MOABB: Trustworthy algorithm benchmarking for BCIs

Vinay Jayaram*†, Alexandre Barachant‡
* Max Planck Institute for Intelligent Systems, Department Empirical Inference, Tübingen, Germany
Email: vjayaram@tue.mpg.de
† IMPRS for Cognitive and Systems Neuroscience, University of Tübingen, Tübingen, Germany
‡ CTRL-Labs, New-York, USA
Email: alexandre.barachant@gmail.com

***Abstract*—BCI algorithm development has long been hampered by two major issues: Small sample sets and a lack of reproducibility. We offer a solution to both of these problems in the form of a software suite that simplifies and streamlines both the issue of finding, downloading, and preprocessing data in a reliable manner, and the issue of a reliable interface to use machine learning methods. By building on recent advances in software for signal analysis via the MNE toolkit, and the unified framework for machine learning offered by the scikit-learn project, we offer a system that can improve BCI algorithm development. To validate this system, we run analyses of a set of state-of-the-art decoding algorithms across many open access datasets. Our analysis confirms that different datasets can result in very different results for identical processing pipelines, highlighting the need for a trustworthy algorithm benchmarking in the field of BCIs., and further that many previously validated results do not hold up when applied across different datasets, which has wide-reaching implications for practical BCIs.**

I. Introduction

Brain-computer interfaces (BCIs) have long presented the neuroscience methods community with a unique challenge. Unlike fields like vision research, where one simply has a database of images and labels, a BCI is defined by a signal recorded from the brain and fed into a computer, which can be influenced in any number of ways both by the subject and by the experimenter. As a result, validating approaches has always been a difficult task. Number of channels, requested task, physical setup, and many other features vary between the numerous publically available datasets online, not to mention issues of convenience such as file format and documentation. Because of this, the BCI methods community has long done one of two things to validate an new approach: Recorded a new dataset, or used one of few well-known, tried-and-true datasets.

Recording a new dataset, though an elegant way to show that a proposed method works online, presents problems for post-hoc analysis. Without making data public, it is impossible to know whether offline classification results are convincing or due to some coding issue or recording artifact. Further, it is well-known that differences in hardware [1], [2], paradigm [3], and subject [3] can have large differences in the outcome of a BCI task, making it very difficult to generalize findings from any single dataset.

When recording a new dataset is not an option, One can turn to some of the standard EEG datasets publicly available. Over the course of the last year and a half, over a thousand journal and conference submissions have been written on the BCI Competition III [4], [5] and IV [6] datasets. Considering that these datasets have been available publically for over a decade, the true number of papers which validate results against them is simply astronomical. While it is impossible to argue about the impact those two datasets have had on the field, relying so heavily on a small number of datasets, with less than 50 subjects total, expose the field to several important issues. In particular, overfitting to the setups offered there is likely.

Lastly, and possibly most problematically, the scarcity of available code for newly published BCI algorithm puts the onus on each individual lab to reproduce the code for all other competing methods in order to make a claim to be comparable with the 'state-of-the-art' (SOA). As a result, the vast majority of novel BCI algorithm papers compare either against other work from the same lab, or old standards such as CSP [7], with or without regularization, and LDA, or simple channel-level variances combined with a classifier of choice [8] .

Computer vision has solved this problem with enormous datasets like [9] bundled with a reliance on a small number of software packages to create new models, notably Tensorflow, Theano, and Pytorch. As BCIs are inextricably linked to human use, it is not helpful to create datasets of such size. Rather, the field requires as many unique people recording data in many contexts in order to create an appropriate benchmark. In contrast to image data, the goal of BCI algorithm development is exclusively to create algorithms that work on data that has not yet been recorded. We propose our platform, the MOABB (Mother Of All BCI Benchmarks) Project, as a candidate for this application.

As an initial validation of this project, we present results on the constrained task of binary classification in two-class imagined motor imagery, as that is the most widely used motor imagery paradigm and allows us to demonstrate the process across the largest number of datasets. However, we note that this is only the first question we attempt to answer in this field. The format allows for many other questions, including different channel types (EEG, fNIRS, or other), multi-class paradigms, and also transfer learning scenarios as described in [10]. ...yeah it's a bit gratuitous. Too much?

II. Methods

Any BCI analysis is defined by three things: A dataset, a context, and a processing pipeline. Here we describe how all of these components are dealt with within our pipeline, and how specifically we set the options for the initial analyses presented here.

