Multi-Regression Decoder with Auxiliary Classifiers

1st Peter Guan 2nd Kexin Huang 3rd Zhongjie Zhang 4th Haonan Zhou

***Abstract*—The current neural decoding algorithms for brain machine interfaces (BMIs) largely uses regression methods to predict the arm position from neuronal ensemble activity. In this regard, we aimed to decode reaching target and movement state information independently using kNN, combined with a basic ridge regression method. Throughout performance comparison between the proposed decoder and the decoders without target/state information, the former performs better than other decoders and the corresponding reconstruction of hand trajectories present a more reasonable path.**

***Index Terms*—kNN, ridge regression, neural decoder**

I. Introduction

As neural signals are widely developed, various algorithms have been investigated to decode arm kinematic information (e.g. position, speed and angular) from recorded motor cortical activities [1, 2]. Meanwhile, both linear and non-linear decoding methods show promised results [3, 4]. Whereas non-linear methods can be better at capturing the relationship between neural activity and arm kinematic information compared to linear fitting methods [5]. However, due to higher computation costs and more complex software engineering accompanying the power of non-linear regression methods, the linear regression methods are often applied in the real-time reconstruction of arm movement trajectories [4]. Nevertheless, ordinary linear regression methods based on least-squares tend to overfit when dealing with high-dimensional, low-ranking and noisy data [6, 7]. Whereas Ridge Regression (RR) can address the issues above via a modified least-squares estimation method [8]. By adding a Tikhonov regularisation constraint to the least-squares cost function to constrain the l2 norm of the regression coefficients, RR discards the unbiased nature of the least-squares method and obtains more realistic and reliable regression coefficients at the cost of some loss of information [7].

In addition, for goal-directed movement, delayed activity (i.e., neural activity associated with movement preparation) has been demonstrated to indicate the upcoming goal and improve the accuracy of the decoded trajectory [9]. Moreover, in order to achieve continuous decoding of the arm trajectory, [10] trained support vector machines (SVM) to estimate the unknown movement state, indicating the significance of the movement state information for continuous decoding.

We propose a multi-model decoder based on ridge regression with two auxiliary classifiers to achieve real-time 2D trajectory reconstruction of the monkey arms. By combining simple RR models, each of which is accurate within a specific target and movement state, the proposed decoder accurately estimates the trajectory of the target-directed reaching movement with multiple targets and still keeps a relatively low computational cost. The k-nearest neighbours (kNN) classifier was chosen as the auxiliary classifier since it excels other Machine Learning classifiers due to its easy implementation and new data-friendly advantages [11]. Furthermore, the functions of these two auxiliary classifiers are capturing the information of the upcoming target from the delay activity of neurons (i.e., the first 300 ms of a spike train) and predicting the current movement state of the arm (i.e., motion or rest), respectively. In addition, the proposed decoder is compared with simple linear regression methods, showing the superiority of the proposed decoder.

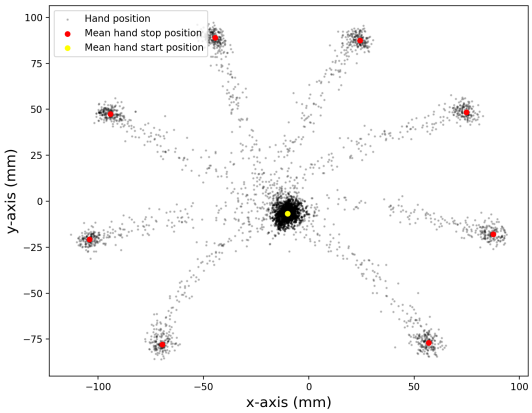

Fig. 1. Trajectory of center-out task. Red dots represent the mean stopping positions of monkey's hand trajectory at each target and the yellow dot at the center of the screen indicates the mean starting position.

II. Data Description

We analyzed behavioral and neural data in a monkey recorded during the center-out task, provided by the laboratory of Prof. Krishna Shenoy at Stanford University. The neural data contains spike trains recorded from 98 neural units while the monkey reached 100 times along each of eight radial locations in $\{30, 70, 110, 150, 190, 230, 310, 350°\}$, where the radial locations are the stop position of the monkey hand, which is approximately 98.30 ± 2.96 mm from the hand start position. The behavioral data contains the monkey's arm trajectory on each trial (Fig. 1). On each trial, with time bins = 1 ms, both the neural data and the arm trajectory are taken from 300 ms before movement start until 100 ms after movement stop. Here, the trial data was divided into training and testing sets in an 80:20 ratio.

