Bayesian Decoder Models with a Discriminative Observation Process

Mohammad R. Rezaei
Department of Electrical Engineering
Isfahan University of Technology
Isfahan 84156-83111, Iran.

Alex E. Hadjinicolaou
Department of Neurology
Massachusetts General Hospital
Boston, MA 02114, U.S.A.

Sydney S. Cash
Department of Neurology
Massachusetts General Hospital
Boston, MA 02114, U.S.A.

Uri T. Eden
Department of Mathematics and Statistics
Boston University
AddressBoston, MA 02215, U.S.A.

Ali Yousefi *
Department of Computer Science
Worcester Polytechnic Institute
Worcester, MA 01609, U.S.A.
ayousefi@wpi.ed

Abstract

The Bayesian state-space neural encoder-decoder modeling framework is an established solution to reveal how changes in brain dynamics encode physiological covariates like movement or cognition. Although the framework is increasingly being applied to progress the field of neuroscience, its application to modeling high-dimensional neural data continues to be a challenge. Here, we propose a novel solution that avoids the complexity of encoder models that characterize high-dimensional data as a function of the underlying state processes. We build a discriminative model to estimate state processes as a function of current and previous observations of neural activity. We then develop the filter and parameter estimation solutions for this new class of state-space modeling framework called the "direct decoder" model. We apply the model to decode movement trajectories of a rat in a W-shaped maze from the ensemble spiking activity of place cells and achieve comparable performance to modern decoding solutions, without needing an encoding step in the model development. We further demonstrate how a dynamical auto-encoder can be built using the direct decoder model; here, the underlying state process links the high-dimensional neural activity to the behavioral readout. The dynamical auto-encoder can optimally estimate the low-dimensional dynamical manifold which represents the relationship between brain and behavior.

1 Introduction

The rapid development of neural recording technologies over the last few decades has enabled the simultaneous recording of neural activity from an ever-increasing number of brain regions. For research groups interested in relating brain activity to higher-level processes, these data are often recorded during some sort of experimental task, together with behavioral or cognitive observations

*

Preprint. Under review.

that are influenced by the task (**?**). The higher dimension and multi-modality of these data necessitate the development of analytical solutions capable of making statistically robust inferences about the underlying brain dynamics and their relationship to the observed correlates (1; 2). A wide variety of statistical and machine learning techniques, broadly known as neural encoder-decoder models, have been developed to address this particular type of problem. Such models are built in two stages: first, a neural encoder model builds the conditional distribution of observed neural data given the underlying neural correlates (such as movement or cognitive state), and then, newly observed neural data are decoded to estimate those correlates by applying Bayes' theorem to the encoder model (3; 4; 5; 6; 7; 8). Although traditional neural encoder-decoder models have been successfully applied to gain insights from low-dimensional data (3; 9; 10), they face multiple modeling challenges when applied to high-dimensional data. One of such problems appears in the encoding step, in which the conditional joint distribution of neural data is built. Due to the large dimension of data, it is hard to properly characterize this distribution. Proposed solutions for this distribution are mainly built upon naive assumptions such as the conditional independence of the individual neural data given the correlates. Even with this assumption, it is not always possible to characterize the distributions of neural data and neural noise (e.g. they may not be stationary), which introduces further complications and makes it difficult to build the encoder model. The fact that the neural correlates generally have a lower dimension compared to the neural data might help to address some of the challenges linked to the classical encoder-decoder modeling methodologies (11; 12; 13; 14). In this work, we propose a Bayesian filter solution for the decoder model in which we build it directly from the neural data ensemble, as opposed to first formulating the encoder model. A specific variation of this modeling approach has been developed by Harrison et al. (15) in which the decoder model is defined as a function of the current-time neural observation and the filter solution is derived for the steady-state condition. Here, we introduce a more general interpretation of this modeling approach, in which the decoder model is defined as a function of current and previous neural data, and the filter solution accounts for time variability present in the observed data. In addition to the filter solution, we also derive the maximum likelihood (ML) estimation for the model parameters using a revised expectation-maximization (EM) technique (16). We then apply our modeling framework to decode the 2-D trajectories of a rat moving through a maze from the ensemble spiking activity of place cells, demonstrating decoding results that are comparable to those of a point-process encoder-decoder model (17), but without the need for an encoder model. Our proposed Bayesian filter solution can be applied to a broader class of neural encoding-decoding problems in which the connection between brain dynamics and neural correlates are defined through a low-dimensional dynamical manifold. For instance, when behavioral readouts are used as the correlates, the manifold will represent the cognitive states that underlie these behaviors. To address these sorts of problems, we propose a modeling solution in which a behavioral encoder model is used together with the direct neural decoder model to find the dynamical manifold linking behavior and the underlying neural activity. The proposed encoder-decoder model can be viewed as a dynamical auto-encoder model with the cognitive states as the latent manifold, and the behavior and neural data as different measures of the same dynamical latent structure. We conclude our findings with an application of this solution to a novel decoding problem in which we seek to decode the communicative intentions of an epileptic study participant (with electrodes implanted for clinical purposes) from their neural activity, by modeling communicative intent as a cognitive state.

