A FACTOR-ANALYSIS DECODER FOR HIGH-PERFORMANCE NEURAL PROSTHESES

G. Santhanam[1], *B.M. Yu*[1,5,6], *V. Gilja*[2], *S.I. Ryu*[1,3], *A. Afshar*[1,4], *M. Sahani*[6], *and K.V. Shenoy*[1,5]

[1]Departments of Electrical Engineering, [2]Computer Science, and [3]Neurosurgery, [4]Medical Scientist
Training Program, [5]Neurosciences Program, Stanford University, Stanford, CA 94305
[6]Gatsby Computational Neuroscience Unit, University College London, London, UK

ABSTRACT

Increasing the performance of neural prostheses is necessary for assuring their clinical viability. One performance limitation is the presence of correlated trial-to-trial variability that can cause neural responses to wax and wane in concert as the subject is, for example, more attentive or more fatigued. We report here the design and characterization of a Factor-Analysis-based decoding algorithm that is able to contend with this confound. We characterize the decoder (classifier) on a previously reported dataset where monkeys performed both a real reach task and a prosthetic cursor movement task while we recorded from 96 electrodes implanted in dorsal premotor cortex. In principle, the decoder infers the underlying factors that co-modulate the neurons' responses and can use this information to function with reduced error rates (1 of 8 reach target prediction) of up to ∼75% (∼20% total prediction error using independent Gaussian or Poisson models became ∼5%). Such Factor-Analysis based methods appear to be effective when attempting to combat directly unobserved trial-by-trial neural variabiliy.

Index Terms— Factor analysis, premotor cortex, brain-machine and brain-computer interfaces, neural prostheses

1. INTRODUCTION

Neural prostheses, which are also termed brain-machine and brain-computer interfaces, aim to substantially increase the quality of life for people suffering from motor disorders, including paralysis and amputation. Neural prostheses translate electrical neural activity from the brain into control signals for guiding paralyzed upper limbs, prosthetic arms, machines and computer cursors. A few research groups have now demonstrated that monkeys (e.g., [1]) and humans can learn to move computer cursors and robotic arms to various target locations simply by activating neural populations that participate in natural arm movements. Although encouraging, even these compelling proof-of-concept laboratory demonstration systems fall short of exhibiting the level of performance needed for many everyday behaviors, and needed to achieve clinical viability.

To address this need for increased performance, we report here a Factor-Analysis- (FA-) based decode algorithm aimed at ameliorating one of the major performance limitations: correlated trial-by-trial neural response variability. Aside from the intrinsic noise present in the neural signaling process, action-potential-emission rates vary across time, be it at a short (trial-to-trial) or long (hours to days) time scale, even when known parameters influencing emission rate are held constant (e.g., upcoming reach direction and extent). Other parameters (factors), including those that may not be known or observable, that are not controlled (e.g., speed of upcoming reach, level of attentiveness, level of fatigue), may influence emission rate and, as a result, contribute "common-mode" variability across the population of neurons.

The need for a new decoder such as the FA-based one presented here was motivated by the observation, made while analyzing data from [1], that the neural response associated with planning to a given reach location changes (modulates) when a part of a high-speed sequence of plans [2]. Motivation for the general type of decoder presented here comes from the considerable effectiveness of a "trial-by-mean" normalization approach that simply divides the response rate of each neuron by the mean response rate across all measured neurons on that trial [3]. To our knowledge this is the first decoder of its kind applied to "plan" activity.

2. METHODS

In [1] we assumed that the spike counts for each neuron were independent once the reach endpoint was specified.[1] This

This work was supported by National Defense Science and Engineering Graduate Fellowships (G.S., B.M.Y., V.G.), National Science Foundation Graduate Research Fellowships (G.S., B.M.Y., V.G.), Medical Scientist Training Program (A.A.), Bio-X Fellowship at Stanford (A.A.), Christopher and Dana Reeve Foundation (S.I.R., K.V.S.), Gatsby Charitable Foundation (M.S., B.M.Y.), NIH (National Institute of Neurological Disorders and Stroke / Collaborative Research in Computational Neuroscience) 5-R01-NS054283-02 (K.S., M.S), and the following to K.S.: Burroughs Wellcome Fund Career Award in the Biomedical Sciences, Center for Integrated Systems at Stanford, NSF Center for Neuromorphic Systems Engineering at Caltech, Office of Naval Research, Sloan Foundation, and Whitaker Foundation. Please address correspondence to `shenoy@stanford.edu`.