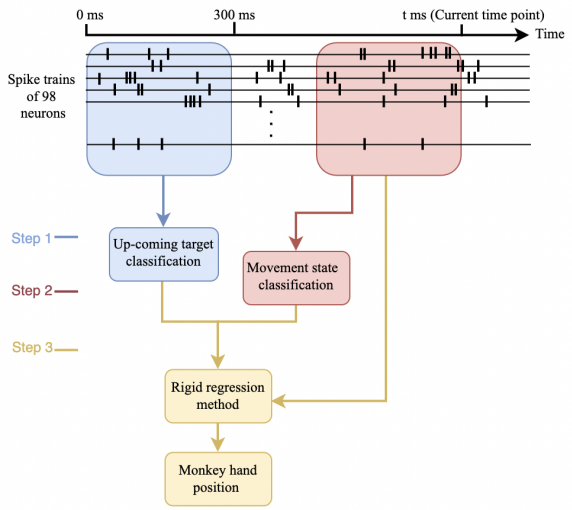

Fig. 2. Proposed decoder anatomy. At step 1, the up-coming targets are classified using the firing rate summarized in the first 300 ms spike train. Step 2 indicates the movement states are classified using the firing rate summarized between t(current time point) and t - 300 ms. In step 3, the monkey's hand position at time t is predicted through the rigid regression model based on the classified target and state.

III. Methodology

The decoder we proposed here is a multi-model decoder since it contains 16 simple RR models. The number of RR models is determined by eight different reaching targets and two movement states (i.e., rest or motion state) in each arm reaching movement. Moreover, each RR model maps from neural activity to hand position given a specific target and movement state. Moreover, two kNN auxiliary classifiers are established to predict the corresponding reaching targets and movement states, respectively, during training. During the test, to continuously decode hand position $\mathbf{p}_t$ (i.e., a vector containing $x-$ and $y-$ axes information) at each time point t, where t is from 320 ms to the end of the trial with an interval of 20 ms, the 98 neurons spike trains from 0 ms to the current time point were taken.

Fig.2 summaries the working pipeline for the proposed decoder. Therefore, reaching target classification will be introduced in Section III-A. Afterwards, movement state classification is illustrated in Section III-B. Next, Section III-C presents the regression architecture. Finally, model performance evaluation metrics are being introduced in Section III-D.

A. Reaching Target Classification

kNN is implemented as the target classifier. The input here is the firing rate of 98 neurons, which is the summation of the spikes in the first 300 ms of each spike train (i.e., neural activity before movement onset). Furthermore, the distance metric is core when implementing the kNN algorithm. Due to the high dimensionality of our data (i.e., firing rate of 98 neurons), Euclidean distance is employed to measure distance statistics information for the reaching target classifier, which is defined by,

$$D_j^{(i)} = \sqrt{\sum_{p=1}^{n}(x_j^p - x_i^p)^2}, \tag{1}$$

where n is the number of valid neurons, underscripts i and j represent the index of candidate in training and testing data, respectively, and x_i^p is the firing rate of $p-\text{th}$ neuron. Accordingly, the predicted reaching target of test candidate i can be determined by the following four-fold steps,

- Calculate the distances between test candidates i and each training sample using (1).
- Sort distances ascendingly and pick up top **30** neighbours.
- Count reaching target category frequency for the aforementioned 30 candidates.
- Use a plurality vote of these neighbors to gain the preferable reaching target M_i of candidate i, where $M_i \in \{30, 70, ..., 350°\}$.

B. Movement State Classification

Similar to Section III-A, kNN is employed to predict the movement state of the monkey's arm $S_k \in \{rest, motion\}$. The difference is that the firing rate used here is obtained by the summation of spikes in the range of $[t-300, t]$, where t is the current time point. At the training phase, the true movement state can be labelled by,

$$S_k = \begin{cases} rest, & t \in [T-100,\ T], \\ motion, & t \in [T-300,\ T-100), \end{cases} \tag{2}$$

where T is the time length of the corresponding spike train.