2 Method

Here we begin by formulating the direct-decoder (D-D) model using a discriminative observation process. We derive the Bayesian filter and parameter estimation solution for this model before generalizing it to form a dynamical auto-encoder. We then propose a revised EM algorithm that helps us to find maximum likelihood estimations of the dynamical auto-encoder model parameters.
In the state-space modeling, care must be taken to build an accurate model of the observation process. We focus on the class of problems for which the dimensionality of neural observations is much larger than the number of state processes, an appropriate constraint for the wide range of problems in neuroscience that deal with multi-electrode neural recordings.

2.1 Direct Decoder Model

Let us assume we have K observations from $k = 1\ to\ K$. Let $\boldsymbol{x_k}$ represent the cognitive state at time index k, and $\boldsymbol{s_k}$ the neural observation process at the same time index k.We define the history term $\boldsymbol{h_k}$ as the subset of previous observations, $\boldsymbol{h_k} \subset \{\ \boldsymbol{s_1}, \cdots, \boldsymbol{s_{k-1}}\}$. As with a Bayes filter solution, our objective is to estimate

$$p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k}}) \tag{1}$$

Using a recursive filter solution (18), the filter update rule at time index k is defined by

$$p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k}}) \propto p(\boldsymbol{s_k}|\boldsymbol{s_{1,\ldots,k-1}}, \boldsymbol{x_k}) \times p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k-1}}) = p(\boldsymbol{s_k}|\boldsymbol{h_k}, \boldsymbol{x_k}) \times p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k-1}}) \tag{2}$$

Given the definition of the history term, we can rewrite the filter update rule as

$$p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k}}) \propto \frac{p(\boldsymbol{x_k}|\boldsymbol{h_k}, \boldsymbol{s_k})}{p(\boldsymbol{x_k}|\boldsymbol{h_k})} \times p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k-1}}) = \frac{p(\boldsymbol{x_k}|\boldsymbol{h_k}, \boldsymbol{s_k})}{p(\boldsymbol{x_k}|\boldsymbol{h_{k-1}}, \boldsymbol{s_{k-1}})} \times p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k-1}}) \tag{3}$$

Now, we can build a recursive solution for the update rule using the Chapman-Kolmogorov equation (19):

$$p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k}}) \propto \frac{p(\boldsymbol{x_k}|\boldsymbol{h_k}, \boldsymbol{s_k})}{\int p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}}) p(\boldsymbol{x_{k-1}}|\boldsymbol{h_{k-1}}, \boldsymbol{s_{k-1}}) d\boldsymbol{x_{k-1}}} \times$$

$$\int p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}}) p(\boldsymbol{x_{k-1}}|\boldsymbol{s_{1,\ldots,k-1}}) d\boldsymbol{x_{k-1}} \tag{4}$$

The fraction term on the right-hand-side of equation (4) represents the likelihood function in the standard state-space model. It is the ratio of two likelihood functions for each value of $\boldsymbol{x_k}$. The denominator defines the likelihood of $\boldsymbol{x_k}$ given the history of observation until time k and the numerator is the likelihood of $\boldsymbol{x_k}$ when considering the current observation together with the history term. This likelihood can be large or small depending on the information being carried by $\boldsymbol{s_k}$ about $\boldsymbol{x_k}$, which changes the posterior distribution of $\boldsymbol{x_k}$ given the observation until time k. We define the state transition process at time index k by

$$\boldsymbol{x_k}|\boldsymbol{x_{k-1}} \sim f(\boldsymbol{x_{k-1}}; \boldsymbol{\theta}) \tag{5}$$

where $\boldsymbol{x_k}$ is the cognitive state variable at time index k and $\boldsymbol{\theta}$ is the set of free parameters of the state equation. We can use the D-D filter solution derived in equation (4) to build the conditional distribution of state for the discriminative model, given the current observation and observation history. We call this the prediction process, which is described as

$$\boldsymbol{x_k}|\boldsymbol{s_k}, \boldsymbol{h_k} \sim f(\boldsymbol{s_k}, \boldsymbol{h_k}; \boldsymbol{\Omega}) \tag{6}$$

where $\boldsymbol{s_k}$ and $\boldsymbol{h_k}$ are the neural activity and history term at time index k, and $\boldsymbol{\Omega}$ is the set of free parameters for the discriminative model.

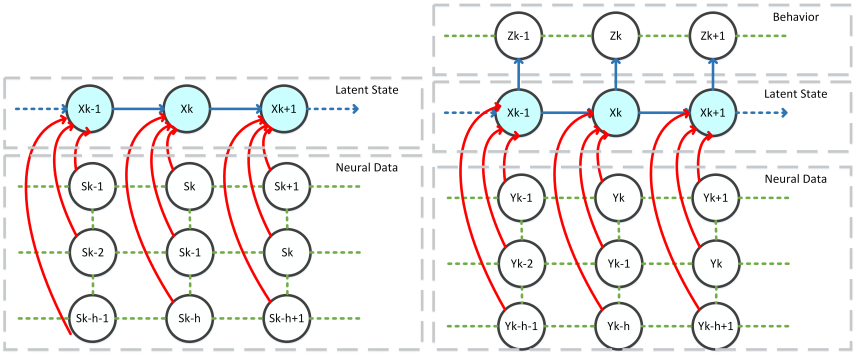
The D-D model (represented by the schematic in Figure 1) is comprised of the state and prediction processes defined by equations (5) and (6). With the prediction process, we no longer require an explicit description of the observation process, or the conditional distribution of the observation. The noise process in the prediction process is well-defined – the noise process is already defined in the state process – and thus it can be easily constructed. The prediction process itself can be modeled using a variety of techniques (e.g., with a generalized linear model (GLM) (11), a neural network (7), or a regularized prediction) and can also incorporate non-linear terms like interaction terms defined by $\boldsymbol{s_k}$ and $\boldsymbol{h_k}$ and their higher-order combinations.

