Strategies for Practical Brain-Computer Interface in Real-World Settings

Nick Merrill[1], Thomas Maillart[1], Benjamin Johnson[2] and John Chuang[1]
[1]*School of Information, UC Berkeley, Berkeley, California, USA*
[2]*???*
ffff@berkeley.edu

Keywords: BCI, Mobile, Ubiquitous Computing

Abstract: Brain-computer interface (BCI) systems are notoriously hard to calibrate to their users, especially with the sorts of low-cost, ergonomic sensors suitable for realistic, out-of-lab settings. In this study, we explore strategies for quickly achieving BCI with a single EEG sensor and minimal computational requirements, looking specifically at the effect of a novel signal processing technique on these factors. We simulate the calibration of a binary BCI on 14 healthy subjects, and describe the tradeoffs between computational speed, time needed to calibrate the system, and the accuracy of the resulting BCI. We discuss possibilities for the design of consumer-ready BCIs, and for the use of predictive agents in sensor-driven applications generally.

1 INTRODUCTION

The intro starts a little sharp with a broad definition. Maybe, it would be good to give more context that relates to our results, and be more precise about what we mean by BCI.

Brain-computer interface (BCI) systems establish a direct communcative link between the brain and an electronic system (Dornhege, 2007; McFarland and Wolpaw, 2011). Recently, the combination of machine learning algorithms and non-invasive electroencephelographs (EEG) has yielded proof-of-concept systems ranging from brain-controlled keyboards and wheelchairs to prosthetic arms and hands (Blankertz et al., 2007b; Milln et al., 2010; D. Mattia, 2011; Hill et al., 2014; Campbell et al., 2010).

There are many reasons why these BCI systems have not found wide adoption outside of lab settings. First of all, they often require large, complex scanning caps, which are impractical for disabled users and generally undesirable for ergonomic reasons (Ekandem et al., 2012; Leeb et al., 2013). Meanwhile, the amount of data produced by these caps is large, which places high requirements on end-user's hardware. Finally, BCIs often require upward of an hour to calibrate to their users, with close supervision by researchers, and may require regular recalibration due to the nonstationary nature of EEG signals. (Vidaurre et al., 2006; Vidaurre et al., 2011a; Blankertz et al., 2007a) Recent work has come close to eliminating traditional calibration, though not using ergonomic headsets.

For BCI systems to enjoy wider adoption "in the wild," they must calibrate to individual users quickly and achieve decent information transfer rates (ITR), but with fewer sensors than their lab-based counterparts, and with noisier signals, as data acquisition will occur while people are performing daily tasks, moving, walking, talking, and so on. As an added challenge, their computational firepower may be limited by the mobile & wearable computing architectures on which they will most likely be deployed.

In this study, we use recordings from a single, dry electroencephalographic (EEG) sensor to simulate the calibration of a simple BCI, and investigate the effect of a novel signal extraction technique on the systems computational performance and accuracy. First, we find that our signal extraction technique significantly increases the computational speed of a classification-based BCI without a significant detriment to accuracy. Second, we find evidence that this technique can be used to build effective mental task classifiers with under a minute of training data.

2 RELATED WORK

2.1 Brain-computer interface "in the wild"

Wider adoption of BCI systems relies on two main streams of research: (i) the development of ergonomic sensors suitable for use in naturalitic settings and (ii) the ability to adapt lab-developed BCI strategies to the new constraints that these sensors impose on our data processing abilities.

Many inexpensive, comfortable EEG devices have come to market, most of which use "dry" electrodes that do not require special gels. Compared to their lab-based counerparts, these devices have many fewer electrodes, thus limited spatial resolution, and produce significantly noisier signals **(is there a benchmark available in the literature?)**. Regardless, past work has demonstrated several mobile-ready BCI systems that use these scanning devices, and the Neurosky MindSet in particular (the headset used in this study - a single, dry EEG electrode placed roughly at FP2, which connects wirelessly to phones and computers, and sells for roughly 100USD) has been used to succesfully detect emotional states, event-related potentials (ERP), and to employ brain-based biometric authentication (Crowley et al., 2010; Grierson and Kiefer, 2011; Chuang et al., 2013). However, the use of consumer EEGs for the direct, real-time control of software interfaces has proven more difficult (Carrino et al., 2012; Larsen and Hokl, 2011). We expect significant improvements from consumer-grade EEG devices in the near future, with more sensors and better signal quality (e.g. Interaxon Muse, Melon headband, Emotiv Insight); however, we expect the signal from these devices will remain noisier than lab-basde counterparts, as people will be wearing and using them while moving, and in uncontrolled environments with ambient electromagnetic signals interfering with endogenous biosignals. **Mobile systems require reducing the bandwidth (currently 1 megaoctet per dry EEG sensor sampling at 512 Hz) ¡– why/what does this mean?** .

