Boosting Performance in Brain-Machine Interface by Classifier-level Fusion based on Accumulative Training Models from Multi-day Data

Huijuan Yang[1], Camilo Libedinsky[2,3,4], Cuntai Guan[1,5], Kai Keng Ang[1] and Rosa Q. So[1]

***Abstract*— The nonstationarity of neural signal is still an unresolved issue despite the rapid progress made in brain-machine interface (BMI). This paper investigates how to utilize the rich information and dynamics in multi-day data to address the variability in day-to-day signal quality and neural tuning properties. For this purpose, we propose a classifier-level fusion technique to build a robust decoding model by jointly considering the classifier outputs from multiple base-training models using multi-day data collected prior to test day. The data set used in this study consisted of recordings of 8 days from a non-human primate (NHP) during control of a mobile robot using a joystick. Offline analysis demonstrates the superior performance of the proposed method which results in 4.4% and 13.10% improvements in decoding (significant by one-way ANOVA and post hoc t-test) compared with the two baseline methods: 1) concatenating data from multiple days based on common effective channels, and 2) averaging accuracies across all base-training models. These results further validate the effectiveness of proposed method without recalibration of the model.**

I. INTRODUCTION

The use of neural cortical neuronal activity signal for controlling cursor, point and click typing, robotic arms and other assistive devices have widely explored [1], [2], [3], [4], [5], [6], [7]. Despite the rapid advancement of brain-machine interface (BMI), nonstationarity is still a problem since the model trained in earlier days may not perform well for test data collected later [3], [7], confining its wide use [3]. The nonstationarity may be due to the changes of physiological patterns (e.g., tuning patterns of neurons) and recording nonstationaries (e.g., electrodes impedance over time) in neural signals. As a result, re-calibration of model by incorporation of newly collected labeled data (e.g., close-loop neural control data) to leverage the old data is always required [1], [2], [5], [6], [7], [8]. A re-calibrated feedback intention-trained Kalman filter method was proposed, where the initial and re-calibrated models were fit using arm trajectories or cursor kinematics, and the estimated intended kinematics with neural signals. The underlying assumption was that the intended kinematics could be best described by the neurally controlled cursor and knowledge of the task [1]. Further quantitative investigation demonstrated that the intention estimation modifications on the decoder had direct improvements in terms of enhancing modulation and reducing the per-channel variance [3]. The long term decoding stability of local field potentials (LFPs) was better than that of the spike signals when the quality of spikes was poor or even no spikes [4], owing to the high redundancy among LFP features in different frequency bands. To enable long periods of practical use of BMI, mitigating the nonstationarity by tracking the statistics of neural activity during pauses, velocity bias correction and periodically recalibrating the decoder was proposed [7]. The wrongly selected words during BMI control can be prevented or undone to increase the typing rate by decoding the recent errors during close-loop BMI from intra-cortical spiking neural activity [6]. The dorsal premotor cortex was identified to be related to the differentiation of success from error trials.

In this paper, we investigate how multi-day data can be best employed for robust decoding without retraining of the decoding model. The hypothesis is that the data from multiple sessions and days possess rich information and the dynamic features can be utilized to reduce the variability between test and training data. For this purpose, we propose to employ data fusion technique (e.g., marjority voting) [8], [9], [10] which is operated on top of the initial base-training models from multiple sessions and days, namely, "boosting performance by classifier-level fusion (BP-ClsFu)".

