Methods for Estimating User State from Real-time fNIRS Data

Samuel Hincks
Tufts University
Medford, MA 02145, USA
samhincks@gmail.com

Daniel Afergan
Google Inc.
Mountain View, CA 94043
afergan@google.com

Robert Jacob
Tufts University
Medford, MA 02145, USA
jacob@cs.tufts.edu

Abstract
Implicit user interfaces depend on an accurate and real-time stream of predictions about the user's state. This paper compares advantages and disadvantages of two overarching approaches for solving the challenge of converting live physiological data into plausible estimates of the user's state in the context of fNIRS-based adaptive user interfaces.

Author Keywords
physiological computing; fNIRS; adaptive user interfaces; machine learning; BCI; passive brain-computer interface; implicit interface; workload

ACM Classification Keywords
H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous

Introduction
In the design and calibration of machine learning algorithms that detect the user's mental state, the human plays two roles. First and foremost, the human must design the classification algorithm, the software which converts unprocessed streaming input into confidence-weighted predictions about the user's state in real time. In addition, the human must provide the baseline measures on which this algorithm is based. In this paper, we will discuss this sec-

ond issue in the context of building algorithms that detect a user's state from functional near-infrared spectroscopy (fNIRS) signals [4, 6, 3], a lightweight and low-cost neuroimaging technique that measures oxygenation changes in surface areas of the brain. We will outline different overarching solutions to the problem of knowing how a real-time constellation of features from fNIRS data correspond to the user's state, and highlight the domain specific constraints: that each user possesses a unique brain and that their time is scarce and comfort important. The machine learning issues discussed, while considered in the context of fNIRS experiments, may generalize more broadly in the domain of physiological computing, where prior knowledge about how a signal correlates with a state must be combined with new information about each user's individual physiology.

Machine Learning and State Induction

A typical workflow for designing an fNIRS-based machine learning algorithm proceeds as follows [1, 7, 2, 5]. Wearing fNIRs, the participant first completes a succession of trials with intermittent rest periods, intended to induce a transient cognitive state of interest. In the background, machine learning software associates computed features from the fNIRS data with a label describing the cognitive state supposedly elicited by this trial. This functions as the instance to a supervised machine learning algorithm. When enough trials have been collected, the machine learning algorithm is evaluated using leave-one-out cross-validation, and, presumably, prepared to classify streaming fNIRS values.

In Afergan (2014), we applied this method to calibrate a machine learning algorithm that classified a participant's cognitive workload as he or she set the flight path of Unmanned Aerial Vehicles (UAVs) in an air traffic control simulation [1]. Participants first completed fifteen 25-second visuospatial 3-backs and fifteen 25-second visuospatial 1-backs, interspersed by a 15 second resting period. These trials were filtered and reduced into 32 features describing the window's mean and slope values which, along with the associated class values, functioned as the training set for a support vector machine (SVM). In the experimental phase, the realtime classification of the SVM controlled the difficulty of the simulation, removing UAVs from user control when they were deemed to be in high cognitive workload and adding UAVs when they were deemed to be in low cognitive workload. This adaptation decreased operator failure rate (as measured by the number of collisions with no-fly zones) by 35% compared to a non-adaptive condition in which UAVs were added or removed intermittently according to predefined script.

This approach has several advantages. It leverages the optimized function fitting methods of machine learning, allowing it to find patterns which might elude a human observer utilizing classical statistical techniques. It has a built-in system for confirming that a state can be reliably induced and measured in a subject by leave-one-out-analysis. It generalizes nicely for calibrating new algorithms: one need only replace the calibration task and corresponding labels. It assumes, rightly, that subjects have different brains and that probe placements differ slightly from one experiment to another. Finally, it doesn't require the designer of the algorithm to possess advanced neuroscientific expertise as the burden of discovering neural correlates of state is left to automated pattern discovery.