A. Datasets

Public BCI datasets exist for a wide range of user paradigms and recording conditions, from continuous usage to single-session to multiple-sessions-per-subject. Within the current MOABB project, we have unified the access to many datasets, described in Table I.

Adding new datasets that can be found on the internet is also simple. The MNE toolkit[21], [22] is used for all preprocessing and channel selection, so any dataset that can be made compatible with their framework can quickly be added to the set of data offered by this project. In addition, the project has test functions to ensure candidate code conforms to the software interface.

B. Context

A *context* is the set of characteristics that defines the preprocessing and validation procedure. To go from a recorded EEG time-series to a pipeline performance value for a given subject or recording session, many parameters must be defined. Trials need to be cut out of the continuous signal and pre-processed, and this is possible in many different ways when taking into parameters such as trial overlap, trial length, imagery type, and more. Once the continuous data is processed into trials, and these trials are fed into a pipeline, the question of how to create training and test sets, and how to report performance, comes into play. We separate these two notions in our software and call them the *paradigm* and the *evaluation* respectively.

1) Paradigm: A paradigm defines how one goes from continuous data to trials for a standard machine learning algorithm to deal with. While not an issue in image processing, it is crucial in EEG and biosignals processing because most datasets do not have exactly the same events defined in the continuous data. For example, many motor imagery sets use a single hand versus both feet imagery, or a single class versus rest; likewise there are many non-motor imageries possible. For any reasonable analysis the specific sort of imagery or ERP must be controlled for, as they all have different characteristics in the data and further are variably effective across subjects [19], [3]. After choosing which events or imageries are valid, the question comes to pre-processing of the continuous data, in the form of ICA cleaning, bandpass filtering, and so on. These must also be identical for valid comparisons across algorithm or datasets. Lastly, there are questions of how to cut the data into trials: What is the trial length and overlap; or, in the case of ERP paradigms, how long before and after the event marker do we use? The answers to all these questions are summed up in the paradigm object.

2) Evaluation: Once the data is split into trials and a pipeline is fixed, there are many ways to train and test this pipeline to minimize overfitting. For datasets with multiple subjects recorded on multiple days, we may want to determine which algorithm functions best in multi-day classification. Or, we may want to determine which algorithm is best for small amounts of data. It is easy to see that there are many possibilities for splitting data into train and test sets, and these must be fixed identically for a given analysis. Furthermore, there is the question of how to report results. Multiclass problems cannot use metrics like the ROC-AUC which provide unbiased estimates of classifier goodness in binary cases; depending on things like the class balance, various metrics have various benefits and pitfalls. Therefore this must also be fixed across all datasets, contingent on the class of predictions the pipelines attempt to make. We define this as our *evaluation*.

C. Pipeline

We define a *pipeline* as the processing that takes one from raw trial-wise data into labels, taking both spatial filtering and classification model fitting into account. A convenient API for dealing with this kind of processing is defined by scikit-learn[23], which allows for easily definable dimensionality reduction, feature generation, and model fitting. To maximize reproducibility we allow pipelines to be defined either by yaml files or through python files that generate the objects, but force all machine learning models to follow the scikit-learn interface.

III. Statistical Analysis

At the end of processing there are scores for every subject in every dataset with every pipeline. The goal of this project is to synthesize these numbers into an estimate of how likely it is that each pipeline out-performs the other pipelines. However, even if imagery type and channel number were held constant, differences in trial amount, sampling rate, and even location and hardware mean that we cannot expect subjects across datasets to be naively comparable. Simply pooling them all and running a paired-sample test would result in artificially inflated significance due to these factors. This would suggest a mixed-factor ANOVA for every unique pair of pipelines to test the null hypothesis that the difference distribution is zero-mean over all datasets. However, we have the secondary problem that this difference distribution, even within a given dataset, is very unlikely to be Gaussian (which is an assumption of an ANOVA). It is well-known that some subjects are BCI illiterate, which implies that no pipeline can reliably out-predict another one on that subset of subjects. Therefore, for large enough datasets, the distribution of differences in pipeline scores is very likely to be at least bimodal.

