2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

Precisely Detecting and Resolving BCI Errors With Case Functions

Chloe N Winston[1], Cailin Winston[1], Caleb Winston[1], Claris Winston[1], Cleah Winston, Rajesh P N Rao[1]

***Abstract*— Brain-computer-interfaces (BCIs), such as neuro-prostheses and restorative speech devices, rely on neural decoding models to accurately extract control and diagnostic information from neural signals. Due to the complexity and variability of neural signals, training models in BCIs requires large amounts of labeled data, and even well-trained models can make erroneous predictions. We demonstrate the ability of case functions and case-stratified-uncertainty sampling to precisely detect errors in BCIs and ultimately improve performance through data correction and targeted data collection.**

***Clinical Relevance*— We establish a novel method to precisely identify brain-computer-interface (BCI) errors in real time in the absence of ground truth data, and thereby suggest (1) a sampling of erroneous BCI predictions for manual correction and (2) tasks for data collection to improve BCI performance.**

I. Introduction

Brain-computer-interfaces (BCIs) are a class of biomedical devices that interface with the brain and can be used to decode and execute intended actions or to monitor brain function [1]. Their performance relies on training an underlying neural decoding model on large amounts of high-quality labeled data. BCIs are highly safety- critical as an error such as an unexpected movement of a prosthetic can have serious consequences. Leveraging ideas from software engineering, we present case functions and case stratified uncertainty sampling to detect and resolve such errors in BCIs.

II. Methods

Case functions are expert-written functions (written in a high-level programming language such as Python) that detect slices, tasks, or errors given a model execution. Slices are subsets of input data (e.g., EEG recordings that exhibit an artifact or are from patients with age >50). Tasks capture BCI-relevant user behavior (e.g., lifting an arm or waking up). Errors are instances of undesirable model predictions, such as rapid or invalid transitions of the output state. Case functions can maintain time-varying state that is used to capture errors.

Due to resource intensiveness of manual error correction, we require an algorithm to sample the error occurrences that are the most significant. Uniform sampling (U) and entropy-based uncertainty sampling (E), traditionally used in active learning, can both be biased [2]. We introduce case stratified-uncertainty (CSU) sampling, which takes a stratified sample, where the error case functions are strata (to address bias) and chooses the most uncertain from each stratum (to address potential low quality of case functions). *ECF* is the set of case functions *CF(x)* that detect errors across model executions x. *H(x)* computes a measure of uncertainty of x. We define CSU sampling using:

$$CSU(x) = \begin{cases} \dfrac{P(CF(x) \mid x)H(x)}{\left(\sum\limits_{x' \in X \wedge CF(x)} H(x')\right)} & \exists\, CF \in ECF : CF(x) \\ 0 & \text{otherwise} \end{cases}$$

Model executions selected with CSU sampling can either be manually corrected (data labeling) or used to compute the distribution of co-occurring tasks/slices to collect labeled data (clinical data acquisition). Corrected or collected data can be used as training data to improve the model.

III. Results

We evaluate our methods on BCI applications (listed in Table I). The motor decoding BCI decodes movement of an individual's fists [3]. The cursor control BCI decodes the position of a controlled cursor [4]. We write case functions to detect errors such as invalid/rapid state transitions or inconsistencies across recording modalities, and to detect tasks such as moving a particular fist in a certain direction.

TABLE I.

Application	**Precision**			**Performance Improvement (%)**		
	U	***E***	***CSU***	***U***	***E***	***CSU***
Motor Decoding	54%	54%	66%	5.09/1.21	6.07/8.73	12.14/18.45
Observation/Execution	27%	14%	45%	-4.65/2.3	-8.14/9.3	12.8/11.6
Cursor Control	5.39	5.40	5.96	1.74/0.46	3.23/-0.49	-2.1/0.66
Field Navigation	1024	1198	1211	0.94/2.16	2.68/3.36	1.97/2.35
Sleep Stage Detection	31%	30%	47%	12.6/1.3	10.5/5.6	10.2/4.1

Table I shows the precision of U, E, and CSU sampling in detecting errors and the percent performance improvement that results from further training on either hand-labeled sampled data (left of '/') or clinically acquired data given sampled distribution of tasks (right of '/'). We found that CSU sampling can more precisely detect errors and generate training data that allows to a 15% increase in model accuracy.

IV. Discussion & Conclusion

In conclusion, expert-written case functions and CSU sampling are novel methods for precisely detecting and resolving BCI errors. We have demonstrated that both labeling CSU-sampled data points and acquiring data based on a CSU-sampled task distribution result in a greater end-to-end improvement in BCI performance given the same labeling or acquisition effort as other sampling methods.

References

[1] Rao RPN. Brain-Computer Interfacing: An Introduction, New York, NY: Cambridge University Press, 2013.
[2] B. Settles, "Active learning literature study", 2009.
[3] G. Schalk, "BCI2000: A General-Purpose Brain-Computer Interface (BCI) System." IEEE Transactions on Biomedical Engineering, 2004.
[4] J. Glaser, et. al, "Machine learning for neural decoding" arXiv.

[1]Authors are with Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle USA, email: cailinw@cs.washington.edu