II. MATERIALS AND METHODS

All procedures and experiments were performed in compliance with the Institutional Animal Care and Use Committee (IACUC). Detailed description of the behavior task, and neural and kinematics recordings can be found in [2], [5]. The data were recorded from a non-human primates (NHP), where the NHP was trained to drive a mobile robot to move forward, turn left or right, or stay still using a joystick [2], [5]. For each trial, a single movement (e.g., left, right, forward or stop) was cued by a liquid reward from the trainer and the reward was given if the NHP can successfully perform the tasks within 15 seconds. The neural spiking signal was acquired with 3 floating microwire arrays (e.g., 96 channels in total) at the left primary motor cortex. The neural signal was sampled at 12,987 Hz and band-pass filtered by an infinite impulse response (IIR) Chebyshev Type II filter, with the low and high cut off frequency of 300 Hz and 5000 Hz, respectively. Synchronized with the neural recordings, two channels of analog joystick signals were also recorded, and four classes (ground truth) representing right/left/forward movements and stop actions were defined with appropriate

*This work is supported by BMRC-EDB IAF and JCO DP grant

[1]H. Yang, C. Guan, K. K. Ang and Rosa Q. So are with Institute for Infocomm Research (I2R), A*STAR Singapore 138632 (email: {hjyang, ctguan, kkang, rosa-so}@i2r.a-star.edu.sg)

[2,3,4]C. Libedinsky is with [2]Department of Psychology, NUS (email: camilo@nus.edu.sg); [3]Singapore Institute for Neurotechnology, NUS; [4]Institute of Molecular and Cell Biology, A*STAR, Singapore.

[5]C. Guan is with School of Computer Science and Engineering, NTU, Singapore (email: ctguan@ntu.edu.sg)

978-1-5090-2809-2/17/$31.00 ©2017 IEEE

voltage thresholds [2], [5]. Data were collected on 8 non-consecutive days spanning 3 months. Several experimental sessions (e.g., 4 or 5) were conducted in a day. Each session consisted of about 20 trials for each direction. Only offline analysis was performed for this study.

III. Our Proposed Method

A. Threshold calculation and spike detection

Spike sorting was not performed since it has not been shown to improve the performance [1], [11]. The threshold was first computed from a segment of the training data, e.g., 0.5 second from the beginning of data by skipping a small interval of the unstable period of filtered data. Thereafter, the number of threshold crossings were calculated for each electrode. This was based on the assumption that the number of distinguishable neurons on an electrode would decrease over time, yet the multi-unit activity is more informative and stable [1]. The threshold (T^r) was calculated by [12]

$$T^r = C_0 * \delta_n \tag{1}$$

where the standard deviation of the background noise (δ_n) was estimated by

$$\delta_n = median(|f_x|/C_1) \tag{2}$$

where $|f_x|$ was the absolute value of band-pass filtered signal f_x of selected length, e.g., 0.5 second of samples, where C_0 and C_1 were the constants and were chosen as 5 and 0.6745, respectively. The choice of "median" instead of "mean" function in calculating the threshold is to take robustness of the threshold into consideration [12].

The detected thresholds (e.g., of dimension $n_c \times 1$, where n_c denoted the number of channels) were used for the detection of spikes both for training and test data. The threshold crossings were found and a small window was employed to center the crossings to further determine whether or not the detected crossings were real spikes.

B. Classifier-level fusion over multiple base-training models

The features used for decoding were the spike counts of effective channels, i.e., the channels with firing rate greater than the threshold firing rate (F_{tr}) (e.g., 2 Hz). The continuous decoder was operated at 10 Hz (i.e., every 100 ms) on the time bin with no delay. For each test day m, session n, the decoded directions for all time-shifting windows were obtained by decoding the direction of movements using the base-training model obtained from the day i, session j of the training data, i.e., denoted as $D_{i,j}^{m,n}(k)$, where k=1,2,...N_t denoted the kth decoding window. The training models contained the following parameters. Firstly, it contained indexes for the effective channels with firing rate not lower than F_{tr} at a training day/session. These indexes were employed as the indexes of effective channels for the test data as well. Secondly, it contained the thresholds obtained from the training data at a day/session ($T_{i,j}^r$), which were employed to calculate the firing rate for both training and test data.