Thankfully, a similar statistical situation has already arisen in the field of medical studies. It is often the case that one wants to measure a quantity from multiple groups of subjects (different hospitals, for examples) at multiple time points, then ask if there was a significant effect of time in the measurements. This type of study design is called a *factorial design* and is equivalent to the design described above (where

Name	Imagery	# Channels	Avg # Trials	# Sessions	# Subjects	Epoch	Citations
Cho et al. 2017	Right, left hand	64	200	1	49	0-3s	[11]
Physionet	Right, left hand	64	45	1	109	1-3s	[12], [13]
Shin et al. 2017	Right, left hand	25	60	1	29	0-10s	[14], [15]
BNCI 2014-001	Right, left hand	22	288	2	9	2-6s	[6]
BNCI 2014-002	Right hand, feet	15	160	1	14	3-8s	[16]
BNCI 2014-004	Right, left hand	3	720	2	9	3-7.5s	[17]
BNCI 2015-001	Right hand, feet	13	400	2	13	3-8s	[18]
BNCI 2015-004	Right hand, feet	30	155	2	10	3-10s	[19]
Alexandre Motor Imagery	Right hand, feet	16	40	1	9	0-3s	[20]

TABLE I: Dataset attributes

instead of time we have different pipelines). Restricting each test to two pipelines gives us a way of generating a p-value for the difference between the two. To deal with the non-Gaussianity, we use non-parametric, rank-based computation of the test statistics as described in [24], and we can use Bonferroni correction to deal with the fact that each pipeline will appear $N_{\text{pipelines}} - 1$ times.

IV. Experiments

To show off the possibilities of this framework, we ran various well-known BCI pipelines from across many papers in order to conduct the first big-data, side-by-side analysis of the state of the art in motor imagery BCIs.

A. Context

For the human paradigm, we choose to look at datasets including motor imagery. Motor imagery is the most-studied sort of imagery for BCIs [25], and we further limit ourselves to the binary case as this has not yet been solved. For evaluations, we choose within-session cross-validation, as this represents the best-case scenario for any pipeline, with minimal non-stationarity.

1) Paradigm: As there are many methods that show that multiple frequency bands can lead to improved BCI performance[26], and further that discriminative data is concentrated in the anatomical frequency bands, we test three preprocessing pipelines: A single bandpass containing both the alpha and beta ranges, from $8-32$Hz, another from $8-32$Hz in 4Hz increments, and a third with the δ, θ, α and β frequencies isolated.

2) Evaluation: The evaluation was chosen to be within-session, as that minimizes the effect of non-stationarity. As this is a binary classification task, the ROC-AUC score was chosen as the metric to score 5-fold cross validation (the splits were kept identical for all pipelines). In order to return a single score per subject, the scores from each session were averaged when multiple sessions were present.

B. Pipelines

Given the amount sheer breadth of models implemented within scikit-learn, attempting an exhaustive model comparison would be impossible. Instead, we implement a selection of pipelines from the BCI literature, as well as the well-known standards of CSP + LDA and channel-level variances + LDA. Specific implemented pipelines are as follows; all hyperparameters were set via cross-validation:

V. Results

The results from analyses with this data will be displayed in three static plots, described below:

1) Score plot: Figure 1 displays the raw results as a point plot per algorithm and dataset. With a reasonably small number of datasets as we have here, the plot is quite informative; however, as the number of datasets increases it will become more and more dense. It describes the distribution of scores over subjects within a single dataset and for a single algorithm.

2) Ranking plot: The ranking plot is intended to be a quick way of ordering algorithms, based on the statistical significance returned by the procedure described in Section III. It describes, which pipelines significantly out-performed others across all datasets.

3) Paired plot: Figures 2 and 3 show ranking plots. In order to look at two algorithms in more detail, for particular pairs of algorithms we also offer a plot of the score in one versus the other, for all subjects across all datasets. This is computed for all pairs of algorithms, but for use in a publication it is cumbersome to add all the plots.

4) Time plot: Figure 4 shows a plot of training size versus fitting time for the model. In addition to accuracy, model fitting time can also affect which pipeline is most appropriate for a given situation. This is a function of both the number of channels, timepoints, and samples, but a point cloud in three-dimensions is difficult to visualize on paper. As such, for these written reports, a plot is generated of the time versus the number of entries in the training matrix, which is the product of the number of channels and the number of samples (the number of timepoints is less important as features are usually computed over all of them). While some of the nuance is lost, it is still easy to see how algorithms compare to each other.

We would last like to note that the results are too complex to easily simplify into static plots. What if one, for example, wants to know whether a pipeline only performs well on a subset of datasets? Because of this an interactive visualization would be strongly preferred.

Figure 1 shows all the results generated by this entire processing chain. Surprisingly, perhaps, the pipelines do not clearly cluster on the dataset level, making it unclear which ones perform best from simply this plot. What is very clear, however, is that different datasets have very clearly different average scores independent of pipeline.

