1}) \mathrm{d}\mathbf{x}_{t-1} = \int p(\mathbf{x}_t|\mathbf{x}_{t-1}) p(\mathbf{x}_{t-1}) \mathrm{d}\mathbf{x}_{t-1} \quad (14)$$

where $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ denotes a transition probability density. The posterior can be estimated used a recursive Bayesian filtering rule (for notation simplicity, we have ignored the subscript $_{1:n}$ for $\mathbf{a}$)

$$\begin{aligned} P(\mathbf{x}_t|\mathbf{a}_{1:t}) &= \frac{P(\mathbf{x}_t, \mathbf{a}_{1:t})}{P(\mathbf{a}_{1:t})} = \frac{P(\mathbf{x}_t, \mathbf{a}_{1:t}|\mathbf{a}_{1:t-1}) P(\mathbf{a}_{1:t-1})}{P(\mathbf{a}_t|\mathbf{a}_{1:t-1}) P(\mathbf{a}_{1:t-1})} \\ &= \frac{P(\mathbf{x}_t|\mathbf{a}_{1:t-1}) P(\mathbf{a}_t|\mathbf{x}_t, \mathbf{a}_{1:t-1})}{P(\mathbf{a}_t|\mathbf{a}_{1:t-1})} \end{aligned} \quad (15)$$

where $P(\mathbf{a}_t|\mathbf{x}_t, \mathbf{a}_{1:t-1}) = P(\mathbf{a}_t|\mathbf{x}_t)$ (because of the statistical independence between $\mathbf{a}_t$ and $\mathbf{a}_{1:t-1}$ in the spike waveform feature space) denotes the data likelihood at the t-th time step.

2) Kernel Regression: Since at every time step t, we observe the current spike waveform feature $\mathbf{a}_t$; ideally, the candidate sample is drawn from the *mode* of the posterior $P(\mathbf{x}|\mathbf{a}_t)$. However, searching for the mode in a high-dimensional covariate space is a very challenging problem. Instead, we can search for the *mean* in the sample space, which may be computed by a continuous multi-input multi-ouput mapping through nonparametric regression $\mathbf{x} = \mathbf{g}(\mathbf{a})$, where $\mathbf{g}(\cdot)$ is a locally smooth multivariate function

$$\mathbf{g}(\mathbf{a}) = \mathbb{E}_{\mathbf{x}|\mathbf{a}}[\mathbf{x}] = \int \mathbf{x} P(\mathbf{x}|\mathbf{a}) \mathrm{d}\mathbf{x} = \frac{\sum_{m=1}^{M} \tilde{\mathbf{x}}_m K\left(\frac{\mathbf{a}-\tilde{\mathbf{a}}_m}{\sigma}\right)}{\sum_{m=1}^{M} K\left(\frac{\mathbf{a}-\tilde{\mathbf{a}}_m}{\sigma}\right)} \quad (16)$$

Eq. (16) is known as *Nadaraya-Watson kernel regression*. However, in the presence of noisy spikes and multi-modes in $P(\mathbf{x}|\mathbf{a})$, this scheme might not be effective. Alternatively, we may draw candidate samples from $P(\mathbf{a}|\mathbf{x})$ using an auxiliary variable [5].

B. Other Issues

Several remaining issues are worth mentioning. First, region-dependent kernel BW parameters can be considered in KDE. For instance, in the low spike-amplitude space, we may use a small BW for the dense noisy spikes, while a large BW is preferred in the median-to-high amplitude space. In addition, finding a meaningful representation of the feature (e.g., by nonlinear transformation) and selecting an appropriate kernel function (in either parametric or nonparametric density estimation) would help separate different feature clusters and improve the decoding accuracy. Second, we have assumed that each sample point contributes equally in KDE. Alternatively, samples can be merged (according to certain similarity measure) and assigned with unequal weights [22], which also helps reduce the sample size.

