Neural Decoding of Monkey's Cortical Activity to Target-Directed Movements Prediction

CORALINE BEITONE[1], TIN LAM DOROTHY CHENG [1], AND YU HEI MARCO CHENG [1]

[1] Dept. of Bioengineering, Imperial College London, London, UK.

Abstract

This report investigated developing a neural decoder for accurate prediction of 8-target-directed hand movements of a monkey based on recordings from 98 motor cortex neural units. Multi-SVMs were combined for an 8-class classification of target directions, followed by the investigation of three trajectory prediction algorithms, averaging estimation (AE), Kalman Filtering (KF) and Linear Regression (LR). Peristimulus time histogram analysis was performed to reduce neural data dimensions for extracting informative features. Results showed that the direction classifier achieved an accuracy of 98.1% with KF being the most suitable and robust decoder that takes into account measurement uncertainties, having a root mean square error of 14.374. AE and LR were overfitted to this specific task and might not be robust in real-life applications.

1. INTRODUCTION

In recent years, the development of neural decoding technology has opened up new avenues for innovation in the field of prosthetics and rehabilitation. By translating neuromuscular signals into motor commands, their implementations in a brain-machine interface (BMI) greatly help patients with severe neuromuscular disabilities to compensate for their lost functions. As the demand for effective devices continues to grow, it has become crucial to design a decoder that can accurately drive a hypothetical prosthesis while supporting the practicalities of its real-time use [1]. Given this perspective, implementing a robust, accurate and causal neural decoder in monkeys is a crucial initial step to guide its optimization, and pinpoint limitations to be addressed for a potential extension to clinical settings. In this report, we designed a decoder to predict the target-directed hand movements of a monkey based on the analysis of its motor cortex activity. Several methods, including Support Vector Machine (SVM), linear regression, Kalman filtering and basic averaging, were explored to produce 3 decoder models. This paper reviews and compares their performances and examines their applications in a clinical setting in order to suggest the most appropriate one for an advanced prosthetic solution adhering to the constraints of real-world applications.

2. METHODS

A. Decoder structure and dataset

To form the pertinent dataset for the intended prediction task, a monkey was asked to repeatedly perform reaching movements toward 8 different targets on a fronto-parallel screen as shown in Fig 1. For each reaching direction, the activity of 98 motor cortex neural units was recorded 300 ms before movement onset until 100 ms after movement ends. In addition, the monkey's 3D arm trajectories were simultaneously tracked during the motion. The experiment was repeatedly performed 100 times for each target to gather an extensive dataset. Based on this data structure then, a forward strategy to design the neural decoder would consist first in providing an estimate of the reaching angle chosen by the monkey. This estimation is treated as a multi-label classification problem. Using the prior knowledge of the reaching angle, the set of possible trajectories can be then limited to the one associated with the predicted direction. As most of the arm motion was contained in the (X,Y) plane of the screen, the prediction task can be reduced to a 2D continuous estimation of the X and Y hand's coordinates at regular time intervals $\Delta t = 20\ ms$. From these general guidelines, the following section then reviews the machine learning methods and statistical approaches explored to build the intended neural decoder.

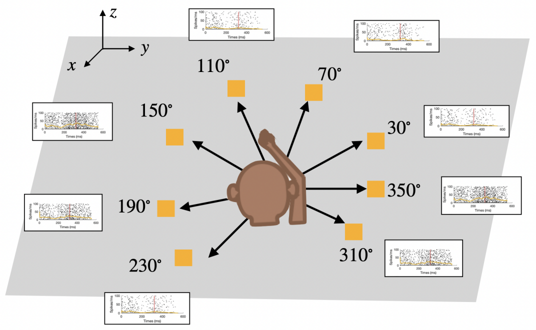

Fig. 1. Experimental settings: the monkey is asked to reach repeatedly one of the 8 directions. Neural activity and 3D hand coordinates are recorded for each trial.