Figure 2 shows the pairwise comparison of the log variance-based pipeline with CSP and the tangent space SVM. What is very clearly shown both in the plots and the statistics is

Name	Preprocessing	Classifier	
CSP + LDA	Trial covariances estimated via maximum-likelihood with unregularized common spatial patterns (CSP). Features were log variance of the filters belonging to the 6 most diverging eigenvalues	Linear Discriminant Analysis (LDA)	[7]
RegCSP + shLDA	Trial covariances estimated by OAS (Chen) with unregularized CSP. This is equivalent to Tikhonov regularization as described in [27]. Features were log variance on the 6 top filters.	LDA with Ledoit-Wolf shrinkage of the covariance term	[27]
rieCSP + shLDA	Trial covariances estimated via maximum-likelihood, CSP class-wise matrices were Riemannian mean of the trial-wise matrices.	LDA with Ledoit-Wolf shrinkage of the covariance term	[**?**]
FBCSP + optSVM	Filter bank of 6 bands between 8 and 35 Hz followed by OAS covariance estimation and unregularized CSP. Log variance from each of the 4 top filters from each sub-band were pooled and the top 10 features chosen by mutual information were used.	A linear support vector machine was trained with its regularization hyperparameter set by a cross-validated grid-search from $[0.01100]$.	[26]
TS + optSVM	Trial covariances estimated via OAS then projected into the Riemannian tangent space to obtain features	Linear SVM with identical grid-search	[28]
AM + optSVM	Log variance in each channel	Linear SVM with grid-search	N/A
FB-AM + optLR	Log variance from each channel in the five anatomical frequency bands $(\delta, \theta, \alpha, \beta, \gamma)$	To replicate the genetic algorithm and channel selection used in the paper, as well as the linear final boundary, we use a logistic regression classifier with an L1 penalty, hyperparameter optimized via grid search	[8]

TABLE II: Processing pipelines

that, for within-session scoring, it is strongly out-performed by both.

Figure 3 shows the pairwise comparison of the pipeline with plain CSP with the two regularized approaches and with tangent space SVM. For both the regularized approaches. while there is a great deal of variance across subjects, the score from CSP and the regularized CSP roughly track each other, such that there is no significant difference between them. The only significant difference is with the tangent space method.

Figure 4 shows how all the described methods compare in terms of processing time. Unsurprisingly, the methods based on Riemannian computation are more computationally expensive at large sample sizes than the other methods (due to the iterative computations and the feature number increase).

VI. Discussion

We present a system for reliably comparing BCI pipelines that is both easily extended to incorporate new datasets and equipped with an automated statistical procedure for determining which pipelines perform best. Furthermore, this system defines a simple interface for submitting and validating new BCI pipelines, which could serve to unify the many methods that exist so far. To test that system, we present results using standard pipelines in contexts that have wide relevance to the BCI community. By looking across multiple, large datasets, it is possible to make statements about how BCIs perform on average, without any sort of expert tuning of the processing chain, and further to see where the major pitfalls still lie.

The results of this analysis suggest that many well-known methods do not reliably out-perform simpler ones, despite the small-scale studies done years ago to validate them. In particular, the world of CSP regularization literature does not appear to have the effect that was originally claimed. It shows, surprisingly, that the major difference in BCI classification isn't actually the algorithm, as of now, but rather the recording and human paradigm characteristics. The two most clear findings to come out of this are that log variances on the channel level are almost never better than CSP or Riemannian methods, and that the tangent space classification pipeline has the best model for single-session classification.

One crucial thing to keep in mind, looking at these results, is that this was all done on within-session classification. Within a single recording session the non-stationarity that has long plagued BCis is kept to a minimum, meaning that regularization is at its least effective. The proper conclusion would therefore not be that regularization does not help CSP, but rather that regularization is not necessary to combat within-recording signal non-stationarity. To determine whether regularization helps across time, it is necessary to do a cross-session evaluation. Unfortunately, there are fewer datasets that have multiple sessions recorded.

The analysis here, though done with over 200 subjects, is still only a fraction of the number of subjects recorded for BCI publications over the years. With more papers that describe more varied setups, the power of this system can only grow, and what this analysis shows most clearly is that the sample size problem in BCIs is bigger than we might have expected. By gathering the data and offering a system for testing algorithms, we hope that this platform in the coming years can help to solve it.

References

[1] A. Searle and L. Kirkup, "A direct comparison of wet, dry and insulating bioelectric recording electrodes," *Physiological measurement*, vol. 21, no. 2, p. 271, 2000.

