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 via a software suite that streamlines both the issues of finding and preprocessing data in a reliable manner, as well as that of using a consistent interface for machine learning methods. By building on recent advances in software for signal analysis implemented in 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 analyze of a set of state-of-the-art decoding algorithms across 12 open access datasets, with over 250 subjects. Our analysis confirms that different datasets can result in very different results for identical processing pipelines, highlighting the need for trustworthy algorithm benchmarking in the field of BCIs, and further that many previously validated methods 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 in vision research, where one 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, the ideal way to show that a proposed method works in practice, 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 **Searle2000**, **Lopez-Gordo2014** paradigm **Allison2010** and subject **Allison2010** can have large differences in the outcome of a BCI task, making it very difficult to generalize findings from any single dataset.

Over the years many datasets have been published online, and serve as an attractive option when time or hardware do not permit recording a new one. In the last year and a half, over a thousand journal and conference submissions have been written on the BCI Competition III **Blankertz2006**, **Schloegl2005** and IV **Tangermann2012** datasets. Considering that these datasets have been available publically for over a decade, the true number of papers which validate results against them is likely much higher. While it is impossible to deny the impact these two datasets have had on the field, relying so heavily on a small number of datasets – with less than 50 subjects total – exposes 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 BCI algorithms old and new 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, easily implementable standards such as CSP **Koles1990** or channel-level variances combined with a classifier of choice **Garrett2003**

Computer vision has solved this problem with enormous datasets like **Deng2009** 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 reasonable to create datasets of such size. Rather, the field requires many different 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 **Jayaram2016**

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 open-source datasets is also simple. The MNE toolkit **Gramfort2014**, **Gramfort2013** 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 offers 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. First, trials need to be cut out of the continuous signal and pre-processed, which is possible in many different ways when taking into account 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 next 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 pipeline to deal with. While not an issue in image processing, as each trial is just one image, 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 datasets with two-class motor imagery use left versus right hand, while some use hands versus feet; there are also many possible non-motor imageries. 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 **Scherer2015a**, **Allison2010** 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 training data. It is easy to see that there are many possibilities for splitting data into train and test sets depending on the question to be answered, 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 other metrics have their own 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 **Pedregosa2011** 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 misleading significances 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**Allison2010** 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.

To deal with this issue while also keeping the framework running fast enough to execute on a normal desktop, we use

Name	Imagery	# Channels	# Trials	# Sessions	# Subjects	Epoch	Citations
Cho et al. 2017	Right, left hand	64	200	1	49	0-3s	**Cho2017**
Physionet	Right, left hand	64	40-60	1	109	1-3s	**Schalk2004**, **Goldberger2000**
Shin et al. 2017	Right, left hand	25	60	3	29	0-10s	**Blankertz2007**, **Shin2017**
BNCI 2014-001	Right, left hand	22	144	2	9	2-6s	**Tangermann2012**
BNCI 2014-002	Right hand, feet	15	160	1	14	3-8s	**Steyrl2016**
BNCI 2014-004	Right, left hand	3	120-160	5	9	3-7.5s	**Leeb2007**
BNCI 2015-001	Right hand, feet	13	200	2/3	13	3-8s	**Faller2012**
BNCI 2015-004	Right hand, feet	30	70-80	2	10	3-10s	**Scherer2015a**
Alexandre Motor Imagery	Right hand, feet	16	40	1	9	0-3s	**Barachant2012a**
Yi et al. 2014	Right, left hand	60	160	1	10	3-7s	**Yi2014**
Zhou et al. 2016	Right, left hand	14	100	3	4	1-6s	**Zhou2016**
Grosse-Wentrup et al. 2009	Right, left hand	128	300	1	10	3-10s	**Grosse-Wentrup2009**
Total:					**275**		

TABLE I: Dataset attributes

a mixture of permutation and non-parametric tests. Within each dataset, either a one-tailed permutation-based paired t-test or Wilcoxon sign-rank test is run (depending on the number of subjects) for each pair of pipelines, generating a p-value for the hypothesis that pipeline a is bigger than pipeline b for each pair of pipelines. These p-values are combined via Stouffer's method, with a weighting given by the square root of the number of subjects, to return a final p-value for each hypothesis. Since each score is compared against $N_{pipelines} - 1$ other scores for the same subject, we also apply Bonferroni correction to protect against false positives. In order to determine effect size, we computed the standardized mean difference within datasets and combined them using the same weighting as was given to Stouffer's method.

IV. Experiment

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 paradigm, we choose to look at datasets including motor imagery. Motor imagery is the most-studied sort of imagery for BCIs **Yuan2014** 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**KaiKengAng2008** and further that discriminative data is concentrated in the anatomical frequency bands, we test two preprocessing pipelines: A single bandpass containing both the alpha and beta ranges, from $8-32$Hz, and another from $8-32$Hz in 4Hz increments. All data was also subsampled to 128Hz, as the memory requirements became prohibitive otherwise.

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 a given subject). In comparison with the more interpretable classification accuracy, the ROC-AUC is less sensitive to imbalanced classes, which is important in this case where the datasets vary heavily. 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 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 + SVM. Specific implemented pipelines are in Table II; all hyperparameters were set via cross-validation.

V. Results

The results from analyses with this data will be displayed in three plots, generated automatically during processing.

1) Score plot: Figure 1 displays the raw results as a point plot per algorithm and dataset.

2) Ordering 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. Greyed-out squares show situations in which the y-axis pipeline did not significantly improve over the x-axis pipeline; values in green squares show the standardized mean difference over all datasets for that particular pair.

3) Paired plot: Figures 3 and 4 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 score in the other, for all subjects across all datasets. This is computed for all pairs of algorithms, and we show a selection here that details results for some of the better-known methods.