In this paper, we investigated how to employ the data from multiple days/sessions prior to the test day/session to boost the decoding performance. Multiple base-training models were obtained based on the training data from each day/session. Each model was employed to decode the test data from a day/session, and the decoded directions from multiple base-training models were collectively employed. Classifier-level fusion such as majority voting was then performed to obtain the final decoded directions [8], [9], [10]. In this way, the wrongly decoded direction from each individual base-training model can be filtered out. This method was based on the assumption that the majority of the base-training models would perform reasonably well, hence, an improved performance and robust decoding can be expected by voting based on these base-training models. The overall idea of the proposed method, namely, "boosting performance by classifier-level fusion (BP-ClsFu)" is illustrated in Fig. 1.

The actual decoded direction for the kth trial was obtained by feeding the decoded directions obtained from multiple base-training models to a classifier-level fusion processor, e.g., majority voting function ($\mathcal{M}_{jv}$), which was given by

$$\tilde{D}_{m,n}(k) = \mathcal{M}_{jv}(D_{m,n}^{i,j}(k)) \tag{3}$$

where i, j denoted the index of day and session of training data; i=1,2,...,N_{rd}, j=1,2,...,N_{rs}, N_{rd} and N_{rs} denoted the number of days and number of sessions for training data, respectively. m, n denoted the test day and session, m=1,2,...,N_{ed}, n=1,2,...,N_{es}, N_{ed} and N_{es} denoted the number of days and number of sessions for test data, respectively. The pseudo codes for majority voting ($\mathcal{M}_{jv}()$) based on the accumulative base-training models of multi-day data for the kth decoding window is detailed in Algorithm 1.

Data: $D_{m,n}^{i,j}(k)$,i=1,2,...,N_{rd}, j=1,2,...,N_{rs}
Result: $\tilde{D}_{m,n}$
$D_{m,n}$=Φ (empty set);
for $i \in \{1, 2, ..., N_{rd}\}$ **do**
  for $j \in \{1, 2, ..., N_{rs}\}$ **do**
    if $D_{m,n}^{i,j}(k) \neq \emptyset$ **then**
      $D_{m,n}$=$[D_{m,n}; D_{m,n}^{i,j}(k)]$;
    end
  end
end
T_c=$unique(D_{m,n})$;
for $v \in \{1, 2, ..., numel(T_c)\}$ **do**
  $N_e(v) = numel(D_{m,n}|(D_{m,n} = T_c(v))$;
end
$id = v|N_e(v) = max(N_e)$;
$\tilde{D}_{m,n}$=$T_c(id)$;
return $\tilde{D}_{m,n}$;

Algorithm 1: The majority voting ($\mathcal{M}_{jv}()$) based on accumulative base-training models of multi-day data

In algorithm 1, $numel()$ and $unique()$ give the number of elements and the unique values with no repetition, respectively. $D_{m,n}^{i,j}(k) \in \{0, 90, 180, 270\}$, which represent the task of moving right, moving forward and moving left, and holding the joystick to stop the movements, respectively. These

decoded directions from the multiple base-training models are collectively voted and the dominant direction (i.e., with maximum number) will be taken eventually as illustrated in Algorithm 1. Linear discriminant analysis (LDA) was used as the classifier to obtain the directions in decoding as an output from a base-training model.

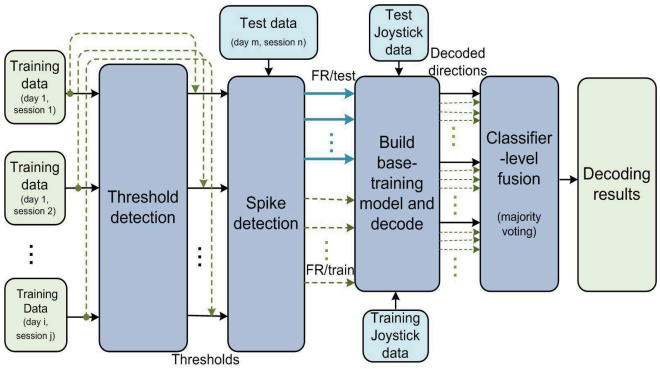

Fig. 1. Block diagram illustrating our proposed "boosting performance by classifier-level fusion (BP-ClsFu)"