For the model prediction step (defined in the next section) we require a smoother solution for the state, which is defined by

$$p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,K}}) = p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k}}) \int \frac{p(\boldsymbol{x_{k+1}}|\boldsymbol{x_k}) p(\boldsymbol{x_{k+1}}|\boldsymbol{s_{1,\ldots,K}})}{p(\boldsymbol{x_{k+1}}|\boldsymbol{s_{1,\ldots,k}})} d\boldsymbol{x_{k+1}} \tag{7}$$

2.2 Dynamical auto-encoder model

Here we discuss how the D-D model can be expanded to a dynamical auto-encoder model (14). Let $\boldsymbol{z_k}$ represent the behavioral observation at time index k, given both $\boldsymbol{s_k}$ and $\boldsymbol{z_k}$. We define the



(a) (b)

Figure 1: (a. Schematic representation of the direct-decoder model. $\boldsymbol{s_k}$ represents neural activity at time index k, h represents the number of previous time points in the history term (determined by model selection techniques), and $\boldsymbol{x_k}$ is the state variable. b. Schematic representation of the dynamical auto-decoder model. $\boldsymbol{z_k}$ represents the behavioral readout at time index k, which is defined as a function of the state processes. The other parts of the model are the same as the D-D model.

history term $\boldsymbol{h_k}$ as the subset of previous observations, $\boldsymbol{h_k} \subset \{\boldsymbol{s_1}, \cdots, \boldsymbol{s_{k-1}}\}$, our objective is to estimate

$$p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k}}, \boldsymbol{z_{1,\ldots,k}}) \tag{8}$$

As with the D-D methodology, we can express (8) as

$$p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k}}, \boldsymbol{z_{1,\ldots,k}}) \propto p(\boldsymbol{s_k}, \boldsymbol{z_k}|\boldsymbol{h_k}, \boldsymbol{x_k}) \times p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k-1}}, \boldsymbol{z_{1,\ldots,k-1}}) \tag{9}$$

We assume that there exists an underlying latent state conditioned on that the $\boldsymbol{s_k}$ and $\boldsymbol{z_k}$ are independent[16]. As a result, we can rewrite equation (9) as

$$p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k}}, \boldsymbol{z_{1,\ldots,k}}) = p(\boldsymbol{s_k}|\boldsymbol{h_k}, \boldsymbol{x_k}) \times p(\boldsymbol{z_k}|\boldsymbol{h_k}, \boldsymbol{x_k}) \times p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k-1}}, \boldsymbol{z_{1,\ldots,k-1}}) \tag{10}$$

Given the definition of the history term, we can rewrite the filter update rule as

$$p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k}}, \boldsymbol{z_{1,\ldots,k}}) \propto \frac{p(\boldsymbol{x_k}|\boldsymbol{h_k}, \boldsymbol{s_k})}{p(\boldsymbol{x_k}|\boldsymbol{h_k})} \times p(\boldsymbol{z_k}|\boldsymbol{x_k}) \times p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k-1}}, \boldsymbol{z_{1,\ldots,k-1}}) \tag{11}$$

where $\boldsymbol{z_{1,\ldots,k-1}}$ is assumed to be indepdent of $\boldsymbol{h_{1,\ldots,k-1}}$ given the state process, $\boldsymbol{x_{1,\ldots,k-1}}$. Now, we can build a recursive solution for the update rule (13):

$$p(\boldsymbol{x_k}|\boldsymbol{s_{1,\ldots,k}}, \boldsymbol{z_{1,\ldots,k}}) \propto \frac{p(\boldsymbol{x_k}|\boldsymbol{h_k}, \boldsymbol{s_k})}{\int p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}}) p(\boldsymbol{x_{k-1}}|\boldsymbol{h_{k-1}}, \boldsymbol{s_{k-1}}) d\boldsymbol{x_{k-1}}} \times p(\boldsymbol{z_k}|\boldsymbol{x_k}) \times$$

$$\int p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}}) p(\boldsymbol{x_{k-1}}|\boldsymbol{s_{1,\ldots,k-1}}, \boldsymbol{z_{1,\ldots,k-1}}) d\boldsymbol{x_{k-1}} \tag{12}$$

As with the D-D model, we describe the state transition process at time index k as a function of the previous state value and $\boldsymbol{\theta}$, the set of free parameters of the state equation.

$$\boldsymbol{x_k}|\boldsymbol{x_{k-1}} \sim f(\boldsymbol{x_{k-1}}; \boldsymbol{\theta}) \tag{13}$$