All of above-mentioned topics will be the subject of our future decoding analysis investigation, using recordings not only from the rat hippocampus, but also from other brain regions (e.g., primate primary motor cortex). In addition, real-time implementations of our transductive neural decoding paradigm using online (parametric or nonparametric) density estimation is currently under investigation.

References

[1] J. Beirlant, E. Dudewicz, L. Gyor, and E. Van Der Meulen, "Nonparametric entropy estimation: An overview," *Int. J. Math. Stat. Sci.*, vol. 6, pp. 17–39, 1997.

[2] Z. I. Botev, J. F. Grotowski and D. P. Kroese, "Kernel density estimation via diffusion," *Ann. Stat.*, vol. 38, pp. 2916–2957, 2010.

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

[4] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley, 1991.

[5] A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods in Practice*, Springer, 2001.

[6] G. Fraser, S. M. Chase, A. Whitford, and A. B. Schwartz, "Control of a brain-computer interface without spike sorting," *J. Neural Eng.*, vol. 6, pp. 055004, 2009.

[7] M. Girolami and C. He, "Probability density estimation from optimally condensed data samples," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 25, pp. 1253–1264, 2003.

[8] K. D. Harris, D. A. Henze, J. Csicsvari, H. Hirase, and G. Buzsaki, "Accuracy of tetrode spike separation as determined by simultaneous intracellular and extracellular measurements," *J. Neurophysiol.*, vol. 84, pp. 401–414, 2000.

[9] D. Huang and T. W. S. Chow, "Enhancing density-based data reduction using entropy," *Neural Computat.*, vol. 18, pp. 470–495, 2006.

[10] F. Kloosterman, S. Layton, Z. Chen, and M. A. Wilson, "Bayesian decoding of unsorted spikes in the rat hippocampus", *J. Neurophysiol.*, under review.

[11] P. Mitra, C. A. Murthy and S. K. Pal, "Density-based multiscale data condensation," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 24, pp. 1–14, 2002.

[12] J. W. Pillow, Y. Ahmadian, and L. Paninski, "Model-based decoding, information estimation, and change-point detection techniques for multineuron spike trains," *Neural Computat.*, vol. 23, pp. 1–45, 2010.

[13] R. Q. Quiroga and S. Panzeri, "Extracting information from neuronal populations: information theory and decoding approaches," *Nat. Rev. Neurosci.*, vol. 10, pp. 173–185, 2009.

[14] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, New York, 1992.

[15] D. L. Snyder and M. I. Miller, *Random Point Processes in Time and Space*, Springer, New York, 1991.

[16] V. N. Vapnik, *Statistical Learning Theory*. Wiley, New York, 1998.

[17] V. Ventura, "Spike train decoding without spike sorting," *Neural Computat.*, vol. 20, pp. 923–963, 2008.

[18] D. S. Won, P. H. E. Tiesinga, C. S. Henriquez, and P. D. Wolf, "Analytical comparison of the information in sorted and non-sorted cosine-tuned spike activity", *J. Neural Eng.*, vol. 4, pp. 322–335, 2007.

[19] F. Wood, M. Fellows, C. Vargas-Irwin, M. J. Black, and J. P. Donoghue, "On the variability of manual spike sorting," *IEEE Trans. Biomed. Engr.*, vol. 51, pp. 912–918, 2004.

[20] R. S. Zemel, P. Dayan and A. Pouget, "Probabilistic interpretation of population codes," *Neural Computat.*, vol. 10, pp. 403–430, 1998.

[21] K. Zhang, I. Ginzburg, B. L. McNaughton, and T. J. Sejnowski, "Interpreting neuronal population activity by reconstruction: unified framework with application to hippocampal place cells," *J. Neurophysiol.*, vol. 79, pp. 1017–1044, 1998.

[22] A. Zhou, Z. Cai, L. Wei and W. Qian, "M-kernel merging: Towards density estimation over data streams," *Proc. 8th Int. Conf. Database Systems for Advanced Applications*, pp. 285–292, 2003.