[1] For the Gaussian models, this assumption was made to avoid a problem of too little training data when fitting a full covariance matrix. For the Poisson

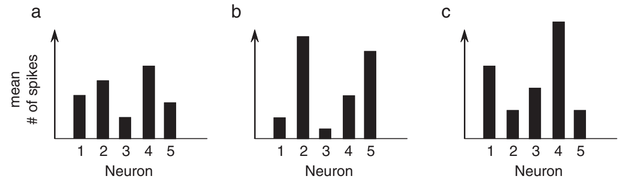

Figure 1: Simple cartoon illustrating how spike counts can co-vary from trial to trial. **a**. Nominal mean spike counts for 5 neurons for a particular reach endpoint. **b**, **c**. Spike counts for two following trials for the same reach endpoint. Activity is elevated or suppressed.

construction implies that there are no high-level factors (e.g., overall attentiveness to the task, reach speed of the upcoming movement, reach curvature) that influence the recorded neural data (other than the reach target itself). If there were, then these factors that are uncontrolled, and often unobserved, would modulate the underlying firing rate of our observed neurons in predictable fashions, thereby inducing measurable neuron-by-neuron correlations in the spike counts that we observe.

In fact, our initial assumptions of conditional independence are certainly gross approximations and are worth revisiting. While one of the primary influences on premotor cortical (PMd) preparatory activity is reach endpoint, there is evidence that activity can depend on several factors other than target location, such as reach speed [4]. If a given model only describes reach endpoint, the model cannot accurately reflect how the firing rate might change if any one of the unaccounted properties (e.g., reach speed) perturbs the underlying firing rate. These fluctuations will appear as response "noise", though the "noise" will be correlated across neurons. For example, consider the cartoon illustration in Fig. 1 where panel a shows the expected number of spike counts of five neurons for a given reach endpoint (e.g., 10 cm rightward reach) and panels b and c show the observed spike counts on two subsequent trials. For panel b, we suggest that the subject might have been planning a slightly faster than average reach. Conversely, for panel c, the subject might have been planning a slightly slower than average reach. Note how the reach speed does not necessarily affect all neurons with the same polarity and magnitude; such heterogeneity was commonly observed.

In reality, we may not know if it is reach speed or some other variable that is causing the trial-by-trial modulation; many different factors can be involved and many of them are simply unobservable (e.g., cognitive attentiveness to the task). We can instead attempt to infer a set of abstract factors for each trial, along with the mapping between the factors and the underlying firing rate of the recorded neurons. A target decoding algorithm can then use this knowledge to avoid mistaking the relatively unimportant trial-to-trial variations as being the signature for an entirely different reach endpoint.

models, independence is a natural consequence of the distribution we chose.

2.1. Latent variable models

Our work is based on "latent variable models" which have been a statistical tool for analyzing empirical data since the early 1900s. Everitt (1984) defines latent variables as "essentially hypothetical constructs invented by a scientist for the purpose of understanding some research area of interest, and for which there exists no operational method for direct measurement. Although latent variables are not observable, certain of their effects on measurable (manifest) variables are observable, and hence subject to study." In our case, the observable (i.e., output) variables are the spiking data from the array of 96 electrodes [1]. The latent variables represent the cognitive state of the subject. Depending on the model setup, they may encapsulate the intended reach endpoint, as well as the uncontrolled and unobserved variables present during the task. We can use the larger number of observed output variables to help triangulate the smaller number of unobserved latent variables of the system.