[2] M. A. Lopez-Gordo, D. Sanchez-Morillo, and F. P. Valle, "Dry EEG electrodes," *Sensors*, vol. 14, no. 7, pp. 12 847–12 870, 2014.

[3] B. Z. Allison and C. Neuper, "Could Anyone Use a BCI?" in *Brain-Computer Interfaces: Applying our Minds to Human-Computer Interaction*, D. S. Tan and A. Nijholt, Eds. London: Springer London, 2010, pp. 35–54. [Online]. Available: https://doi.org/10.1007/978-1-84996-272-8_3

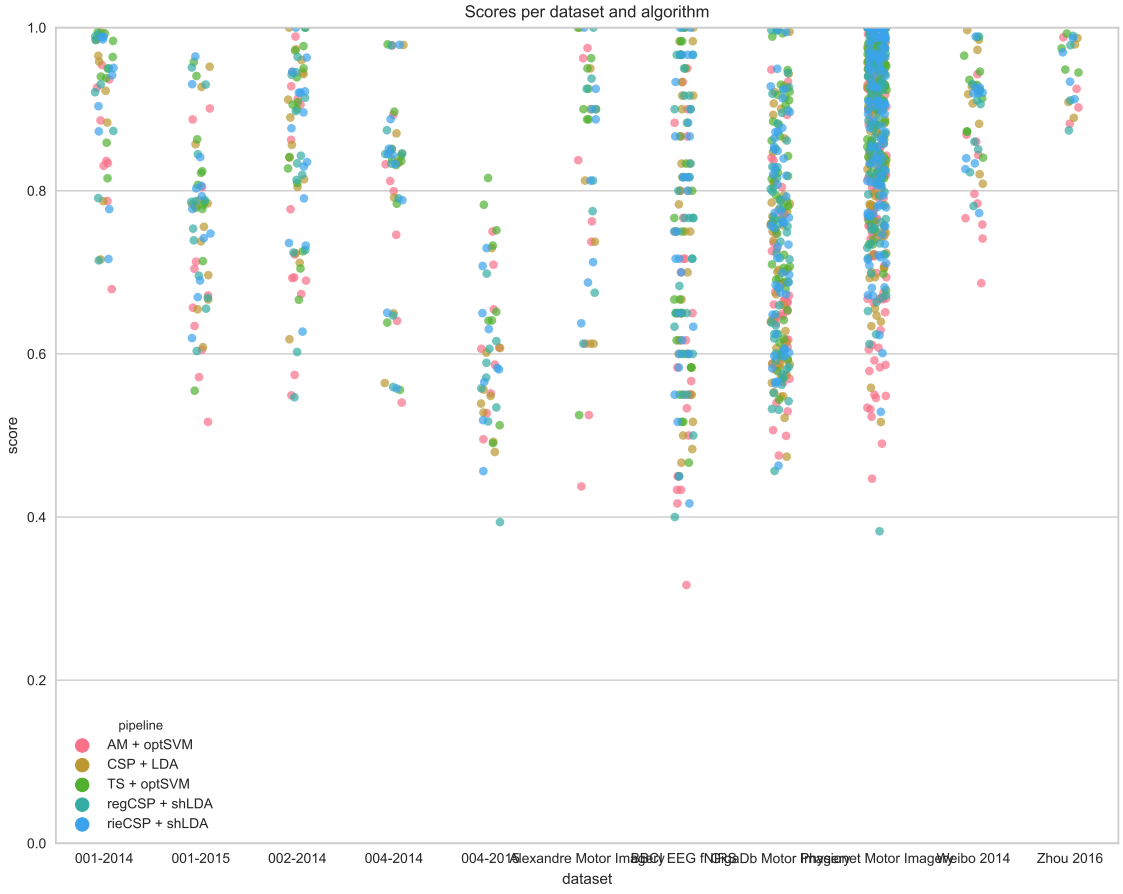

Fig. 1: Visualization of all generated scores, across all datasets.

[4] B. Blankertz, K. R. Müller, D. Krusienski, G. Schalk, J. R. Wolpaw, A. Schloegl, G. Pfurtscheller, J. R. Millan, M. Schroeder, and N. Birbaumer, "The {BCI} Competition {III}: Validating Alternative Approaches to Actual {BCI} Problems," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 14, no. 2, pp. 153–159, 2006.

[5] A. Schloegl, "Results of the {BCI} Competition 2005 for data set {IIIa} and {IIIb}," Institute for Human-Computer Interfaces - {BCI} Lab, University of Technology Graz, Austria, Tech. Rep., 2005.