To transition BCI from the lab into naturalistic environments, we must squeeze more signal out of fewer, and less reliable, sensors. Furthermore, since BCIs are envisioned largely as always-available input devices, they will likely be deployed on mobile processors and perhaps even embedded processing systems; our computational resources may be more similar to that of a smartphone than of a desktop workstation, and it is feasible that we may need to do some processing "in the cloud" (ie., on a more powerful server to which the client sends data over the network, similar to the way Apple's Siri processes voice data). For effective BCI to occur in these environments, we must extract signal in a maximally efficient way so as to limit our computational footprint, and perhaps even to minimize the size of data if we wish to ship it to an external server.

2.2 Statistical signal processing in EEG-based BCI

BCI systems generally aim to recognize a user's mental gestures as one of a finite set of discrete symbols, a problem that can be thought of as a pattern recognition task (Lotte et al., 2007). The difficulty of this task stems primarily from the variable and non-stationary nature of neural signals: the "symbols" we wish to identify are expressed differently between individuals, and even vary within individuals from trial to trial (Vidaurre et al., 2006; Vidaurre et al., 2011b).

In order to compensate for variability in BCI signals, recent work has leveraged adaptive classification algorithms to distinguish between mental gestures (Lotte et al., 2007; Vidaurre et al., 2011b) *steal some lines introing classificatino algos............maybe steal a line explaining what a classifier is in the context of a BCI system, and how we train one.......* In classification algorithms generally, larger feature vectors require that an exponential increase in the amount of data needed to describe classes, a property known as "the curse of dimensionality" (Jain et al., 2000; Raudys and Jain, 1991). Traditionally, BCI applications rely on dense, high-diemsnional feature vectors produced by multi-electrode scanning caps with high temporal resolution, so dimensionality represents a major bottleneck in training classification algorithms. This bottleneck threatens the responsiveness of BCI from a user experience standpoint and places high requirements on end users' hardware.

2.3 Online, co-adaptive calibration

Learning to control a BCI system involves more than an adaptive software algorithm. Shenoy et al (2006) frame BCI learning as a cooperation between two adaptive systems: the BCI's algorithms and the human user (Shenoy et al., 2006). By building interfaces in which the user and the BCI "co-adapt" during an interactive calibration step, past work has turned BCI novices into competent users over the course of hours instead of days or weeks, and without manual calibration by a researcher (Vidaurre et al., 2006; Vidaurre et al., 2011a; Vidaurre et al., 2011b). Past work on co-adaptive BCI has used a several-step approach in which the system feeds preprocessed data to an adaptive classifier, which uses new and past data to optimize and recalculate itself, either during intermittent, offline steps or continuously online (Vidaurre et al., 2006; Lu et al., 2009; Das et al., 2013). During calibration, users perform "labeled" (that is, known) mental gestures in order to produce samples for the classifier. Meanwhile, the classifier performs various experiments in which it attempts to establish which features of the data are most informative. Systems may generate multiple models in parallel and combine their decisions democratically (an "ensemble approach"). After several calibration steps, the system is able to estimate the user's control by assessing its model's accuracy on samples it has already recorded.

2.4 Co-adaptive BCI in naturalistic settings

For the control of interface systems, it is crucial that mental gestures be actuated intentionally, and that the system's interpretation be immediately verifiable by the user. (McFarland and Wolpaw, 2011) *Maybe a line here about how the system needs to be fast for responsiveness* Efficient calibration is particularly crucial for real-world use, as EEG signals vary between subjects, and could even change within the same subject over time. From a technical standpoint, calibration amounts to the training and re-training of one or several adaptive algorithms. Calibration can be processing-intensive on a mobile device, especially if the system is computing multiple candidate models. This requires a great deal of online signal processing, which entails not only the computational time required to train the classifier but also the space required to handle the data and the time required to read and write the data from memory or from disk.