The two classic methods to reduce dimensionality, and in essence reveal the underlying latent variables, are Principal Components Analysis (PCA) and Factor Analysis (FA). As shown in [5], both of these techniques posit a probabilistic model with the following form:

$$\mathbf{x} \sim \mathcal{N}(\mathbf{0},\, \mathbf{I}) \tag*{$\langle 1 \rangle$}$$

$$\mathbf{y} \mid \mathbf{x} \sim \mathcal{N}(\mathbf{C}\mathbf{x},\, \mathbf{R})\,. \tag*{$\langle 2 \rangle$}$$

The latent state vector, $\mathbf{x} \in \mathbb{R}^{p\times 1}$, is Gaussian distributed with mean $\mathbf{0}$, covariance $\mathbf{I}$, and is unobserved. The output, $\mathbf{y} \in \mathbb{R}^{q\times 1}$, is then generated from a Gaussian distribution. The matrix $\mathbf{C} \in \mathbb{R}^{q\times p}$ provides the mapping between latent state and observations, and $\mathbf{R} \in \mathbb{R}^{q\times q}$ is a diagonal covariance matrix of the output noise process. The vectors $\mathbf{x}_n$ and $\mathbf{y}_n$ denote independent draws from this model over N observations (trials), with $n \in \{1, \ldots, N\}$. For non-zero centered $\mathbf{y}$, the mean across all training trials must be first subtracted before fitting and applying the model.

PCA (or rather, sPCA[2]) and FA can be viewed as effective ways to parameterize a full covariance matrix on the high-dimensional observations $\mathbf{y}$. Indeed, Eqs. 1–2 imply that $\mathbf{y} \sim \mathcal{N}(\mathbf{0},\, \mathbf{C}\mathbf{C}' + \mathbf{R})$. The first term in the covariance, $\mathbf{C}\mathbf{C}'$, attempts to capture the "common-mode" variability across the neural population. The second term, $\mathbf{R}$, represents the independent variability of the spiking process for each neuron. Whereas PCA assumes that this spiking variabililty is identical for each neuron, FA allows different neurons to have different levels of spiking variability. Because the spiking variability is known to vary with the mean spike rate, which may be different for different neurons, we focus on FA for the remainder of this work.

[2] The "sensible" PCA (sPCA) model is a probabilistic approach to PCA and yields the same mapping between latent states and observations as conventional PCA. This is demonstrated by Roweis *NIPS* 1997.

The procedure of system identification, or "model training," requires learning the parameters from the observed data. The observed data includes N trials of $\mathbf{y}$, an independent and identically distributed (i.i.d.) sequence of vectors $(\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N)$ denoted by $\{\mathbf{y}\}$. With the model shown in Eqs. 1–2, we only consider a single reach endpoint. Restricting the fit to only a single endpoint allows for the characterization of the unobserved factors that influence the observations.

The model-fitting procedure is an *unsupervised* problem since the states are "hidden" and therefore unknown; we cannot use known values of the latent variables to help fit the parameters $\mathbf{C}$ and $\mathbf{R}$. The classic approach to system identification in the presence of unobserved latent variables is the Expectation-Maximization (or EM) algorithm. The algorithm maximizes the likelihood of the observed data over the model parameters (i.e., $\theta = \{\mathbf{C}, \mathbf{R}\}$). This results in the parameters that correspond to the highest data likelihood $P(\{\mathbf{y}\} \mid \theta)$. We can then estimate the most likely $\mathbf{x}$ for the observed data $\mathbf{y}$. The fitting procedures are described in [5].

One open question is how to select p, the number of latent dimensions. The objective of model training is to best describe the training data within the constraints imposed by Eqs. 1–2. However, with too many latent dimensions the model training procedure will explain the training data so well through the latent space that there will be unrealistically small amounts of independent observation noise ($\mathbf{R}$). This is contrary to obtaining a simpler model (fewer latent dimensions) with a more reasonable amount of observation noise. We used the standard approach of partitioning data into training and validation sets to assess at which choice of p overfitting becomes a problem. Choosing p is part of the process of "model selection."

2.2. Poisson output model

Standard FA uses a Gaussian noise model but this might not be the most appropriate for our type of data. Recall that our output variables are the spike counts from the recorded neurons and these are naturally nonnegative integers. Furthermore, the means of these data are relatively low (e.g., <10). Hence, such data is not necessarily well-suited for a Gaussian distribution. Neural count data are usually considered to be Poisson or Poisson-like in their distribution.