For the auto-encoder model, we have two processes: (1) a prediction process similar to what has been derived for the D-D model, and (2) an observation process. These are described by

$$\boldsymbol{x_k}|\boldsymbol{s_k}, \boldsymbol{h_k} \sim f(\boldsymbol{s_k}, \boldsymbol{h_k}; \boldsymbol{\Omega}) \tag{14}$$

$$\boldsymbol{z_k}|\boldsymbol{x_k} \sim f(\boldsymbol{x_k}; \boldsymbol{\omega}) \tag{15}$$

where equation (14) is analogous to equation (6) and equation (15) is the $\boldsymbol{z_k}$ observation process. In (15), $\boldsymbol{\omega}$ is the set of free parameters describing the behavioral encoder model.
In the D-D and dynamical auto-encoder models described so far, we assume the model parameters and the state dimension are known. In the next section, we describe how the model parameters can be estimated given either $\boldsymbol{s_k}$ or both $\boldsymbol{s_k}$ and $\boldsymbol{z_k}$.

2.3 Model Parameter Estimation

We use the expectation-maximization algorithm (20) to find maximum likelihood estimates of the model parameters, which can be any subset of $\{\boldsymbol{\theta}, \boldsymbol{\Omega}, \boldsymbol{\omega}\}$. The EM algorithm is an established solution to perform maximum likelihood estimation of model parameters when there is an unobservable process or missing observations (14). In the D-D and auto-encoder models, the state variable $\boldsymbol{x_k}$ is the unobserved process, the latent dynamical variable. Here we discuss the EM solution for the auto-encoder model, given that the D-D model is a specific form of the auto-encoder model. The solution recursively estimates the model parameters $\{\boldsymbol{\theta}^{(r)}, \boldsymbol{\Omega}^{(r)}, \boldsymbol{\omega}^{(r)}\}$, based on an updated posterior distribution of $\boldsymbol{x_{0,\ldots,K}}$ and the observation from the previous EM iteration, $\{\boldsymbol{\theta}^{(r-1)}, \boldsymbol{\Omega}^{(r-1)}, \boldsymbol{\omega}^{(r-1)}\}$. The EM includes two steps: expectation and maximization (21). The expectation step (or Q function) is defined by

$$Q = \boldsymbol{E}_{\boldsymbol{x_{0,\ldots,K}}|\boldsymbol{s_{1,\ldots,K}},\boldsymbol{z_{1,\ldots,K}}}[p(\boldsymbol{x_0}) \prod_{k=1}^{K} p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}}) \times p(\boldsymbol{s_k}|\boldsymbol{x_k}, \boldsymbol{h_k}) \times p(\boldsymbol{z_k}|\boldsymbol{x_k})] \tag{16}$$

The Q function can be rewritten as

$$Q = \boldsymbol{E}_{\boldsymbol{x_{0,\ldots,K}}|\boldsymbol{s_{1,\ldots,K}},\boldsymbol{z_{1,\ldots,K}}}[p(\boldsymbol{x_0}) \prod_{k=1}^{K} p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}}) \times p(\boldsymbol{z_k}|\boldsymbol{x_k}) \times \frac{p(\boldsymbol{x_k}|\boldsymbol{s_k}, \boldsymbol{h_k})}{\int p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}}) p(\boldsymbol{x_{k-1}}|\boldsymbol{s_{k-1}}, \boldsymbol{h_{k-1}})} d\boldsymbol{x_{k-1}}] \tag{17}$$

Expanding the Q function yields

$$Q = \boldsymbol{E}_{\boldsymbol{x_{0,\ldots,K}}|\boldsymbol{s_{1,\ldots,K}},\boldsymbol{z_{1,\ldots,K}}}[log\ p(\boldsymbol{x_0}) + \sum_{k=1}^{K} log\ p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}}) + \sum_{k=1}^{K} log\ p(\boldsymbol{z_k}|\boldsymbol{x_k}) + \sum_{k=1}^{K} log\ p(\boldsymbol{x_k}|\boldsymbol{s_k}, \boldsymbol{h_k}) - \sum_{k=1}^{K} log \int p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}}) p(\boldsymbol{x_{k-1}}|\boldsymbol{s_{k-1}}, \boldsymbol{h_{k-1}}) d\boldsymbol{x_{k-1}}] \tag{18}$$

The Chapman-Kolmogorov equation in (18) can be expressed as

$$\int p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}}) p(\boldsymbol{x_{k-1}}|\boldsymbol{s_{k-1}}, \boldsymbol{h_{k-1}}) d\boldsymbol{x_{k-1}} = \boldsymbol{E}_{\boldsymbol{x_{k-1}}|\boldsymbol{s_{k-1}},\boldsymbol{h_{k-1}}}[p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}})] \tag{19}$$