B. SVM-based direction classifier

In order to ascertain the target direction among the 8 potential reaching directions that the monkey aims for, we utilized a Support Vector Machine (SVM) approach. This algorithm is a supervised binary classifier that looks for a separating hyperplane $\tau = \mathbf{w}^T\mathbf{x} + b = 0$ for which the margin between the samples of two classes of data is maximised. Its principle can be illustrated in Fig 2. Given a set of d-dimensional input vectors $x_j \in \mathbb{R}^d$ and their corresponding labels $y_j = \pm 1$, the algorithm uses the closest data points (support vectors) from both classes to guide the separation of the data. By setting the two margins lines ($\mathbf{w}^T\mathbf{x} + b = 1$, $\mathbf{w}^T\mathbf{x} + b = -1$) passing through the support vectors, the algorithm then learns the parameters $\mathbf{w}$ (weights) and b (bias) which maximise the distance $\frac{2}{||\mathbf{w}||}$ from both classes. Based on the optimal solution $(\mathbf{w}_*, b_*)$, a decision line is then placed in the middle of the margin, and the classification of any input vector $\mathbf{z}$ is given by $class(z) = sgn(\mathbf{w}_*^T\mathbf{z} + b_*)$ [2].

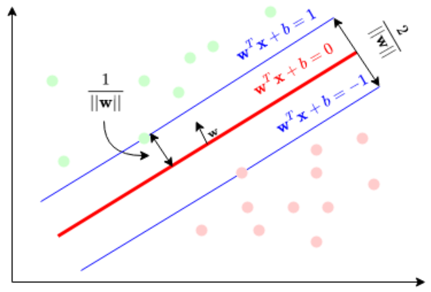
$\mathbf{w}^T\mathbf{x}+b=1$
$\mathbf{w}^T\mathbf{x}+b=0$
$\mathbf{w}^T\mathbf{x}+b=-1$
$\frac{2}{||\mathbf{w}||}$
$\frac{1}{||\mathbf{w}||}$
$\mathbf{w}$

Fig. 2. Visual illustration of SVM

To fully exploit SVM in the particular case of our classification problem, first, a kernel trick had to be used to pre-transform the nonlinearly separable spike train recordings, into a sufficiently high dimensional space where two classes are linearly separable. As a common choice for the kernel function, the radial basis function (RBF) was chosen, defined as $G(x_1, x_2) = exp(-||x_1 - x_2||^2)$. Next, as the direction prediction is an 8-class classification problem, we broke down this multi-classification into 4 binary classification problems solved by 4 SVMs. Table 1 shows the combination of 4 SVMs for the 4 binary classification tasks. During training, with the assumption that pre-movement spiking data encode the monkey's chosen direction, we used the average firing rate as spike count over the first 320 ms as the input data for every neuron at each trial. They are fed into the SVM in two groups, each group containing data from 4 directions as categorized in Table 1. Each SVM then learns to separate these two groups of input data, with the hyperplane parameters stored for testing. In testing, the test data are fed into all 4 SVMs for classification between either Class 0 or Class 1. The reaching direction can then be determined using IF-ELSE rule-based approach. For example, test data with direction 1 should be classified into Classes $[0, 1, 1, 1]$ with the 4 corresponding SVMs. Then, the final decision of direction 1 is made through a rule-based system, allowing simple and efficient multi-class classification.

SVM	Class 0	Class 1	SVM	Class 0	Class 1
1	1, 2, 3, 4	5, 6, 7, 8	**3**	3, 4, 5, 6	7, 8, 1, 2
2	2, 3, 4, 5	6, 7, 8, 1	**4**	4, 5, 6, 7	8, 1, 2, 3

Table 1. Modelling the 8-class classification problem by a combination of 4 binary SVM models

C. Trajectories decoding

Once the reaching direction is determined, we are left with the regression problem of continuously estimating the monkey hand's coordinates every time bin $t_k = 20\ ms$. For that, we first implemented a neural data dimensionality reduction, followed by three methods, average trajectory estimation, Kalman filtering and linear regression with the least squares method, operating on the training data starting at $t = 320ms$, when the monkey's motion begins. Details of their implementations are described below.

C.1. Neural Dimensionality reduction based on Peristimulus time histogram (PSTH) analysis

Unlike SVM, which can support some degrees of outliers, trajectory regression is oversensitive to noisy recordings which can cause significant error accumulations in the continuous estimation process. Therefore, a finer selection of the initial neurons' subset, relevant to the monkey's hand movement, needs first to be conducted. To do this, a visual analysis of the neurons' PSTH was preferred over unsupervised methods such as PCA, to maintain a level of control and interpretability over the data. The manual selection process was conducted by privileging neurons showing a clear change (increase or decrease) of activity at the onset of the movement ($t \approx 300\ ms$) and during the first 200 ms of the motion. This analysis was performed for each reaching direction, leading to extracting 8 neural subsets. Then, using a trial-and-error approach during the testing phase of the decoders, these ensembles were further reduced to a common set of 20 relevant neurons, enabling the decoders to show their best performance in terms of root-mean-squared-error (RMSE) values.