There are two possibilities to contend with this issue. One approach is to modify the raw data by first applying a square-root to the counts and then centering the data about zero. It can be shown that the approximation error induced when using a Gaussian distribution to fit Poisson data is diminished if the Poisson data is first square-rooted. The transformed data is then used in the standard FA. Results from this approach are reported here. The second option is to alter the generative model to allow for Poisson distributed noise in the output variables. This was also derived, but not reported here since results were comparable to the square-root approach.

2.3. Extensions to accommodate multiple targets

The FA methods described so far are intended to be used with data collected while the subject is preparing to reach to a single target. To use FA to help decode reach endpoints, we tried two different forms of the generative model. The first closely mimics the decode algorithms that we used in [1]. We fit a separate FA model (Eqs. 1–2) for each reach target and decoded by choosing the maximum-likelihood reach endpoint. I.e., we determined which reach endpoint's $P(\mathbf{y})$ reports the highest probability density for that particular trial's neural data. We term this approach FA$_{\text{sep}}$, it can work well in some cases, and we do not consider it further here. The second approach (FA$_{\text{cmb}}$) is to share the same output mapping between target locations and incorporate the effect of reach endpoint through the shared latent space. The difference between these models is subtle but important: in the former the generative model defines a different latent space for each reach endpoint, while in the latter a single latent space is used and the data for each endpoint is separated by their different means in the latent space. An example of how the resulting FA$_{\text{cmb}}$ clusters might appear is shown in Fig. 2. We chose the number of latent dimensions here to be $p = 3$ to allow for convenient plotting of the data.

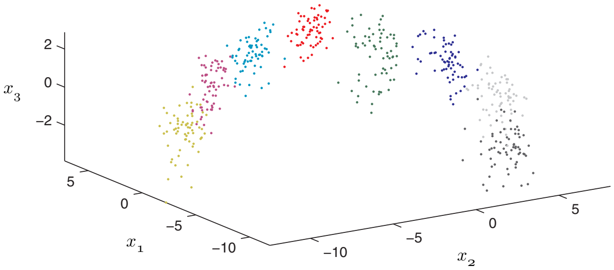

Figure 2: Single latent space example (FA$_{\text{cmb}}$). Each point corresponds to the inferred latent space variable $\mathbf{x}$ for a given trial. The coloring of the data points denotes the endpoint of the upcoming reaching arm movement target. `H20041217`.

3. RESULTS

To assess the potential performance benefits of using FA$_{\text{cmb}}$ to help decode reach endpoints, we turned to a dataset that exhibited trial-by-trial variability and in which shared processes contributed heavily to the overall data variability. In our recent brain-computer interface (BCI) experiments, we presented a mix of BCI trials (short trials, chained rapidly together) and standard reach trials (Fig. 1 in [1]). The BCI trials from multi-hour experimental sessions with two monkeys (`G20040427` and `H20040928`) had individual trial lengths of approximately 400 ms. For the real reaches, most trials had plan periods greater than 400 ms and we discarded any catch trials with timings shorter than this. Therefore, we could analyze neural activity up to ∼400 ms after target presentation regardless of the trial type (BCI versus real reach).

We know from previous data analysis that there can be substantial gain modulation as a chain of BCI trials progresses [2] and that simply normalizing single-trial responses by the average firing rate across the array (on that trial) can considerably improve decode performance [3]. Therefore, we trained on a set of data that included *both* BCI trials *and* reach trials (some decode improvement was also found when considering BCI or reach trials alone). This resulted in an ideal type of dataset. The FA methods could potentially represent the gain modulation as an underlying factor and the target decoder could perhaps benefit from this more accurate model.

Figure 3 shows a comparison between the simple Poisson-based decoder used in [1] and the $\mathrm{FA}_{\mathrm{cmb}}$ decoder. The $\mathrm{FA}_{\mathrm{cmb}}$ model had p = 8 latent dimensions. We have plotted the decode error so as to better illustrate the difference between the two methods. A number of neural count window lengths were tested for each monkey (termed $\mathrm{T}_{\mathrm{int}}$ in [1]), each beginning 150 ms after target onset (termed $\mathrm{T}_{\mathrm{skip}}$ in [1]). The performance differential between simple Poisson-based decoding and $\mathrm{FA}_{\mathrm{cmb}}$ decoding was appreciable. For long window lengths, the performance improvement can be quite substantial (up to ∼15%) in both monkeys.