When the state process is linear with an additive Gaussian noise and the prediction process is a multi-variate normal, there is a closed form solution for this expectation. As a result, the Q function can be maximized to find maximum likelihood estimate of the model parameters. To derive a more general solution, we can rewrite Q function as

$$Q = \boldsymbol{E}_{\boldsymbol{x_{0,\ldots,K}}|\boldsymbol{s_{1,\ldots,K}},\boldsymbol{z_{1,\ldots,K}}}[log\ p(\boldsymbol{x_0}) + \sum_{k=1}^{K} log\ p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}}) + \sum_{k=1}^{K} log\ p(\boldsymbol{z_k}|\boldsymbol{x_k}) + \sum_{k=1}^{K} log\ p(\boldsymbol{x_k}|\boldsymbol{s_k}, \boldsymbol{h_k}) - \sum_{k=1}^{K} log\ \boldsymbol{E}_{\boldsymbol{x_{k-1}}|\boldsymbol{s_{k-1}},\boldsymbol{h_{k-1}}}[p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}})]] \tag{20}$$

Since $log(E(f(x))) \leq E[log(f(x))]$, we can exchange the log and expectation operations to yield a lower bound for Q, which can be written as

$$Q \geq \boldsymbol{E}_{\boldsymbol{x_{0,\ldots,K}}|\boldsymbol{s_{1,\ldots,K}},\boldsymbol{z_{1,\ldots,K}}}[log\ p(\boldsymbol{x_0}) + \sum_{k=1}^{K} log\ p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}}) + \sum_{k=1}^{K} log\ p(\boldsymbol{z_k}|\boldsymbol{x_k}) + \sum_{k=1}^{K} log\ p(\boldsymbol{x_k}|\boldsymbol{s_k}, \boldsymbol{h_k})] - \sum_{k=1}^{K} \boldsymbol{E}_{\boldsymbol{x_{0,\ldots,K}}|\boldsymbol{s_{1,\ldots,K}},\boldsymbol{z_{1,\ldots,K}}} \boldsymbol{E}_{\boldsymbol{x_{k-1}}|\boldsymbol{s_{k-1}},\boldsymbol{h_{k-1}}}[log\ p(\boldsymbol{x_k}|\boldsymbol{x_{k-1}})] \tag{21}$$

where, the expectation $\boldsymbol{x_{k-1}}$ in the last term, defined by the prediction process, is a function of the model free parameters. The upper bound defined here is helpful given the complexity of finding

the expectation will be the same as other terms of the expectation which appears in EM algorithms defined with standard observation process. The updated parameter set at iteration (r) can be found by maximizing Q if there is a closed form solution for the expectation defined in equation (19) or its lower bound defined in equation (21). The maximization is defined by

$$\{\boldsymbol{\theta}^{(r)}, \boldsymbol{\Omega}^{(r)}, \boldsymbol{\omega}^{(r)}\} = argmax_{\boldsymbol{\theta},\boldsymbol{\Omega},\boldsymbol{\omega}}\, Q_l \tag{22}$$

The optimization step in equation (22) can be calculated analytically or numerically (e.g. gradient descent (22)). After each iteration, a new set of parameters are estimated and the EM routine is stopped when a stopping criteria based on the likelihood growth or parameter changes is satisfied.

3 Result

Having derived the direct-decoder framework, we now discuss how it can be used to decode two different correlates from neural data: (1) a rat's 2-D trajectories, moving through a W-shaped maze, and (2) the inferred communicative intentions of a human study participant engaged in conversation.

3.1 Decoding movement trajectories using the direct-decoder model

In this section, we seek to decode the movement trajectories of a rat moving through a W-shaped maze from the ensemble spiking activity of 62 hippocampal place cells (17). In this experiment, the rat's 2-D position is measured using video tracking software, so we can assess the direct-decoder model performance using the actual position in the maze as ground truth. We used a 15-minute-long recording of the experiment, with a time resolution of 33 milliseconds. The first 85% of the recording ($\sim$13 minutes) was used to train the prediction and state process models, and the remaining 15% ($\sim$2 minutes) was used to test the model's decoding performance.
The rat's position in the maze at time interval k is specified by the state variable $\boldsymbol{X}_k = (x_k, y_k)$, where x_k and y_k represent the rat's x- and y-coordinates. The state process is defined by

$$\boldsymbol{X}_k = \boldsymbol{A}\boldsymbol{X}_{k-1} + \boldsymbol{Q}, \;\; A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \;\; Q \sim N(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}) \tag{23}$$

where the covariance matrix Q is assumed to be diagonal with σ_x^2 and σ_y^2 terms encoding the coordinate variances. We estimate these two variances empirically using the rat's movement during the training session. For the prediction process, we assume the state $\boldsymbol{X}_k$ can be predicted using a linear regression model where the predictor variables are the ensemble spiking activity $\boldsymbol{s}_k \in R^{62}$, each spike train is filtered with a Gaussian window with length 20, at the current and previous time points and the noise process follows a normal distribution. We build two regression models; one for each coordinate, as a function of ensemble spiking activity. The prediction process and its decompsition into two predictor models can be expressed as

$$p(\boldsymbol{X}_k|\boldsymbol{s}_k, \boldsymbol{h}_k) \sim p(x_k|y_k, \boldsymbol{s}_k, \boldsymbol{h}_k) \times p(y_k|\boldsymbol{s}_k, \boldsymbol{h}_k) \tag{24}$$

where h_k represents the history of the ensemble spiking activity from the previous time intervals. Note that because the rat's movement is bounded by the maze, the state process (defined for the rat movement inside the maze) may be misspecified. To address this issue, we add a penalty term to the prediction process that accounts for the toplogy of the maze (17). The revised prediction process is expressed as