C.2. Heuristic model - Averaging Estimation

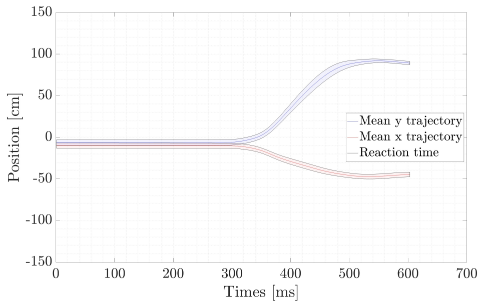

Fig. 3. Mean trajectories over trials for a representative direction ($\theta = 110°$) with standard deviations.

The analysis of the monkey's hand trajectories revealed that their variances over trial were relatively small for any reaching angle θ (Fig 3). We thus suggested, as a first approach, to use the empirical mean as a reasonable estimator of the actual trajectories. A heuristic model based on the average of the X and Y coordinates from the training dataset was thus implemented to assign each reaching direction its corresponding averaged trajectory. Using the direction estimated by SVM, the prediction of the X and Y coordinates over time then simply reduced in retrieving the mean trajectory associated to that direction.

C.3. Kalman Filtering

Kalman Filtering is a specific instantiation of Bayesian filtering with the assumption that both the dynamics and the measurement probabilities are linear Gaussian models. Its probabilistic inference can be visualised with the following Dynamic Bayesian Network (Fig 4).

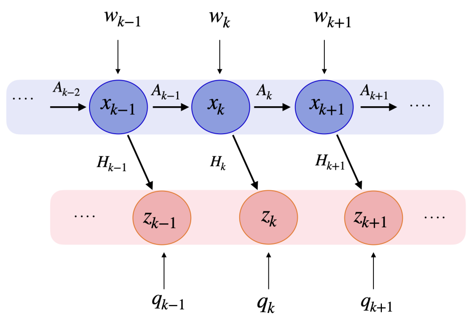

Fig. 4. Graphical model of a Linear Dynamical System

In the decoding task, the linear Gaussian distribution assumption implies that the state vector $\mathbf{x_k} = [x(t_k), y(t_k), v_x(t_k), v_y(t_k)]$, corresponding to the monkey's hand position and velocity at any time bin t_k, are linearly related to the observations vector $\mathbf{z_k} \in \mathbb{R}^N$ containing the firing rates of the N selected neurons at time t_k. Mathematically, the following relations can be derived:

$$\begin{cases} z_k = H_k x_k + q_k \ , k \in \{1, ..., M\} \quad (generative\ model) \\ x_{k+1} = A_k x_k + w_k \ , k \in \{1, ..., M\} \quad (state\ equation) \\ q_k \sim \mathcal{N}(0, Q_k) \\ w_k \sim \mathcal{N}(0, W_k) \end{cases}$$

where M is the number of time steps in the trial, $H_k \in \mathbb{R}^{N\times4}$ is a matrix linearly relating hand states to firing rates, $A_k \in \mathbb{R}^{4\times4}$ is the coefficient matrix, q_k and w_k are respectively the observation and the system noise with their associated covariance matrix $Q_k \in \mathbb{R}^{N\times N}$ and $W_k \in \mathbb{R}^{4\times4}$. By assuming that A_k, H_k, W_k, Q_k are time-independent, these model parameters can then be estimated from the training data using a least squares estimation:

$$\underset{A}{argmin}\left(\sum_{k=1}^{M-1} \|x_{k+1} - Ax_k\|^2\right) \ and \ \underset{H}{argmin}\left(\sum_{k=1}^{M} \|z_k - Hz_k\|^2\right)$$

where $\|.\|$ is the L2 norm. Given A and H, W and Q can then be estimated, details are provided in [3]. In this particular decoding task, 8 set of models parameters (A_j, H_j, W_j, Q_j), $j \in \{1, .., 8\}$ had to be derived. For that, each training data $(\mathbf{x}, \mathbf{y}, \mathbf{v_x}, \mathbf{v_y})_{i,j}$ from trial $i \in \{1, .., 100\}$ and direction j, was used to produce a set of 'one-trial' model parameters $(A_{i,j}, H_{i,j}, W_{i,j}, Q_{i,j})$. The resulting set of model parameters (A_j, H_j, W_j, Q_j) was then obtained by averaging over trials.