IV. Experimental Evaluation and Discussion

The performance of our proposed method was evaluated based on the neural spiking data collected from 8 non-consecutive days (spanning over 3 months), with different number of sessions (i.e., 4, 4, 5, 4, 4, 5, 5, 4) in each day. In the evaluation, the data of all the sessions of current processed day (i.e., the test data) were evaluated based on all the base-training models obtained from the data recorded prior to that. For example, the test data from day 4 (D4) would be decoded using all the base-training models obtained from data recorded in day 1 (D1), day 1 and 2 (D1 and D2), and day 1, 2 and 3 (D1, D2 and D3), i.e., leading to three evaluations in total for D4. This is to investigate its effects on decoding performance by incrementally including data from more days in model training. It should be noted that each base-training model was obtained from training data of 1 session of 1 day in current implementation. Three types of decoding accuracies were obtained. Firstly, the accuracies for all the sessions of a test day were obtained by the majority voting based on accumulative base-training models of multi-day data, which was denoted as "Acc-Vot". Second, the decoding performance of all sessions of the test day was evaluated by concatenating all the data from the incrementally selected days/sessions prior to this date (denoted as "Acc-Conc"). In doing so, those common effective channels with firing rate higher than the threshold firing rate were found across data of all the incrementally selected training days and sessions. The accuracies of these two evaluations were also compared with accuracies of all sessions of current test day obtained by averaging the decoding accuracies using each individual base-training model (i.e., obtained based on the training data recorded prior to the test day), denoted as "Acc-TrSs". The overall comparison results are shown in Fig. 2, where the median accuracy among the accuracies of all the test sessions was reported. The relative increase of the accuracy of our proposed method ("Acc-Vot") with reference to the other two baseline methods, i.e, "Acc-Conc" and "Acc-TrSs" is reported in Table I. It can be observed from

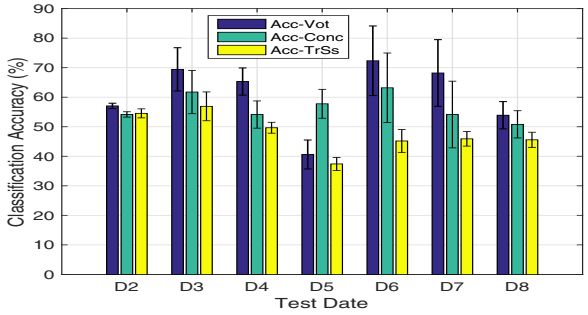

Fig. 2. Comparison of the accuracy obtained for each test day using our proposed method, i.e.,"Acc-Vot", and that of two baseline methods, i.e., "Acc-Conc" and "Acc-TrSs". Note that the median among the accuracies of all test sessions is shown.

TABLE I
Comparison of performance (%) between our proposed method (Acc-Vot), and the two baseline algorithms ("Acc-Conc" and "Acc-TrSs").

	Acc. of Test Day (%)							
	D2	D3	D4	D5	D6	D7	D8	Average
Acc-Vot	57.1	69.4	65.3	40.6	72.4	68.2	53.9	61.0
IncWt. Acc-Conc	2.9	7.7	11.2	-17.2	9.2	14.1	3.1	4.4
IncWt. Acc-TrSs	2.5	12.5	15.6	3.2	27.2	22.3	8.3	13.1

IncWt.: increase with reference to.

Fig. 2 and Table I, our method outperformed the baseline methods for most days. The decoding accuracy was worse compared to "Acc-Conc" only for the test day 5 (D5), where a very low accuracy was obtained for each individual base-training model. In this case, the assumption of using the voting method has been violated since the majority of the base-training models performed below 50%, hence voting did not greatly improve the performance. To summarize, the median accuracy of "Acc-Vot" was 4.4% higher than that of "Acc-Conc" method. Furthermore, the accuracy of our proposed method was 13.1% higher than "Acc-TrSs". The comparison with "Acc-TrSs" was included to show how much improvements can be made by including more data in the training. It should be noted that the size of training data for "Acc-Vot" and "Acc-Conc" was always the same, which varied with the number of days/sessions being incrementally included. In contrast, the averaging of the accuracies from multiple base-training models was employed in "Acc-TrSs", where each base-training model was obtained with 1 session from 1 day data as discussed earlier.