3 METHOD

The data used in this experiment were taken from a previous study. As our research involved human subjects, our experimental procedures were approved by an Institutional Review Board. We recruited 15 subjects to participate in our study, all of whom were UC Berkeley undergraduate or graduate students. Each subject met with investigators in a quiet, closed room for two 40-50 minute sessions on two separate days. We briefed subjects on the objective of the study, fitted them with a Neurosky MindSet headset, and provided instructions for completing each task. As the subjects performed each task, we recorded readings from the headset (i.e., difference of potential and power spectrum every half second).

3.1 Tasks

3.2 Signal extraction

The Neurosky MindSet SDK delivered a power spectrum of its data every half second. Offline, we compress the data in the temporal dimension, taking the middle n seconds of the recording, where $n = \{0.5, 1.0, 1.5, 2.0, ..., 8.0, 8.5, 9.0\}$.

The Neurosky software computes a power spectrum every half second. The maximum frequency is 256Hz, as the maximum sampling rate of Neurosky hardware is 512Hz. The power spectrum is computed with discrete bins of 1/4 Hz. Each bin represents the intensity of activation of a frequency range (e.g., between 1 and 1.25 Hz) in a half-second time window. There are therefore 1024 values reported for one power spectrum. Our samples are more or less 10 seconds, which means around 20 power spectra computed per sample. Based on the signal quality also recorded some power spectra are removed (Benjamin knows the exact filtering method).

Thereafter, the signal extraction method consists of two main steps: (i) to build a statistical significance out of the n power spectra produced in each sample, and (ii) to "compress" the information into a smaller vector size using a median filter. For each bin of 1/4 Hz, we can compute a median value out of the n power spectra generated for one sample. We obtain a discrete probability density function (pdf) with each bin being the median of the corresponding bins in the n power spectra. At this stage, we have a discrete pdf of 1024 bins for the whole sample. This method represents a "stacking" of several probability density functions into one representing the statistical average of all the others.

Binning the pdf is a simple way to "compress" the information contained in the original power spectrum. The basic idea is to take the median of several bins. For instance, four contiguous bins (1-1.25,1.25-1.5,1.5-1.75,1.75-2) have the values (4,4,5,5) the value resulting from combining these values into one bin would be 4.5. The pdf is heavy-tailed and it is desirable to arrange the "compression" bins in a way it provides relevant information on the whole distribution. One way to do this efficiently is to arrange the compression bins in a logarithmic fashion.

Figure **??** shows, in double logarithmic scale, the original 1024 bins (blue dots) of the pdf obtained from averaging the n power spectra of one sample, and the resulting "compressed" pdf with 100 log-bins. As we can see, the log-binning preserves the structure of the pdf.

In summary, we build a probability density function of brain frequencies as captured by the Neurosky hardware, from the n power spectra of each sample. We then used a log-binning method to reduce the 1024 bins from the original power spectrum to an arbitrary smaller number of log-bins (e.g., 100 log-bins on the Figure). This method makes a sort of statistical averaging by stacking, and then "compresses" the result in way that it is easy to use in a classifier.

At this point, we have a one-dimensional signal flattened in the time dimension, computing the mean magnitude of each bin over all readings in the recording - a row vector with one entry for each each measured frequncy bin.

3.3 Classifier

Support vector machines (SVM) are a set of supervised machine learning methods that take labeled example data to create a model that can be used to predict the classes of unlabeled data. SVMs use a hyperplane (an n-dimensional construct in n+1 dimensional space) to draw discriminatory boundaries between classes. In contrast to linear discriminant analysis, which has a long history of use in BCIs, SVMs select the hyperplane that maximizes distance from the nearest training points, which has been shown to increase the model's generalizability (Burges, 1998). For more on SVM's in BCI: (Garrett et al., 2003; Grierson and Kiefer, 2011)

a diagram could help understand how SVM works

In this study, we use LinearSVC, (Fan et al., 2008) a wrapper for LibLinear exposed in Python through the ScikitLearn library. (Pedregosa et al., 2011) We chose LinearSVC primarily because its underlying C implementation is very performant, and because linear kernels performed as well or better than nonlinear ones in early experimentation, corroborating the findings of previous studies. (Garrett et al., 2003; Lotte et al., 2007) We use the default settings for LinearSVC: a C of 1.0, squared hinge loss function, and a tolerance parameter of of 1e-4.