During the testing phase, the estimation of the state vector $\mathbf{x_k}$ at time t_k, and for the direction j, is then performed using the two following steps:

1) The *a priori* step where we obtain a first estimate of $\mathbf{x_k}$ from the state equation: $\mathbf{\bar{x}_k} = A_j \mathbf{\hat{x}}_{k-1}$. The a priori error covariance matrix, $\mathbf{P_k^-}$, is also computed to assess the performance of the estimation: $\mathbf{P_k^-} = A_j \mathbf{P_{k-1}} A_j^T + W_j$ with $\mathbf{P_k}$ the *a posteriori* estimate error covariance matrix. For the initialisation, we let the prediction equal to the real initial condition, that is $\mathbf{P_0} = 0$ and $\mathbf{\hat{x}_0}$ being the true initial hand's dynamics.
2) The *a posteriori* step where the prediction $\mathbf{\bar{x}_k}$ is corrected using the information from the firing rate at time t_k: $\mathbf{\hat{x}_k} = \mathbf{\bar{x}_k} + K_k(\mathbf{z_k} - H_j \mathbf{\bar{x}_k})$ with $K_k = \mathbf{P_k^-} H_j^T (H_j \mathbf{P_k^-} H_j^T + Q_j)^{-1}$ the Kalman gain. The a posteriori error covariance matrix is also updated by $\mathbf{P_k} = (I - K_k H_j)\mathbf{P_k^-}$.

C.4. Least Squares Linear Regression

The Least Squares Linear Regression Model is a widely employed statistical method for estimating the relationship between a dependent variable and one or multiple independent variables. This method is founded on the concept of reducing the total squared discrepancies between the predicted values and the actual observed data. The linear regression model takes the form:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \epsilon \quad \textbf{(1)}$$

where y is the dependent variable (hand movement velocities), $x_1, x_2, \ldots, x_n$ are the independent variables (firing rates), $\beta_0, \beta_1, \ldots, \beta_n$ are the coefficients to be estimated, and ϵ is the bias term [4]. The least squares method aims to find the coefficients that minimize the residual sum of squares (RSS):

$$\text{RSS} = \sum_{i=1}^{N} (y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_n x_{in}))^2 \quad \textbf{(2)}$$

It is used to approximate the relationship between velocities $(v_x(t_k), v_y(t_k))$ of the hand movement and neurons' firing rates. For this decoder, we trained 8 distinct least squares regression models. For each direction, the velocities data and all neurons' firing rates of each trial are served as the input data to train each model separately, aiming to find the model parameters that minimize the RSS. The 8 sets of model parameters are stored for prediction. During the testing phase, we input the firing rates of the test spikes calculated and utilize the pre-trained model parameters to predict the hand movement velocities at each time step. Hand's positions $x(t_k), y(t_k)$ are then retrieved from multiplying the time bin (20ms).

3. RESULTS

To train and implement the neural decoder, we randomly split the dataset into 80% training data and 20% testing data. The multi-SVM trajectory direction classification task showed an accuracy score of 0.981. Averaging estimation, Kalman Filtering (KF) and Linear Regression (LR) were then used for trajectory prediction with their performances compared in terms of root-mean-square error

(RMSE) value. Their RMSEs are 11.029, 14.374 and 18.953 respectively, with the actual and predicted trajectories plotted in Figure 5.