Comparison of the dynamic changes of the accuracies for all the test sessions across multiple test days are shown in Fig. 3. It can be observed that the performance of different sessions across days varies greatly, especially for day 5 ("D5"), where a low performance was obtained for our proposed voting-based method ("Acc-Vot") and that of the averaging across base-training models ("Acc-TrSs"). Apart

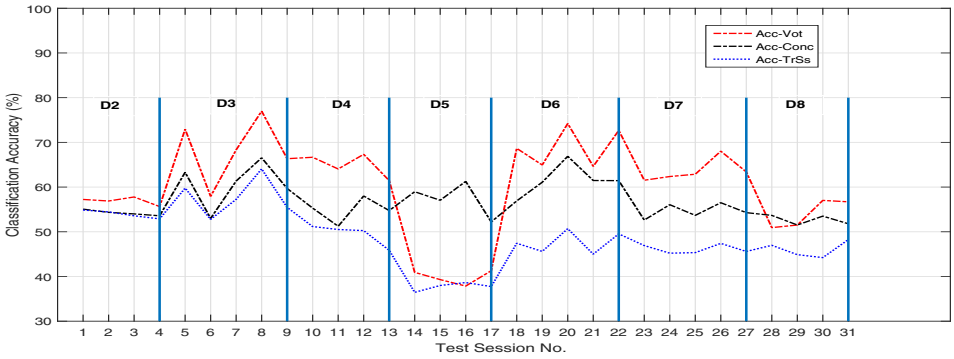

Fig. 3. Comparison of the dynamic changes of the accuracies for all test sessions across 7 days (D2-D8) using proposed method ("Acc-Vot") and the two baseline methods: by concatenating common effective channels ("Acc-Conc") and averaging across all accuracies obtained using each individual base-training model ("Acc-TrSs").

from this day, our proposed method consistently performed better than that of the concatenating method ("Acc-Conc") and "Acc-TrSs". This result demonstrates that the performance can be boosted by properly utilizing all existing data to build a robust model. Of course, this proposed method fails when the majority of the individual models perform poorly (e.g., D5). A possible solution is to increase the size of training data for each base-training model. For example, data from multiple sessions of one day can be employed to obtain the base-training model instead of only using data of one session in the current approach. In this way, the performance will be further improved from the proposed classifier-level fusion with the improved performance from each individual base-training model. A one-way ANOVA test between the classification accuracies of different methods was conducted, the results showed that there was a significant effect at $p<0.05$ significance level for the three methods ($F(2,90)=20.49$, $p=4.66e-8$). Post hoc comparisons using t-test between every two methods revealed that was significant difference between mean accuracy of "Acc-Vot" and "Acc-Conc" (p-value=0.01), and between "Acc-Vot" and "Acc-TrSs" (p-value=5.3e-10).

V. Conclusions

This study investigated day-to-day decoding in a BMI application and proposed a classifier-level fusion technique to fuse the classifier outputs from multiple base-training models obtained from multi-day data. Experimental evaluation based on data recorded in 8 days demonstrated that the proposed method can improve the decoding performance in comparison with the methods of concatenating data from multiple days, and averaging the accuracies across all individual base-training models. The improvements are significant for one-way ANOVA and post hoc t-test. Future work will investigate the effective size of training data to be selected for base-training models such that the boosting performance can be guaranteed even for those days with low base performance.