But the approach also suffers notable limitations. First, unless an old model is recycled, each experiment requires a tedious calibration period for every new subject and session. To minimize calibration effort, most experiments use a limited number of trials to protect the subject's time and comfort and avoid fatigue. Having so few instances might

constrain the potential of the machine learning algorithm and also limit the number of features it can use to avoid overfitting. While certain cognitive states can be easy to induce (e.g. cognitive workload), other states simply don't invite easy induction, and lack on-off switches (e.g. emotion). Furthermore, each subject has unique cognitive endowment, and so the stimuli necessary to induce an intense state of cognitive workload likely differs across subject. One subject's 1-back may be as cognitively demanding as another subject's 2-back. A subject could give up on a trial, thereby supplying a mislabeled instance to the classifier. Even if the state has been reliably induced, there is a risk it doesn't generalize to the more general cognitive state under investigation in the experiment. A subject may utilize specialized cognitive resources for solving an n-back - resources which they then don't use in the more general case of cognitive workload. Finally, and perhaps most fatally, the trial-based machine learning approach locks the machine learning algorithm to a specific temporal alignment requiring clean transitions from a baseline state into the state under investigation. It is thus not entirely unrealistic to expect the machine learning algorithm to function effectively only when the real-time data aligns with the trial: exactly k seconds after a transition from a baseline state into a workload state.

Given this extensive list of shortcomings, it is worthwhile considering the space of reasonable alternatives.

Recycling Data

The requirement of a tedious and necessarily brief calibration process could be solved by leveraging old data from other sessions and subjects. Instead of discarding a subject's data once it has supplied the training set for that experiment, high quality data ought to be preserved and added to a growing pool of generalizable data. In the best case, the needed parameters of a classification algorithm have already been customized prior to the subject's arrival. The calibration period can instead be used to discover a better location fitting for the probes, or it might be used to supplement the existing model with additional preferentially weighted instances from that subject. Since the capability of the cross-subject model can be gauged quickly, the experimenter can decide after the subject has arrivied whether or not a new model needs to be trained for him or her.

By analyzing the cross subject results of several cognitive workload datasets, it seems that this approach somewhat reduces the mean classification accuracy. In other words, when data from a subject is entirely omitted and only data trained on other subjects is considered, classification accuracies drop somewhat but not dramatically. We therefore recommend that this practice is used only as a back-up. The experimenter should be ready to train a specific model only if the general model fails to provide plausible measures of their real-time state.

A Hard Coded Model

A promising alternative to calibration-based machine learning requires hardcoding a model for detecting the user's mental state. Since it is known from both fNIRS and fMRI literature that high cognitive workload tends to require computing in the dorsolateral prefrontal cortex, a possible algorithm could simply output high cognitive workload when the sensors probing this region exhibit a transient increase in measured oxygenation. Like the cross-subject approach, this cuts out the training period. It also overcomes the potential risk of a benchmark task not eliciting a pristine and general high workload state for every trial. In addition, it gives a natural estimate of the intensity of the state since, presumably, a greater increase in oxygenation denotes a more intense state of cognitive workload. Finally, and most

fruitfully, this algorithm can circumvent the issue of temporal alignment by keeping the windows for which it computes slope short (e.g. 6 seconds or the time it takes for a state to physiologically manifest) and encouraging its subscriber (the adaptive system) to use its output as a transition detector, not as a continuous barometer of the user's absolute state.

These advantages come at the expense of forfeiting the potential added information discovered by routine machine learning. While there is an average case profile for what cognitive states look like, it is possible that individual subjects exhibit activation profiles unique to them.

If this is by far the simpler approach and potentially also superior, then why has it not been used in more fNIRS-based real-time systems? We think there are two main reasons for this. First, real-time algorithms are born from offline algorithms and offline algorithms are evaluated in a manner that does not truly represent their real-time capability. The leave-one-out-evaluation method inflates the capability of an algorithm whose intended use is to classify real-time data since it only ever tests trials that belong to the same niche of cognitive workload (the calibration task), have the same resting baselines that immediately precede it, and also have temporal alignment to the data on which it was trained. Of course a specialized machine learning based approach will triumph in this test. Second, the hard coded approach requires a more extensive suite of filtering techniques working in real-time. Presumably, the discerning capability of machine learning allows it to avoid being tricked by systemic / respiratory influences to the signal that would undoubtedly set off a naive slope detector. Recent innovations to fNIRS processing that utilizes a near-source detector pair and adaptive filtering is therefore essential to this approach [8].