C. Regressions

The mapping between estimated hand position $\hat{\mathbf{p}}_\mathbf{t}$ and firing rate x_i^t at time t can be modelled by regression methods in the light of the two preceding auxiliary classifiers, which is defined by,

$$\hat{\mathbf{p}}_\mathbf{t} = f(x_i^t | M_i, S_k), \tag{3}$$

where $f(\cdot)$ denotes regression model mapping function. Generally, linear regression is qualified for this project. In addition, the firing rate generation mechanism is the same as defined in Section III-B. Hence, the cost function solved by Least Square Error (LSE) is given by,

$$L_1 = (\mathbf{p}_\mathbf{t} - \hat{\mathbf{p}}_\mathbf{t})^T(\mathbf{p}_\mathbf{t} - \hat{\mathbf{p}}_\mathbf{t}) \ \Rightarrow f(\cdot) \equiv \min\ L_1, \tag{4}$$

where $\mathbf{p}_\mathbf{t}$ indicates true hand position of time t.

Although the LSE is an unbiased method to fit data for linear regression, it could introduce significant variances compared to true and predicted values in multicollinearity problems, which is proved by the Variance Inflation Factor(VIF). And Ridge Regression is able to fix such issue by employing a regulation as shown here,

$$L_2 = L_1 + \lambda\omega^T\omega \ \Rightarrow f(\cdot) \equiv \min\ L_2, \tag{5}$$

where ω is the regression weights of $f(\cdot)$, and λ is the regulation parameter with the range of $[0, 1]$, which is 0.9 here.

TABLE I
CLASSIFIER PERFORMANCE.

Classifier	Accuracy	FPR
Reaching Target	97%	4×10^{-3}
Movment State	91%	0.09

TABLE II
MODEL PERFORMANCE, WHERE 'RMSE' IS DEVELOPED BASED ON THE CLASSIFICATION LABELS FROM THE CORRESPONDING EMPLOYED CLASSIFIERS, WHEREAS 'RMSE WITH TRUE LABELS' ASSUMES 100% CLASSIFICATION ACCURACY.

Employed Classification	Regression Model	RMSE (mm)	RMSE with True Label (mm)
Targets + Movement States	Linear	11.51 ± 0.40	11.88 ± 0.33
	Ridge	$\mathbf{7.95 \pm 0.25}$	8.09 ± 0.22
Targets	Linear	12.09 ± 0.47	12.07 ± 0.38
————	Linear	19.85 ± 0.51	19.85 ± 0.73

D. Evaluation and Metrics

Classifier performance is generally calculated by comparing the predictions generated by the classifier on the test dataset with the true class labels of the instances from this dataset [12]. Normally, accuracy is a good choice for this project. However, it is worth paying more attention to False Positive (FP) samples here since a candidate will skip from the true hand position trajectory once fed into an erroneously chosen regressor. Hence, average False Positive Rate (FPR) [13] becomes another classification metric to measure classifier performance, which is derived as,

$$FPR = \frac{FP}{FP + TN}, \tag{6}$$

where TN is the true negative sample in classification. Moving on to the hand trajectory estimation performance, it is found that the closer between two trajectories (i.e., the predicted hand position trajectory and the true one), the less error that the models can make. And Root Mean Square Error (RMSE) is a good estimator for measuring the standard deviation of the distribution of such errors, which is defined by,

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{p_t} - \hat{\mathbf{p}}_\mathbf{t})^\mathbf{2}}, \tag{7}$$

where N is the total trail number.

IV. RESULT AND DISCUSSION

A. Decoder Performance

Table I shows the performance of two auxiliary classifiers, where the accuracy of both two classifiers achieve as higher as 90%, demonstrating the feasibility of predicting movement target and state from the neural activity. The FPR of the target classifier is 4×10^3, which is negligible so that some wrong estimating reaching directions would not increase final RMSE remarkably. In comparison, a little large FPR was shown in the movement state classifier, which might be since the monkey's hand may not have entirely stopped in the last 100 ms.

From Table II and Fig 3, reconstructed trajectories obtained by only linear regression are inaccurate and unnatural, and the corresponding RMSE is 19.85 mm. The cause of errors might be that linear regression does not accurately capture the non-linear relationship that might exist between neural activity and multi-target oriented movements of the arm.