$$p(\boldsymbol{X}_k|\boldsymbol{s}_k, \boldsymbol{h}_k) \sim p(x_k|y_k, \boldsymbol{s}_k, \boldsymbol{h}_k) \times p(y_k|\boldsymbol{s}_k, \boldsymbol{h}_k) \times L(x_k, y_k) \tag{25}$$

where $L(x_k, y_k)$ is close to zero for x-y coordinates outside the maze area and one otherwise. To find the optimal number of time points to use in the history term for the x- and y-coordinate regression models, we use a forward modeling selection process. For the y-coordinate model, the null model is defined by the current observation of the spiking activity and the ensemble spiking activity of the previous time points are added recursively. We use a BIC (23) criterion to determine when additional time points do not improve the model fit. For the x-coordinate model, the null model is y_k and the ensemble spiking activity of the current and previous time points was added to find the optimal length of the history term. As was done for the y-coordinate, we use a BIC criterion to find the proper length of the history term for the x-coordinate model. For this dataset, the history term for the x-coordinate considers 12 previous time points ($\sim$ 396 milliseconds) and the y-coordinate includes 18 time points

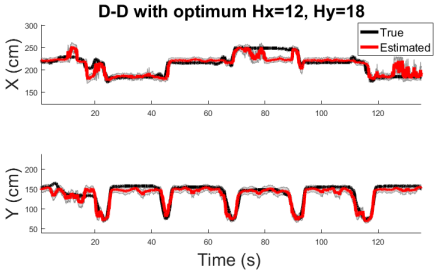


Table 1: Decoded coordinates for the test dataset. D-D result with optimum values for $hx = 12$ and $hy = 18$.

Table 2: Decoder performance comparison.

Method	RMSE	95% HPD
Exact numerical solution.	**12.7**	85.8%
D-D result with hx=1 and hy=1.	17.7	85.9%
D-D result with optimum hx and hy.	13.8	**88.7%**
D-D result with no smoothing criteria.	16.3	88.6%

($\sim$ 594 milliseconds).
To assess the performance of our framework, we decoded test dataset trajectories using four different models: (1) an exact point process decoder model, described in (17), (2) a D-D model with one-step history terms, (3) a D-D model with optimal history term lengths (Figure 1), and (4) a D-D model with optimal history term lengths, but without knowledge of the state transition process (Table 2). Here, model performance is quantified by the root mean squared error (RMSE) and the percentage of time for which the coordinate estimate stayed within the 95% highest posterior density (HPD) region. Our results show comparable performance between the (exact) point process model and the D-D models. The performance of the one-step D-D model suggests the necessity of incorporating more time points in the history term in building a more robust decoder model. The performance result for the prediction process reflects the importance of the prediction step in the filter solution, which uses a much greater portion of ensemble spiking activity compared with the D-D model without one-step prediction element.

3.2 Decoding communicative intent using the dynamical autoencoder

Here we investigate how neural recordings and behavioral readout of a human study participant may be processed to infer an internal, cognitive state. More specifically, our goal is to capture and quantify the participant's intent to verbally communicate. The dataset in this section (unpublished data; Hadjinicolaou, Cash, et al., Massachusetts General Hospital) features aspects of verbal communication between an epileptic participant and their companions, who can include hospital staff, family, friends, and study investigators. Study participants were implanted with intracranial (sEEG) electrodes for clinical monitoring of their epilepsy, for the duration of their stay in the telemetry ward. Neural data were acquired at a sampling rate of 2 kHz using a 128-channel neural signal processor recording system (Blackrock Microsystems, UT) and neighboring channels were re-referenced with a bipolar montage to mitigate volume conduction (24). All spoken dialog within the recording interval was captured and transcribed to yield individual word timings that are synchronized to the neural data. These word timings comprise the behavioral readout for the patient, z_k; we characterize its conditional distribution using a point-process observation model (10). The prediction process builds the relationship between intention state x_k, and neural features $\boldsymbol{s}_k \in^{40}$, whose values consist of spectral power in the 4-8 Hz and 70-115 Hz bands (Chronux) for the subset of bipolar recording channels identified by lasso regularization (MATLAB).
We assume the prediction process follows a normal distribution, where the expected value is a linear function of the neural features [16]. The intention state x_k is characterized by a random walk model (9). The auto-encoder model is defined by

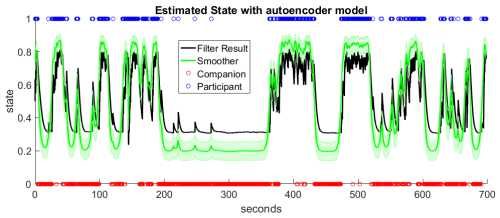
$$x_0 \sim N(m_0; \sigma_0^2) \tag{26}$$

$$x_k|x_{k-1} \sim N(x_{k-1}; \sigma_\epsilon^2), \quad x_{ck}|x_{(c-1)k} \sim N(x_{(c-1)k}; c\sigma_\epsilon^2) \tag{27}$$

$$p(z_k = 1) \approx \lambda_k \Delta, \quad \lambda_k = exp(a_0 + a_1 x_k + a_2 \boldsymbol{p}_k + a_3 \boldsymbol{q}_k) \tag{28}$$

$$\begin{cases} x_k|\boldsymbol{s}_k, \boldsymbol{h}_k \sim N(w_0 + \boldsymbol{w}_1^T \boldsymbol{s}_k + \boldsymbol{w}_2^T \boldsymbol{h}_k, \sigma_v^2) & mod(k, c) = 0 \\ x_k \sim unif_{\alpha_k \to \infty}(-\alpha, \alpha) & Otherwise \end{cases} \tag{29}$$