Can these approaches be combined?

Since each of these approaches carry significant advantages and disadvantages, it would be productive to consider how the three approach can be assimilated into one superior method. It would borrow the insight that sophisticated and non obvious-patterns can be extracted adaptively during the session. It would leverage the fact that former models can be recycled and lend insight into the user under investigation. And it would embrace neuroscientific literature and perhaps also curb its ambition to detect the user's state in absolute instead relative terms.

In one reconciliation of these ideas, the design of the algorithm could precede as follows.

1. Recruit a diverse but small set of template subjects, preferably subjects with excellent self-awareness (e.g. meditators or dispositional self monitors)
2. Measure brain activity as these subjects complete ordinary computer tasks: writing, reading, programming, drifting away into task unrelated rumination.
3. Concurrently, extract features from this data known from the literature to correlate with the state over relatively short periods of time, and compute the relative improbability of these features.
4. When a feature exhibits a relatively improbable value, actively query the subject, and invite them to plot how their state might have changed recently along any dimensions of interest; label the data according to this value
5. Train a machine learning to distinguish these dimensions from the ordinary state. Let the output of the classifier represent state transitions.

This method has its own drawbacks. It may be the case that the template subjects have a systematically different wiring that prevents proper generalization. The apparent absence of a 'perfect solution' suggests a problem with intrinsic difficulty and need for creative ingenuity and research. The heart of the issue relates to the challenge of obtaining a mental state on demand. Once it is in fact there, an ordinary mind tends to lack the meta-reflection and motivation needed to alert the algorithm of an appropriate label. Looking forward, our hunch is that the solution will involve less reliance on the individualized artificial state induction paradigm and more on a hybrid multi-subject approach trained on organic states that emerge naturally.

References

[1] Daniel Afergan, Evan M Peck, Erin T Solovey, Andrew Jenkins, Samuel W Hincks, Eli T Brown, Remco Chang, and Robert JK Jacob. 2014a. Dynamic difficulty using brain metrics of workload. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 3797–3806.

[2] Daniel Afergan, Tomoki Shibata, Samuel W Hincks, Evan M Peck, Beste F Yuksel, Remco Chang, and Robert JK Jacob. 2014b. Brain-based target expansion. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 583–593.

[3] Audrey Girouard, Erin Treacy Solovey, Leanne M Hirshfield, Krysta Chauncey, Angelo Sassaroli, Sergio Fantini, and Robert JK Jacob. 2009. Distinguishing difficulty levels with non-invasive brain activity measurements. In *Human-Computer Interaction–INTERACT 2009*. Springer, 440–452.

[4] Christian Herff, Dominic Heger, Ole Fortmann, Johannes Hennrich, Felix Putze, and Tanja Schultz. 2013. Mental workload during n-back taskâĂŤquantified in the prefrontal cortex using fNIRS. *Frontiers in human neuroscience* 7 (2013).

[5] Erin Solovey, Paul Schermerhorn, Matthias Scheutz, Angelo Sassaroli, Sergio Fantini, and Robert Jacob. 2012. Brainput: enhancing interactive systems with streaming fnirs brain input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2193–2202.

[6] Erin Treacy Solovey, Audrey Girouard, Krysta Chauncey, Leanne M Hirshfield, Angelo Sassaroli, Feng Zheng, Sergio Fantini, and Robert JK Jacob. 2009. Using fNIRS brain sensing in realistic HCI settings: experiments and guidelines. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. ACM, 157–166.

[7] Erin Treacy Solovey, Daniel Afergan, Evan M Peck, Samuel W Hincks, and Robert JK Jacob. 2015. Designing implicit interfaces for physiological computing: Guidelines and lessons learned using fNIRS. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21, 6 (2015), 35.

[8] Quan Zhang, Gary E Strangman, and Giorgio Ganis. 2009. Adaptive filtering to reduce global interference in non-invasive NIRS measures of brain activation: how well and when does it work? *Neuroimage* 45, 3 (2009), 788–794.