References

[1] V. Gilja, P. Nuyujukian, C. A. Chestek, J. P. Cunningham, B. M. Yu, J. M. Fan, M. M. Churchland, M. T. Kaufman, J. C. Kao, S. I. Ryu, and K. V. Shenoy, "A high-performance neural prosthesis enabled by control algorithm design," *Nat Neurosci.*, vol. 15, no. 12, pp. 1752–7, 2012.

[2] Z. Xu, R. So, K. K. Toe, K. K. Ang, and C. Guan, "On the asynchronously continuous control of mobile robot movement by motor cortical spiking activity," in *Conf Proc IEEE Eng Med Biol Soc. (EMBS 2014)*. IEEE, 2014, vol. 2014, pp. 3049–52.

[3] Fan JM, Nuyujukian P, Kao JC, Chestek CA, Ryu SI, and Shenoy KV., "Intention estimation in brain-machine interfaces," *J Neural Eng.*, vol. 11, no. 1, pp. 016004, 2014.

[4] Wang D, Zhang Q, Li Y, Wang Y, Zhu J, Zhang S, and Zheng X., "Long-term decoding stability of local field potentials from silicon arrays in primate motor cortex during a 2d center out task," *J Neural Eng.*, vol. 11, no. 3, pp. 036009, 2014.

[5] C. Libedinsky, R. Q. So, Z. Xu, K. K. Toe, D. Ho, C. Lim, L. Chan, Y. Chua, L. Yao, J. H. Cheong, J. H. Lee, K. V. Vishal, Y. Guo, Z. N. Chen, L. K. Lim, P. Li, L. Liu, X. Zou, K. K. Ang, Y. Gao, W. H. Ng, B. S. Han, K. Chng, C. Guan, M. Je, and S.-C. Yen, "Independent mobility achieved through a wireless brain-machine interface," *PLoS ONE*, vol. 11, no. 11, pp. 1–13, 2016.

[6] N. Even-Chen, S. D. Stavisky, J. C. Kao, S. I. Ryu, and K. V. Shenoy, "Auto-deleting brain machine interface: Error detection using spiking neural activity in the motor cortex," in *Conf Proc IEEE Eng Med Biol Soc. 2015 (EMBS 2015)*. IEEE, 2015, vol. 2015, pp. 71–5.

[7] Jarosiewicz B, Sarma AA, Bacher D, Masse NY, Simeral JD, Sorice B, Oakley EM, Blabe C, Pandarinath C, Gilja V, Cash SS, Eskandar EN, Friehs G, Henderson JM, Shenoy KV, Donoghue JP, and Hochberg LR, "Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface," *Sci Transl Med.*, vol. 7, no. 313, pp. 313ra179, 2015.

[8] Wenyuan Dai, Qiang Yang, Gui-Rong Xue, and Yong Yu, "Boosting for transfer learning," in *Proceedings of the 24th International Conference on Machine Learning*, New York, NY, USA, 2007, ICML '07, pp. 193–200, ACM.

[9] H. Yang, C. Guan, K. K. Ang, K. S. Phua, and C. Wang, "Selection of effective eeg channels in brain computer interfaces based on inconsistencies of classifiers," in *Conf Proc IEEE Eng Med Biol Soc. 2014 (EMBS 2014)*. IEEE, 2014, vol. 2014, pp. 672–5.

[10] D. Lahat, T. Adali, and C. Jutten, "Multimodal data fusion: An overview of methods, challenges, and prospects," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1449–77, 2015.

[11] Christie BP, Tat DM, Irwin ZT, Gilja V, Nuyujukian P, Foster JD, Ryu SI, Shenoy KV, Thompson DE, and Chestek CA, "Comparison of spike sorting and thresholding of voltage waveforms for intracortical brain-machine interface performance," *J Neural Eng.*, vol. 12, no. 1, pp. 016009, 2015.

[12] RQ Quiroga, Z Nadasdy, and Y Ben-Shaul, "Unsupervised spike detection and sorting with wavelets and superparamagnetic clustering," *Neural Comput.*, vol. 16, no. 8, pp. 1661–7, 2004.