[6] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K. J. Miller, G. R. Müller-Putz, G. Nolte, G. Pfurtscheller, H. Preissl, G. Schalk, A. Schlögl, C. Vidaurre, S. Waldert, and B. Blankertz, "Review of the {BCI} Competition {IV}," *Frontiers in Neuroscience*, vol. 6, p. 55, jul 2012. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnins.2012.00055/abstract

[7] Z. J. Koles, M. S. Lazar, and S. Z. Zhou, "Spatial patterns underlying population differences in the background {EEG}," *Brain Topography*, vol. 2, no. 4, pp. 275–284, 1990.

[8] D. Garrett, D. Peterson, C. Anderson, and M. Thaut, "Comparison of linear, nonlinear, and feature selection methods for {EEG} signal classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 11, no. 2, pp. 141–144, jun 2003. [Online]. Available: http://ieeexplore.ieee.org/document/1214704/

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 248–255.

[10] V. Jayaram, M. Alamgir, Y. Altun, B. Scholkopf, and M. Grosse-Wentrup, "Transfer Learning in Brain-Computer Interfaces," *Computational Intelligence Magazine, IEEE*, vol. 11, no. 1, pp. 20–31, 2016.

[11] H. Cho, M. Ahn, S. Ahn, M. Kwon, and S. C. Jun, "EEG datasets for motor imagery braincomputer interface," *GigaScience*, vol. 6, no. 7, pp. 1–8, jul 2017. [Online]. Available: http://academic.oup.com/gigascience/article/6/7/1/3796323/EEG-datasets-for-motor-imagery-braincomputer

[12] G. Schalk, D. McFarland, T. Hinterberger, N. Birbaumer, and J. Wolpaw, "BCI2000: A General-Purpose Brain-Computer Interface (BCI) System," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1034–1043, jun 2004. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/15188875 http://ieeexplore.ieee.org/document/1300799/

[13] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet," *Circulation*, vol. 101, no. 23, pp. e215 LP – e220, jun 2000. [Online]. Available: http://circ.ahajournals.org/content/101/23/e215.abstract

[14] B. Blankertz, G. Dornhege, M. Krauledat, K. R. Müller, and G. Curio, "The non-invasive Berlin Brain-Computer Interface: Fast acquisition of effective performance in untrained subjects," *NeuroImage*, vol. 37, no. 2, pp. 539–550, 2007.

[15] J. Shin, A. von Luhmann, B. Blankertz, D.-W. Kim, J. Jeong, H.-J. Hwang, and K.-R. Muller, "Open Access Dataset for EEG+NIRS Single-Trial Classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 10, pp. 1735–1745, oct 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7742400/

[16] D. Steyrl, R. Scherer, J. Faller, and G. R. Müller-Putz, "Random forests

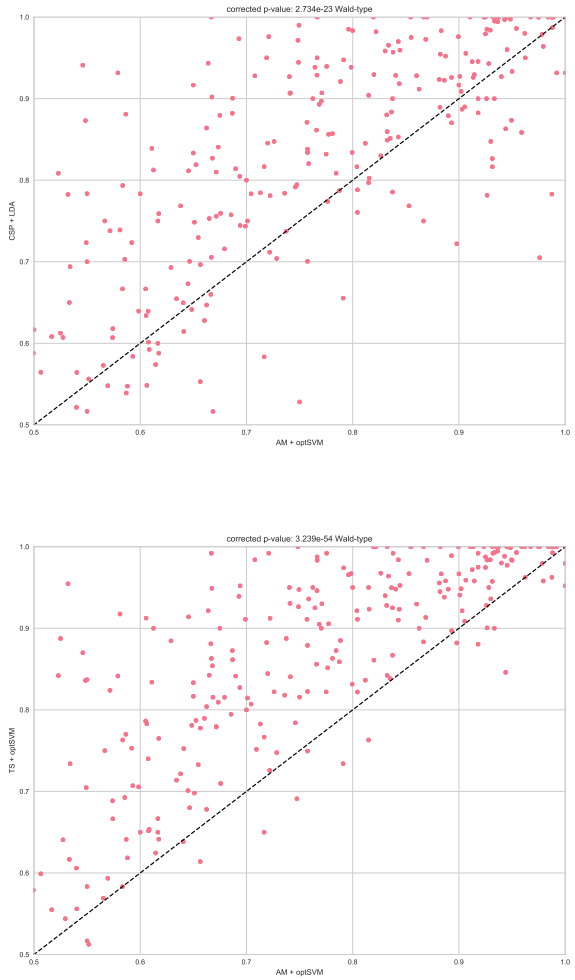

Fig. 2: Paired plots of log variance versus basic CSP and the tangent space projection. It does consistently worse than both CSP and the tangent space method.