3.4 Experiment 1

For each subject, we generated every pair of two tasks (a binary BCI) and cross-validated our SVM seven times, the recommended default from scikit-learn, on recordings for those two tasks. We used ScikitLearn's built-in cross-validation toolkit, which was configured to perform each of the seven cross-validation steps using different splits of trial data in the training and testing sets. We varied the number of bins in the samples we fed to the SVM and the length of recordings (a slice of our original recording from one second to 1+n seconds). For every task pair processed, we recorded mean classification accuracy across all cross-validation trials.

As an additional performance audit, we timed our SVM at training time and testing time. In this measure, we fit an SVM to all the data for two task-pairs from five randomly-selected subjects, and repeated this process ten thousand times at different bin sizes and different lengths, collecting the minimum time observed in each series of attempts. In order to establish a proper estimate, we time the SVM after all data has been loaded to memory, disregarding the time it takes to load the data from disk.

3.5 Experiment 2

For each subject, we spliced all recordings into 0.5-second chunks, each one representing a single power spectrum reading from our headset. We repeated our previous calibration routine of testing all possible task pairs; however, in contrast to our methods in the last experiment, we trained the SVM only on the first recordings from each user (that is, data collected from the first few trials each subject performed), and tested the SVM only on recordings from later trials. In order to simulate the constraints of quick, naturalistic interaction, we tested on only the first reading (i.e., the first 1/2 second) in each trial. We varied the number of seconds of data used to train the SVM. We also varied the number of bins in each recording. Once again, we measured mean classifier accuracy on all items in our training set.

3.6 Experiment 3

Using the 0.5-second recordings from Experiment 2, we simulated the calibration process of all fourteen subjects. We began with sixty seconds of recording from the three tasks most commonly associated with best-case performance (base, pass, color). We then performed a seven-fold cross-validation on every permutation of two of these tasks (base versus task, pass versus task, pass versus color, etc). The taskpair with the highest mean score across all seven cross-validations was selected for a more robust calibration, in which the reamining 80 seconds of recordings for each taskpair were used to validate the SVM trained on the first 120 seconds of taskpair data. If the resulting score was below 75%, we added the first 60 seconds of recording from the taskpair that was next most correlated with bestcase accuracy. This process was repeated iteratively until all taskpairs were added, or until one taskpair got over 75% accuracy. We attempted this calibration process at a number of bin sizes and recorded the accuracy achieved among subjects and the number of seconds of training data required to achieve the resulting accuracy.

4 RESULTS

4.1 Experiment 1

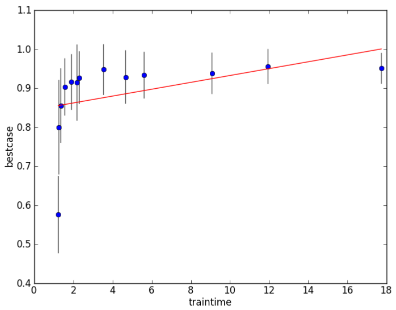

Figure 1: Mean best-case accuracy among all subjects compared to time needed to train the classifier.

We ran an ordinary least squares regression on number of bins and classifier accuracy at recording lengths of four seconds. Number of bins was positively correlated with classifier accuracy, with each bin correpsonding to 1.13% gain in accuracy (R-squared = .773, p <.001).

We ran an additional ordinary least squares regression on number of bins in the EEG data and the time it takes to train an SVM on those data, again at recordings of four seconds. Number of bins was positively correlated with time to train classifier (slope = 0.5 R-squared = .947, p <.001).

Since bin size has a linear relationship with both accuracy and SVM training time, we compare accuracy and training time directly in Figure 1.

4.2 Experiment 2

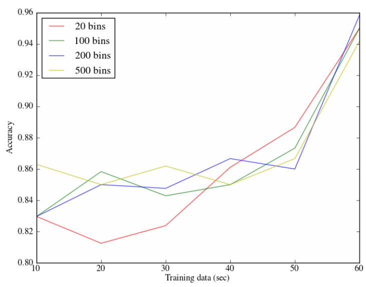

Figure 2: Mean best-case accuracy compared to number of seconds of data in training set.

When using 0.5-second recordings to train the SVM, we found no evidence of a significant difference in SVM accuracy at bin sizes ranging from 20 to 500.