(a)



(b)

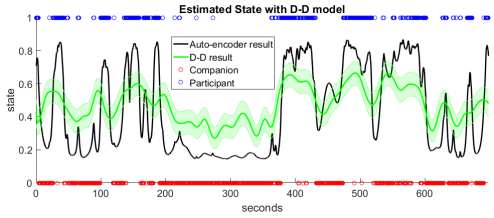


Table 3: Communicative intent estimation using the dynamical auto-encoder model . a. Estimated intent state using both behavioral readout z_k, and neural activity $\boldsymbol{s}_k$. b. Estimated intent state using only neural activity. we use a BIC criterion to find the proper length of the history term for the D-D model. For this dataset, the history term for the intention state considers h=1 previous time points (2 seconds).

Table 4: MLE of the auto-encoder model parameters.

Par.	Initial	Optimized
m_0	0	0.71%
σ_0	0.5	0.23
σ_ϵ	0.5	0.04%
a_0	0	-3.94
a_2	0	1.22
a_3	0	-1.25
σ_v	0.5	0.07
h	0	1
w_0	0	0.25
$\boldsymbol{w}_1$		
$\boldsymbol{w}_2$		

where $\sigma_\epsilon^2, (m_0, \sigma_0^2), (a_0, a_1, a_2, a_3), (w_0, \boldsymbol{w}_1, \boldsymbol{w}_2, \sigma_v^2)$ are the model free parameters – to avoid model identifiability issue, we set a_1 to 1 . $\boldsymbol{p}_k$ and $\boldsymbol{q}_k$ are point-process model history terms, which correspond to the number of spoken words from the patient and physician over the last 400 milliseconds. The processing interval Δ is set to 50 milliseconds; small enough that the probability of more than one word in the interval is negligible. Based on the dataset characteristics behavioral observation and neural activity observation update rates are different; neural activity observation gets updated about every 2 seconds. In equation (28), $c = 2s/(50ms) = 40$ defines the update rate of neural activity. For this problem, the objective is to estimate the state process, which simultaneously maximizes the likelihoods of the behavioral readout and neural recording. To compute maximum-likelihood state estimates, we need to estimate the model parameters, which can be performed using the parameter solution technique discussed in section 2.c (i.e. by maximizing the lower bound of the Q function, Q_l). Since Q_l is the expected value of the full log-likelihood, it can be used in the BIC criteria previously used for the D-D model selection. Figure 3.a shows the intent state estimation given behavioral readout z_k, neural activity $\boldsymbol{s}_k$, and history term $\boldsymbol{h}_k$. The decoding result using only neural activity is shown in Figure 3.b. Our results suggest that the behavioral readout can be used as a surrogate observation to tune parameters of the D-D model, which is the objective of the dynamical auto-encoder model developed here. Table 4 show initial values of the model parameters together with their optimized values at the EM local maximum.

The state process in the auto-encoder model defined in equation (26) to (27) is one dimensional. The auto-encoder model can be expanded to a multivariate state process. For the multivariate state process, we assume there are n-independent state processes which can be linked to the behavior. We then use a sparse prior on the state processes and use a Bayesian Lasso solution to find the subset of state processes representing the behavior and neural activity.

4 Discussion

Here, we introduced a Bayesian decoder model with a discriminant observation process. In the development of the D-D model, we justified the importance of history term, using the decoding task discussed in section 3.A, which is absent in the previous work (15). We then expanded the D-D model to a dynamical auto-encoder model which let us link the behavioral readout and high-dimensional neural recording in the estimation of low-dimensional manifold representing emotional or cognitive

state. We demonstrated the application of the auto-encoder model in a clinical experiment and showed that participant communication intent can be estimated through neural data. We derived model identification and filter solution for the dynamical auto-encoder model in a more general setup. The methodology described here is aligned with the need for a scalable encoder-decoder model with high-dimensional neural recording, and it can be applied in different modalities of neural data. For the auto-encoder model, we briefly discussed how the dimension of the state process can be expanded. For future research, we focus on auto-encoder models with multidimensional state processes and also different categories of the behavior. We also focus on goodness-of-fit analysis to provide a systematic solution in the identification of the optimal state process dimension.

5 Acknowledgments

References

[1] L. A. Jorgenson, W. T. Newsome, D. J. Anderson, C. I. Bargmann, E. N. Brown, K. Deisseroth, J. P. Donoghue, K. L. Hudson, G. S. Ling, P. R. MacLeish *et al.*, "The brain initiative: developing technology to catalyse neuroscience discovery," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 370, no. 1668, p. 20140164, 2015.