in non-invasive sensorimotor rhythm brain-computer interfaces: a practical and convenient non-linear classifier," *Biomedical Engineering / Biomedizinische Technik*, vol. 61, no. 1, pp. 77–86, jan 2016. [Online]. Available: http://www.degruyter.com/view/j/bmte.2016.61.issue-1/bmt-2014-0117/bmt-2014-0117.xml

[17] R. Leeb, F. Lee, C. Keinrath, R. Scherer, H. Bischof, and G. Pfurtscheller, "BrainComputer Communication: Motivation, Aim, and Impact of Exploring a Virtual Apartment," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 4, pp. 473–482, dec 2007. [Online]. Available: http://ieeexplore.ieee.org/document/4359220/

[18] J. Faller, C. Vidaurre, T. Solis-Escalante, C. Neuper, and R. Scherer, "Autocalibration and Recurrent Adaptation: Towards a Plug and Play Online ERD-BCI," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 20, no. 3, pp. 313–319, may 2012. [Online]. Available: http://ieeexplore.ieee.org/document/6177271/

[19] R. Scherer, J. Faller, E. V. C. Friedrich, E. Opisso, U. Costa, A. Kübler, and G. R. Müller-Putz, "Individually Adapted Imagery Improves Brain-Computer Interface Performance in End-Users with Disability," *PLOS ONE*, vol. 10, no. 5, p. e0123727, may 2015. [Online]. Available: http://dx.plos.org/10.1371/journal.pone.0123727

[20] A. Barachant, "Commande robuste d'un effecteur par une interface cerveau machine EEG asynchrone," mar 2012. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01196752/

[21] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. S. Hämäläinen, "MNE software for processing MEG and EEG data," *NeuroImage*, vol. 86, pp. 446–460, feb 2014. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/24161808 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC3930851 http://linkinghub.elsevier.com/retrieve/pii/S1053811913010501

[22] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, and M. Hämäläinen, "MEG and EEG data analysis with MNE-Python," *Frontiers in Neuroscience*, vol. 7, p. 267, dec 2013. [Online]. Available: http://journal.frontiersin.org/article/10.3389/fnins.2013.00267/abstract

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011. [Online]. Available: http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

[24] K. Noguchi, Y. R. Gel, E. Brunner, and F. Konietschke, "nparLD: An R Software Package for the Nonparametric Analysis of Longitudinal Data in Factorial Experiments," *Journal of Statistical Software*, vol. 50, no. 12, 2012. [Online]. Available: https://goescholar.uni-goettingen.de/handle/1/9492

[25] H. Yuan and B. He, "Brain-computer interfaces using sensorimotor rhythms: current state and future perspectives." *IEEE transactions on bio-medical engineering*, vol. 61, no. 5, pp. 1425–35, may 2014. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/24759276 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4082720

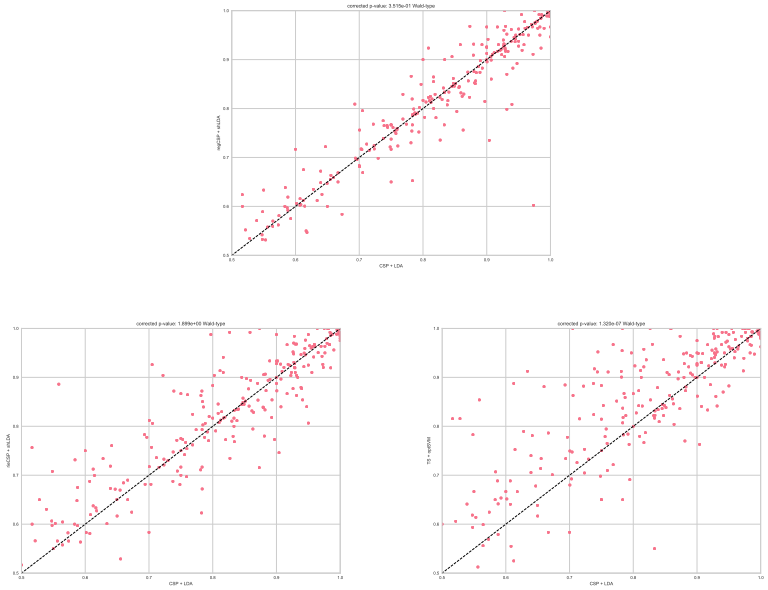

Fig. 3: Paired plots of CSP versus the two regularized methods and the tangent space SVM. Interestingly, across datasets neither of the regularization methods does significantly better in within-session classification. However, the tangent space method does reliably out-perform it