We can also express the improvements in single-trial decode accuracy in terms of the ITRC (Information Transfer Rate Capacity) metric espoused in [1]. For these BCI datasets, the total ITRC would have increased by approximately 1–1.25 bps if we would have used $\mathrm{FA}_{\mathrm{cmb}}$ during real-time experiments. This constitutes an ITRC increase of 15-20%.

4. DISCUSSION

We investigated the use of a more sophisticated decode algorithm in the hopes that we can achieve higher prosthetic performance. FA techniques were used to help better account for trial-by-trial variations in uncontrolled and unobserved aspects of the prosthetic task. We applied minor extensions to the conventional FA model and adapted it for the purpose of decoding target endpoint. We found that using an entirely separate model for each reach endpoint was not as effective as fitting a single model to the entire dataset. The latter strategy ($\mathrm{FA}_{\mathrm{cmb}}$) requires fewer model parameters and may be less prone to estimation error and overfitting. Surprisingly, the complicated extensions to support Poisson distributions (not reported in detail here) were deemed unnecessary since the Gaussian-based models did equally well (at least for $\mathrm{T}_{\mathrm{int}}$ $>$ approximately 70 ms), and even better in some instances, when data were square-root transformed.

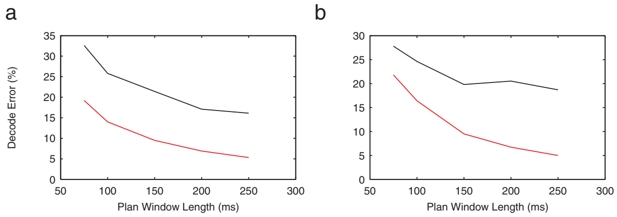

Figure 3: Comparison of simple Poisson-based decoder (black) with the $\mathrm{FA}_{\mathrm{cmb}}$ decoder (red). **a**. Monkey G (dataset `G20040427`). Models were trained on the first 75 trials per condition and tested on the remaining 76 trials per condition in the dataset. **b**. Monkey H (dataset `H20040928`). The training set consisted of 65 trials per condition and the test set had 67 trials per condition.

The utility of the FA methodology was demonstrated with our brain-computer interface (BCI) datasets from [1], where the task design had different operating modes (BCI vs. real-reach trials). This resulted in much more shared variability and $\mathrm{FA}_{\mathrm{cmb}}$ was able to consistently and significantly outperform the conventional methods. For a clinical prosthetic setup, the situation of mixing BCI and real reach trials would not be realistic since the patient would be paralyzed. However, even for a clinical BCI the set of actions available to the patient may be so heterogeneous that there may be underlying factors that significantly modulate the outputs, even though the factors are irrelevant to the task itself. If this is the case, FA can be one tool by which the system designer can combat performance limitations and degradation.

5. REFERENCES

[1] G. Santhanam, S. I. Ryu, B. M. Yu, A. Afshar, and K. V. Shenoy, “A high-performance brain-computer interface,” *Nature*, vol. 442, pp. 195–198, 2006.

[2] R. S. Kalmar, V. Gilja, G. Santhanam, S. I. Ryu, B. M. Yu, A. Afshar, and K. V. Shenoy, “PMd delay activity during rapid sequential movement plans,” *Program No. 519.17. 2005 Abstract Viewer/Itinerary Planner. Washington, DC: Soc. for Neurosci.*, Nov. 2005.

[3] V. Gilja, R. S. Kalmar, G. Santhanam, S. I. Ryu, B. M. Yu, A. Afshar, and K. V. Shenoy, “Trial-by-trial mean normalization improves plan period reach target decoding,” *Program No. 519.18. 2005 Abstract Viewer/Itinerary Planner. Washington, DC: Soc. for Neurosci.*, Nov. 2005.

[4] M. M. Churchland, G. Santhanam, and K. V. Shenoy, “Preparatory activity in premotor and motor cortex reflects the speed of the upcoming reach,” *Journal of Neurophysiology*, vol. 96, pp. 3130–3146, 2006.

[5] S. Roweis and Z. Ghahramani, “A unifying review of linear gaussian models,” *Neural Comput*, vol. 11, no. 2, pp. 305–345, 1999.