With the use of information on upcoming targets, the RMSE is reduced by 39.1% and the reconstructed trajectory becomes smoother and more directly towards the target compared to the unknown target. It indicates the importance of target information extraction in this multi-target task. However, there is still a significant error in the prediction of the arm at rest. Furthermore, most reconstructed trajectories pass through the target instead of stopping around the target. It might be because the neural activities have different patterns when the monkey is in rest or motion states. Meanwhile, the time the arm was in motion in each experiment was much longer than the time the arm was stopped (last 100 ms), leading to a data imbalance issue in regression training. Therefore the regression model is inaccurate in predicting when the arm is at rest.

When both classifiers are employed, there is a slight decrease in RMSE compared to that with only the target classifier. Furthermore, One-way repeated-measures ANOVA, where confidence $\alpha = 0.01$, reveals no influence on RMSE ($p < 0.001$) whether we use state classifier decoding. Furthermore, the trajectory reconstructed using two classifiers is similar to that using only the target classifier. Then the variance inflation factor (VIF) method shows that there are 86 VIFs in 98 larger than 2.5, which means that the multicollinearity problem is considerable.

The RMSE decreases significantly When the regression model changes from ordinary linear regression to RR. And the *post hoc* test identified that there is significant difference between RMSEs from Linear Regression ($mean = 11.51, std = 0.40$) and RR ($mean = 7.95, std = 0.25$)($p < 0.001$), which denotes that the multicollinearity issue is fixed through employing Ridge Regression.

Overall, the proposed RR-based decoder with two auxiliary classifiers achieves lowest RMSE (i.e., 7.95 mm) compared to the other decoders.

B. Future work

Due to physical constraints and physiology, there is a time delay between nerve activity and arm movement in biological life [14]. Thus, some previous works introduced an optimal time lag in decoding neural activity, achieving better results [15, 16]. The typical time lag ranges from 10 to 100 ms [16]. While due to the long time window (300 ms) used in the proposed decoder, the experimental results show that performance does not benefit from the time lag. Thus no time lag is applied in our method.

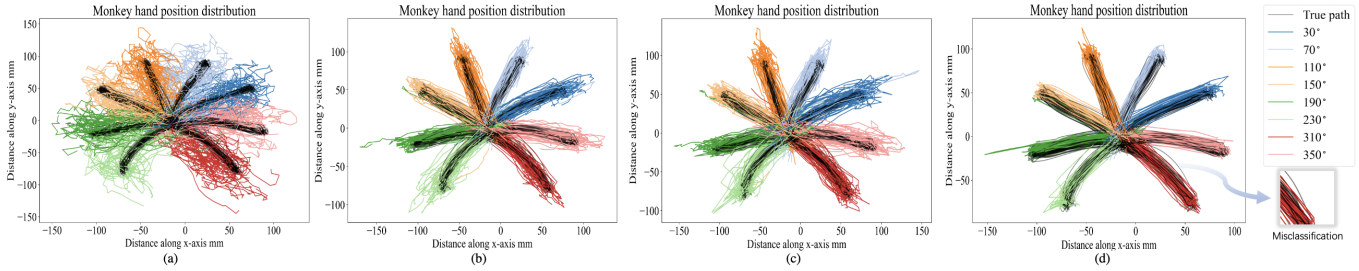

Fig. 3. Monkey hand's position results applying on (a) ordinary linear regression only, (b) linear regression with reaching target classifier, (c) linear regression with target classifier and movement state classifier and (d) RR with target classifier and movement state classifier.

Even though kNN can introduce a robust performance for reaching target and movement states classifiers, it is time confusing to choose a proper value of k, and the estimation phase is high architecture complexity. In contrast, the Bayesian classifier can utilise prior information directly to predict future work, which is a good idea to be implemented in the future so that the pipeline can automatically and directly work for neural scientists. In addition, [15] mentioned that the Kalman filter could indicate the uncertainty of hand movement and improve hand position precision, which is worth trying to check whether our hand performance can be improved or not.

V. Conclusion

Here, we report a multi-RR decoder with two auxiliary classifiers to achieve real-time 2D trajectory reconstruction of monkey arms. Such proposed estimation architecture introduces around 60% improvement in RMSE compared to the linear regression only since it can extract the target and movement state information.

Contribution

All code works are evenly done by Peter Guan, Kexin Huang and Zhongjie Zhang. As for the report, Zhongjie Zhang, Kexin Huang and Peter Guan mainly focus on Sections I, II and III. Moreover, the lasting parts are shared by the aforementioned three group members. As for Haonan Zhou, he simply made the Table II and ANOVA test in Section IV. Finally, the approximate percentage of contribution is listed in table below.