4.3 Experiment 3

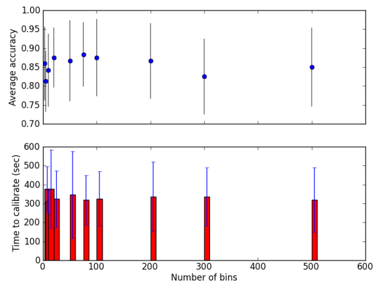

Figure 3: Mean time to calibrate at different bin sizes (bottom) and mean post-calibration classifier accuracy at different bin sizes (top).

still wondering how to report these results.... no statistically significant difference in time-to-calibrate, and very rarely is there a statistically significant difference in the post-calibration accuracy.... .

5 DISCUSSION

We find that logarithmic binning dramatically decreases the computational expense of EEG-based calibration and classification without a significant detriment to accuracy. Further, we find that this technique is compatible with a training strategy capable of reaching acceptable classification rates with only a few minutes of training time. Since we used generallylow-cost, consumer hardware, our technique could make commercial BCI more feasible.

The conclusions to be drawn from this study are limited in a few regards. First, calibration and classification were performed offline, so factors involving the user interface (such as feedback) are not taken into account. We cannot be sure, for instance, that our findings with short splices of ten-recordings data will persist when a system solicits recordings of only a second or under. Furthermore, a few of our tasks (e.g. the color task) relied on exogenous stimuli, which may be impractical in naturalistic settings for ergonomic reasons. Finally, we did not compare our findings to traditional signal processing methods in EEG.

Logarithmic binning could enable co-adaptive, online BCI with as few as one dry EEG sensor, making online calibration much more performant on mobile or embedded processors with limited computational resources. Alternatively, since logarithmic binning dramatically decreases the size of data fed to the classification algorithm, the technique could allow calibration to occur in the cloud - the BCI could pre-process the data on board, bin it, and ship this data to a more powerful server, which could process it online. By some combination of cloud-based and on-board processing, BCIs could gain from the accuracy of computationally expensive analytics without having to perform these computations on-board.

Future work could implement a system that calibrates a BCI online. Due to the small size of binned EEG signals, such a system could use a client/server architecture in which expensive processing (such as training multiple SVMs) is offloaded from the users system to a more powerful processor in teh cloud. This system could be used for the calibration of direct-control BCIs by attempting to find groups of tasks for which the classifier has high discriminatory power. A similar system could be used for long-term, affective recordings as well.

By collecting EEG data in the wild, we hope to discover more about how the mind behaves outside of laboratory environments. A particular interest is the non-stationary nature of neural recordings. By analyzing chronic EEG recordings at scale, we hope to make observations about how EEG signals change their expression over time, which could enable us to build more accurate BCIs that require less frequent re-calibration.

REFERENCES

Blankertz, B., Dornhege, G., Krauledat, M., Mller, K.-R., and Curio, G. (2007a). The non-invasive berlin braincomputer interface: fast acquisition of effective performance in untrained subjects. *NeuroImage*, 37(2):539550.

Blankertz, B., Krauledat, M., Dornhege, G., Williamson, J., Murray-Smith, R., and Mller, K.-R. (2007b). A note on brain actuated spelling with the berlin brain-computer interface. In Stephanidis, C., editor, *Universal Access in Human-Computer Interaction. Ambient Interaction*, number 4555 in Lecture Notes in Computer Science, pages 759–768. Springer Berlin Heidelberg.

Burges, C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

Campbell, A., Choudhury, T., Hu, S., Lu, H., Mukerjee, M. K., Rabbi, M., and Raizada, R. D. (2010). NeuroPhone: brain-mobile phone interface using a wireless EEG headset. In *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, page 38. ACM.

Carrino, F., Dumoulin, J., Mugellini, E., Khaled, O., and Ingold, R. (2012). A self-paced BCI system to control an electric wheelchair: Evaluation of a commercial, low-cost EEG device. In *Biosignals and Biorobotics Conference (BRC), 2012 ISSNIP*, pages 1–6.

Chuang, J., Nguyen, H., Wang, C., and Johnson, B. (2013). I think, therefore i am: Usability and security of authentication using brainwaves. In Adams, A., Brenner, M., and Smith, M., editors, *Financial Cryptography and Data Security*, volume 7862 of *Lecture Notes in Computer Science*, pages 1–16. Springer Berlin Heidelberg.