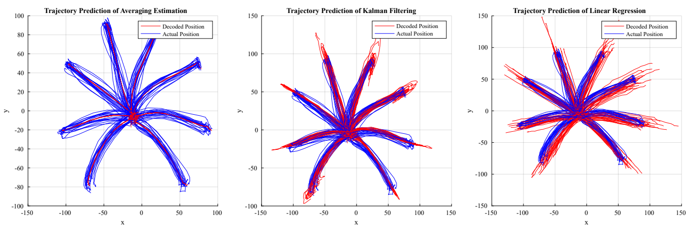

Fig. 5. Predicted and actual trajectories from averaging (RMSE = 11.029), KF (RMSE = 14.374) and LR (RMSE = 18.953)

4. DISCUSSION

From the results, our multi-SVMs algorithm, combined with the kernel trick, is able to accurately predict directions for any trajectory. For trajectory prediction, the averaging method gives the lowest RMSE compared to Kalman filtering and linear regression, whose performances are affected by trajectories overshooting. This discrepancy may come from the variability of the neural recordings used by these latter methods, which, as opposed to the uniformity of the observed trajectories, can in turn cause variability in the prediction output. We also acknowledge that a "manual" selection of neurons may not be optimal and certainly alters the performance of the prediction of Kalman filter and linear regression. Future work would be to explore accurate neurons selection techniques. In particular, several tools listed in [1] could be interesting to consider, such as Cao et al.'s [5] use of mutual information to determine which neurons modulated for the reaching direction versus hand configuration during the movement.

However, it is noteworthy that a strict comparison of the decoders' performance in terms of RMSE value would be insufficient to evaluate the real capabilities of these models. Although averaging shows the best RMSE value, its implementation is only feasible due to the large consistency between training and testing data. Fundamentally, it is not a proper "decoding" method, since it simply outputs the average trajectory observed in the training data without considering the neural recordings. Figure 5 illustrated this where, despite the varying starting positions of the monkey, all predicted trajectories are the same for every direction. It is obvious that it would not be robust in real-life situations where the movement might be much more complex than a simple linear trajectory.

A similar comment can be made for the Least Squares Linear Regression Model. This algorithm is optimal for predicting simple hand trajectories as in this specific task, but might not be suitable for real-life situations where hand movements could be not linear. Alternatively, one can also imagine an algorithm applying a nonlinear transformation to a piecewise linear prediction of the trajectory. This would eventually amount to train a neural network, but the temporal complexity of the decoder would then be reduced, making its use for a real-time operation difficult.

As a result, Kalman Filtering is the most suitable decoder for trajectory prediction. Unlike the typical use of machine learning algorithms that provide maximum likelihood estimates of the decoded variable, Bayesian decoding approaches use the probability distribution over all possibilities for the decoded outputs, providing information on the estimate's uncertainty. In this regard, Kalman filter gives a measure of uncertainty, accounts for noise in the data, and can predict more natural and complex trajectories, qualities which are highly desirable for neural prosthesis applications. Its only weakness could be the assumption of a Gaussian-distributed neural firing, where the actual distribution appears to be Poissonian. A potential solution would be to extend the linear Kalman to a switching Kalman which accounts for the non-Gaussianity of the neural observations as reported in [6].

5. CONCLUSION

To decode the target-directed trajectories from the monkey's neural spiking data, we used a multi-SVMs classification algorithm to first classify the reaching direction from pre-motion spiking data. Three methods were then implemented for trajectory prediction. Considering the RMSE performance and the practicability of these models in a clinical setting, Kalman filter appeared as the most relevant decoder for neural prostheses, by maintaining an uncertainty representation and having the ability to recursively predict more natural and complex movements.

6. ATTRIBUTIONS

Each member equally contributed and helped write the report, with the allocation of work listed below:
Coraline: Kalman, Averaging, Select Neurons, Discussion
Dorothy: SVM, Kalman, Results, Discussion
Marco: Introduction, Linear Regression, Discussion

Github Page

REFERENCES

1. Z. Li, "Decoding methods for neural prostheses: where have we reached?" Front Syst Neurosci (2014).
2. N. Christianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods* (Cambridge University Press, 2000).
3. W. Wu, M. Black, Y. Gao, E. Bienenstock, M. Serruya, and J. Donoghue, "Inferring hand motion from multi-cell recordings in motor cortex using a kalman filter," (2002).
4. K. Ajitesh, "Ordinary least squares method: Concepts examples," (2023).
5. Y. Cao, Y. Hao, Y. Liao, K. Xu, W. Y, S. Zhang, and et al, "Information analysis on neural tuning in dorsal premotor cortex for reaching and grasping." Comput.Math.MethodsMed (2013).
6. W. Wu, M. J. Black, D. Mumford, E. Bienenstock, and J. P. Donoghue, "Modeling and decoding motor cortical activity using a switching kalman filter," IEEE Transactions on Biomed. Eng. (June 2004).