[2] J. W. Krakauer, A. A. Ghazanfar, A. Gomez-Marin, M. A. MacIver, and D. Poeppel, "Neuroscience needs behavior: correcting a reductionist bias," *Neuron*, vol. 93, no. 3, pp. 480–490, 2017.

[3] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown, "A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects," *Journal of neurophysiology*, vol. 93, no. 2, pp. 1074–1089, 2005.

[4] L. Paninski, J. Pillow, and J. Lewi, "Statistical models for neural encoding, decoding, and optimal stimulus design," *Progress in brain research*, vol. 165, pp. 493–507, 2007.

[5] T. P. Coleman, M. Yanike, W. A. Suzuki, and E. N. Brown, "A mixed-filter algorithm for dynamically tracking learning from multiple behavioral and neurophysiological measures," *The dynamic brain: An exploration of neuronal variability and its functional significance*, pp. 1–16, 2011.

[6] J. I. Glaser, R. H. Chowdhury, M. G. Perich, L. E. Miller, and K. P. Kording, "Machine learning for neural decoding," *arXiv preprint arXiv:1708.00909*, 2017.

[7] M. R. Rezaei, A. K. Gillespie, J. A. Guidera, B. Nazari, S. Sadri, L. M. Frank, U. T. Eden, and A. Yousefi, "A comparison study of point-process filter and deep learning performance in estimating rat position using an ensemble of place cells," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 4732–4735.

[8] A. Yousefi, M. R. Rezaei, K. Arai, L. M. Frank, and U. T. Eden, "Real-time point process filter for multidimensional decoding problems using mixture models," *bioRxiv*, p. 505289, 2018.

[9] 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," *Journal of Neuroscience*, vol. 18, no. 18, pp. 7411–7425, 1998.

[10] U. T. Eden, L. M. Frank, R. Barbieri, V. Solo, and E. N. Brown, "Dynamic analysis of neural encoding by point process adaptive filtering," *Neural computation*, vol. 16, no. 5, pp. 971–998, 2004.

[11] I. M. Park, M. L. Meister, A. C. Huk, and J. W. Pillow, "Encoding and decoding in parietal cortex during sensorimotor decision-making," *Nature neuroscience*, vol. 17, no. 10, p. 1395, 2014.

[12] M. J. Prerau, A. C. Smith, U. T. Eden, M. Yanike, W. A. Suzuki, and E. N. Brown, "A mixed filter algorithm for cognitive state estimation from simultaneously recorded continuous and binary measures of performance," *Biological cybernetics*, vol. 99, no. 1, pp. 1–14, 2008.

[13] S. Vyas, N. Even-Chen, S. D. Stavisky, S. I. Ryu, P. Nuyujukian, and K. V. Shenoy, "Neural population dynamics underlying motor learning transfer," *Neuron*, vol. 97, no. 5, pp. 1177–1186.e3, Mar. 2018. [Online]. Available: https://doi.org/10.1016/j.neuron.2018.01.040

[14] A. Yousefi, I. Basu, A. C. Paulk, N. Peled, E. N. Eskandar, D. D. Dougherty, S. S. Cash, A. S. Widge, and U. T. Eden, "Decoding hidden cognitive states from behavior and physiology using a bayesian approach," *Neural Computation*, vol. 31, no. 9, pp. 1751–1788, Sep. 2019. [Online]. Available: https://doi.org/10.1162/neco_a_01196

[15] M. C. Burkhart, D. M. Brandman, B. Franco, L. R. Hochberg, and M. T. Harrison, "The discriminative kalman filter for bayesian filtering with nonlinear and nongaussian observation models," *Neural Computation*, vol. 32, no. 5, pp. 969–1017, 2020.

[16] A. Yousefi, A. C. Paulk, T. Deckersbach, D. D. Dougherty, E. N. Eskandar, A. S. Widge, and U. T. Eden, "Cognitive state prediction using an em algorithm applied to gamma distributed data," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015, pp. 7819–7824.

[17] A. Yousefi, A. K. Gillespie, J. A. Guidera, M. Karlsson, L. M. Frank, and U. T. Eden, "Efficient decoding of multi-dimensional signals from population spiking activity using a gaussian mixture particle filter," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 12, pp. 3486–3498, 2019.

[18] Z. Chen *et al.*, "Bayesian filtering: From kalman filters to particle filters, and beyond," *Statistics*, vol. 182, no. 1, pp. 1–69, 2003.

[19] D. Koks and S. Challa, "An introduction to bayesian and dempster-shafer data fusion," DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION SALISBURY (AUSTRALIA) SYSTEMS . . . , Tech. Rep., 2003.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[21] D. A. Van Dyk, "Fitting mixed-effects models using efficient em-type algorithms," *Journal of Computational and Graphical Statistics*, vol. 9, no. 1, pp. 78–98, 2000.

[22] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[23] E. Wit, E. v. d. Heuvel, and J.-W. Romeijn, "'all models are wrong...': an introduction to model uncertainty," *Statistica Neerlandica*, vol. 66, no. 3, pp. 217–236, 2012.

[24] A. M. Bastos and J.-M. Schoffelen, "A tutorial review of functional connectivity analysis methods and their interpretational pitfalls," *Frontiers in systems neuroscience*, vol. 9, p. 175, 2016.