Crowley, K., Sliney, A., Pitt, I., and Murphy, D. (2010). Evaluating a brain-computer interface to categorise human emotional response. In *ICALT*, page 276278.

D. Mattia, F. Pichiorri, M. M. R. R. (2011). Brain computer interface for hand motor function restoration and rehabilitation. In *Towards Practical Brain Computer Interfaces*. Springer, Biological and Medical Physics, Biomedical Engineering.

Das, D., Chatterjee, D., and Sinha, A. (2013). Unsupervised approach for measurement of cognitive load using EEG signals. pages 1–6. IEEE.

Dornhege, G. (2007). *Toward Brain-computer Interfacing*. MIT Press.

Ekandem, J. I., Davis, T. A., Alvarez, I., James, M. T., and Gilbert, J. E. (2012). Evaluating the ergonomics of BCI devices for research and experimentation. *Ergonomics*, 55(5):592598.

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR: a library for large linear classification. *J. Mach. Learn. Res.*, 9:18711874.

Garrett, D., Peterson, D., Anderson, C., and Thaut, M. (2003). Comparison of linear, nonlinear, and feature selection methods for eeg signal classification. *IEEE*

Transactions on Neural Systems and Rehabilitation Engineering, 11(2):141–144.

Grierson, M. and Kiefer, C. (2011). Better brain interfacing for the masses: progress in event-related potential detection using commercial brain computer interfaces. page 1681. ACM Press.

Hill, N. J., Ricci, E., Haider, S., McCane, L. M., Heckman, S., Wolpaw, J. R., and Vaughan, T. M. (2014). A practical, intuitive braincomputer interface for communicating yes or no by listening. *Journal of Neural Engineering*, 11(3):035003.

Jain, A. K., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition: A review. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(1):437.

Larsen, E. A. and Hokl, C.-s. J. (2011). *Classification of EEG Signals in a Brain- Computer Interface System*.

Leeb, R., Perdikis, S., Tonin, L., Biasiucci, A., Tavella, M., Creatura, M., Molina, A., Al-Khodairy, A., Carlson, T., and Milln, J. d. R. (2013). Transferring braincomputer interfaces beyond the laboratory: Successful application control for motor-disabled users. *Artificial Intelligence in Medicine*, 59(2):121–132.

Lotte, F., Congedo, M., Lcuyer, A., Lamarche, F., Arnaldi, B., et al. (2007). A review of classification algorithms for EEG-based braincomputer interfaces. *Journal of neural engineering*, 4.

Lu, S., Guan, C., and Zhang, H. (2009). Unsupervised brain computer interface based on intersubject information and online adaptation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 17(2):135–145.

McFarland, D. J. and Wolpaw, J. R. (2011). Brain-computer interfaces for communication and control. *Commun ACM*, 54(5):60–66.

Milln, J. D. R., Rupp, R., Mller-Putz, G. R., Murray-Smith, R., Giugliemma, C., Tangermann, M., Vidaurre, C., Cincotti, F., Kbler, A., Leeb, R., Neuper, C., Mller, K.-R., and Mattia, D. (2010). Combining brain-computer interfaces and assistive technologies: State-of-the-art and challenges. *Front Neurosci*, 4.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, . (2011). Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:28252830.

Raudys, S. J. and Jain, A. K. (1991). Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on pattern analysis and machine intelligence*, 13(3):252264.

Shenoy, P., Krauledat, M., Blankertz, B., Rao, R. P. N., and Mller, K.-R. (2006). Towards adaptive classification for BCI. *Journal of Neural Engineering*, 3(1):R13–R23.

Vidaurre, C., Sannelli, C., Mller, K.-R., and Blankertz, B. (2011a). Co-adaptive calibration to improve BCI efficiency. *Journal of Neural Engineering*, 8(2):025009.

Vidaurre, C., Sannelli, C., Mller, K.-R., and Blankertz, B. (2011b). Machine-learning-based coadaptive calibration for brain-computer interfaces. *Neural Computation*, 23(3):791–816.

Vidaurre, C., Schloogl, A., Cabeza, R., Scherer, R., and Pfurtscheller, G. (2006). A fully on-line adaptive BCI. *IEEE Transactions on Biomedical Engineering*, 53(6):1214–1219.