[26] Kai Keng Ang, Zhang Yang Chin, Haihong Zhang, and Cuntai Guan, "Filter Bank Common Spatial Pattern (FBCSP) in Brain-Computer Interface," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, jun 2008, pp. 2390–2397. [Online]. Available: http://ieeexplore.ieee.org/document/4634130/

[27] F. Lotte and C. Guan, "Regularizing common spatial patterns to improve {BCI} designs: unified theory and new algorithms," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 2, pp. 355–362, 2011.

[28] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Classification of covariance matrices using a Riemannian-based kernel for BCI applications," *Neurocomputing*, vol. 112, pp. 172–178, jul 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231213001574

corresponding author: Vinay Jayaram (email: vjayaram@tue.mpg.de)

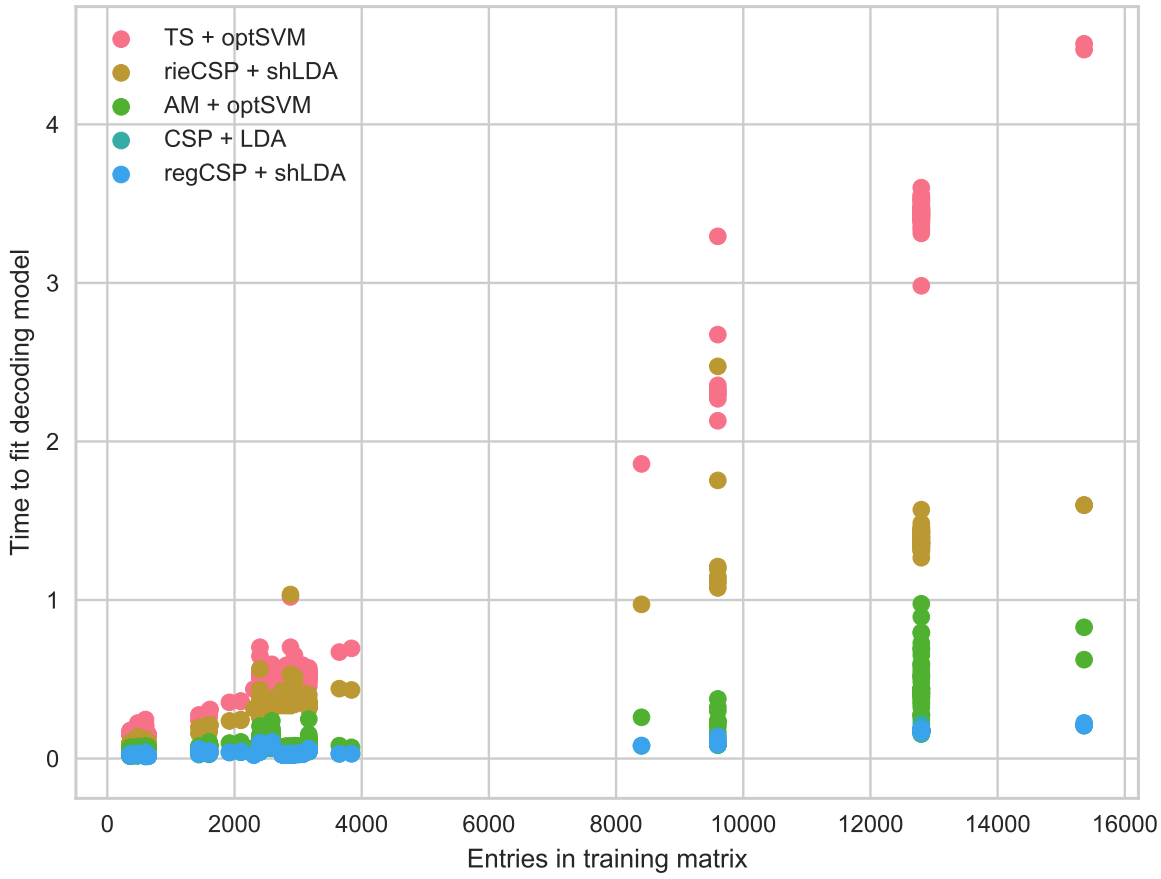

Fig. 4: Time plot of all pipelines across all datasets, as a function of number of entries in the training matrix