TABLE III
Group Member Contribution

Name	Pter Guan	Kexin Huang	Zhongjie Zhang	Haonan Zhou
Count	$\geq 30\%$	$\geq 30\%$	$\geq 30\%$	$\leq 10\%$

References

[1] M. D. Serruya, N. G. Hatsopoulos, L. Paninski, M. R. Fellows, and J. P. Donoghue, "Instant neural control of a movement signal," *Nature*, vol. 416, no. 6877, pp. 141–142, 2002.

[2] J. Park and S.-P. Kim, "Estimation of speed and direction of arm movements from m1 activity using a nonlinear neural decoder," in *2019 7th International Winter Conference on Brain-Computer Interface (BCI)*. IEEE, 2019, pp. 1–4.

[3] L. Paninski, M. Fellows, N. Hatsopoulos, and J. Donoghue, "Temporal tuning properties for hand position and velocity in motor cortical neurons. submitted," *J. of Neurophysiology*, 2001.

[4] Z. Li, J. E. O'Doherty, T. L. Hanson, M. A. Lebedev, C. S. Henriquez, and M. A. Nicolelis, "Unscented kalman filter for brain-machine interfaces," *PloS one*, vol. 4, no. 7, p. e6243, 2009.

[5] K. Xu, Y. Wang, S. Zhang, T. Zhao, Y. Wang, W. Chen, and X. Zheng, "Comparisons between linear and nonlinear methods for decoding motor cortical activities of monkey," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2011, pp. 4207–4210.

[6] J.-A. Ting, A. D'Souza, K. Yamamoto, T. Yoshioka, D. Hoffman, S. Kakei, L. Sergio, J. Kalaska, M. Kawato, P. Strick *et al.*, "Variational bayesian least squares: an application to brain–machine interface data," *Neural Networks*, vol. 21, no. 8, pp. 1112–1131, 2008.

[7] R. Foodeh, S. Ebadollahi, and M. R. Daliri, "Regularized partial least square regression for continuous decoding in brain-computer interfaces," *Neuroinformatics*, vol. 18, no. 3, pp. 465–477, 2020.

[8] M. A. van Gerven, Z. C. Chao, and T. Heskes, "On the decoding of intracranial data using sparse orthonormalized partial least squares," *Journal of neural engineering*, vol. 9, no. 2, p. 026017, 2012.

[9] K. V. Shenoy, D. Meeker, S. Cao, S. A. Kureshi, B. Pesaran, C. A. Buneo, A. P. Batista, P. P. Mitra, J. W. Burdick, and R. A. Andersen, "Neural prosthetic control signals from plan activity," *Neuroreport*, vol. 14, no. 4, pp. 591–596, 2003.

[10] J. M. Antelis, L. Montesano, A. Ramos-Murguialday, N. Birbaumer, and J. Minguez, "Decoding Upper Limb Movement Attempt from EEG Measurements of the Contralesional Motor Cortex in Chronic Stroke Patients," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 1, pp. 99–111, jan 2017.

[11] E. H. J. L. Fix, "Discriminatory Analysis Nonparametric Discrimination: Consistency Properties," USAF School of Aviation Medicine, Randolph Field, Texas, Tech. Rep., 1951. [Online]. Available: https://apps.dtic.mil/dtic/tr/fulltext/u2/a800276.pdf

[12] P. Cichosz, "Assessing the quality of classification models: Performance measures and evaluation procedures," *Central European Journal of Engineering*, vol. 1, no. 2, pp. 132–158, 2011.

[13] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, jun 2006.

[14] D. W. Moran and A. B. Schwartz, "Motor cortical representation of speed and direction during reaching," *Journal of neurophysiology*, vol. 82, no. 5, pp. 2676–2692, 1999.

[15] W. Wu, M. Black, Y. Gao, M. Serruya, A. Shaikhouni, J. Donoghue, and E. Bienenstock, "Neural decoding of cursor motion using a kalman filter," *Advances in neural information processing systems*, vol. 15, 2002.

[16] 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," in *SAB'02-workshop on motor control in humans and robots: On the interplay of real brains and artificial devices*. Citeseer, 2002, pp. 66–73.