4) Time plot: Figure 5 shows a plot of accuracy 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, though obviously poor-performing models are not ideal regardless of accuracy. As there is a large variety in channels and trial numbers among the datasets in this analysis, we can infer the sensitivity of the fitting to the input size from the variance in the X direction, and the range of scores from the Y direction. The best pipeline would lie in the top right, while the worst would be in the bottom left.

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)	**Koles1990**
RegCSP + shLDA	Trial covariances estimated by OAS (Chen) with unregularized CSP. This is equivalent to Tikhonov regularization as described in **Lotte2011** Features were log variance on the 6 top filters.	LDA with Ledoit-Wolf shrinkage of the covariance term	**Lotte2011**
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	**Barachant2010a**
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].	**KaiKengAng2008**
TS + optSVM	Trial covariances estimated via OAS then projected into the Riemannian tangent space to obtain features	Linear SVM with identical grid-search	**Barachant2013**
AM + optSVM	Log variance in each channel	Linear SVM with grid-search	N/A

TABLE II: Processing pipelines

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 different average scores independent of pipeline. The overall ranking of algortihms can be found in Figure 2.

Figure 3 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 that, for within-session scoring, it is strongly out-performed by both.

Figure 4 shows the pairwise comparison of 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 5 shows how all the described methods compare in terms of processing time. 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 increase in feature number).

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. Rather, the major difference in BCI classification isn't actually the algorithm, as of now, but 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.

Looking at these results, one crucial thing to keep in mind 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 as of now too few datasets with multiple recorded sessions.

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.

VII. Conclusion

We present the MOABB project, a codebase that simplifies the application of machine learning methods to EEG data and provides an interface to reliable, reproducible BCI methods results. We validate the initial version of this analysis using 275 subjects drawn from many different labs and published open-access online, and show that with more data many previously validated conclusions come into question.

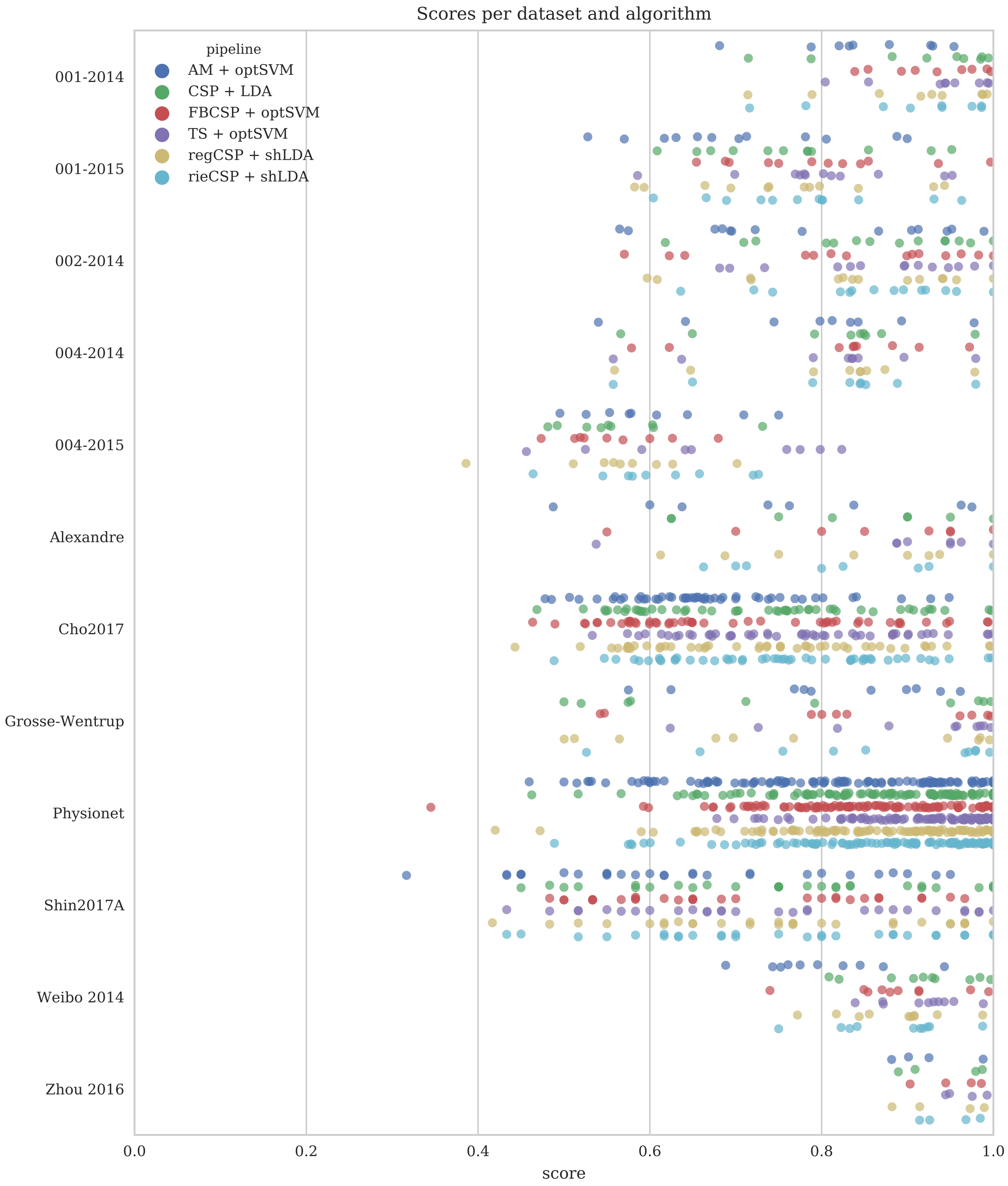

Fig. 1: Visualization of all generated scores, across all datasets.

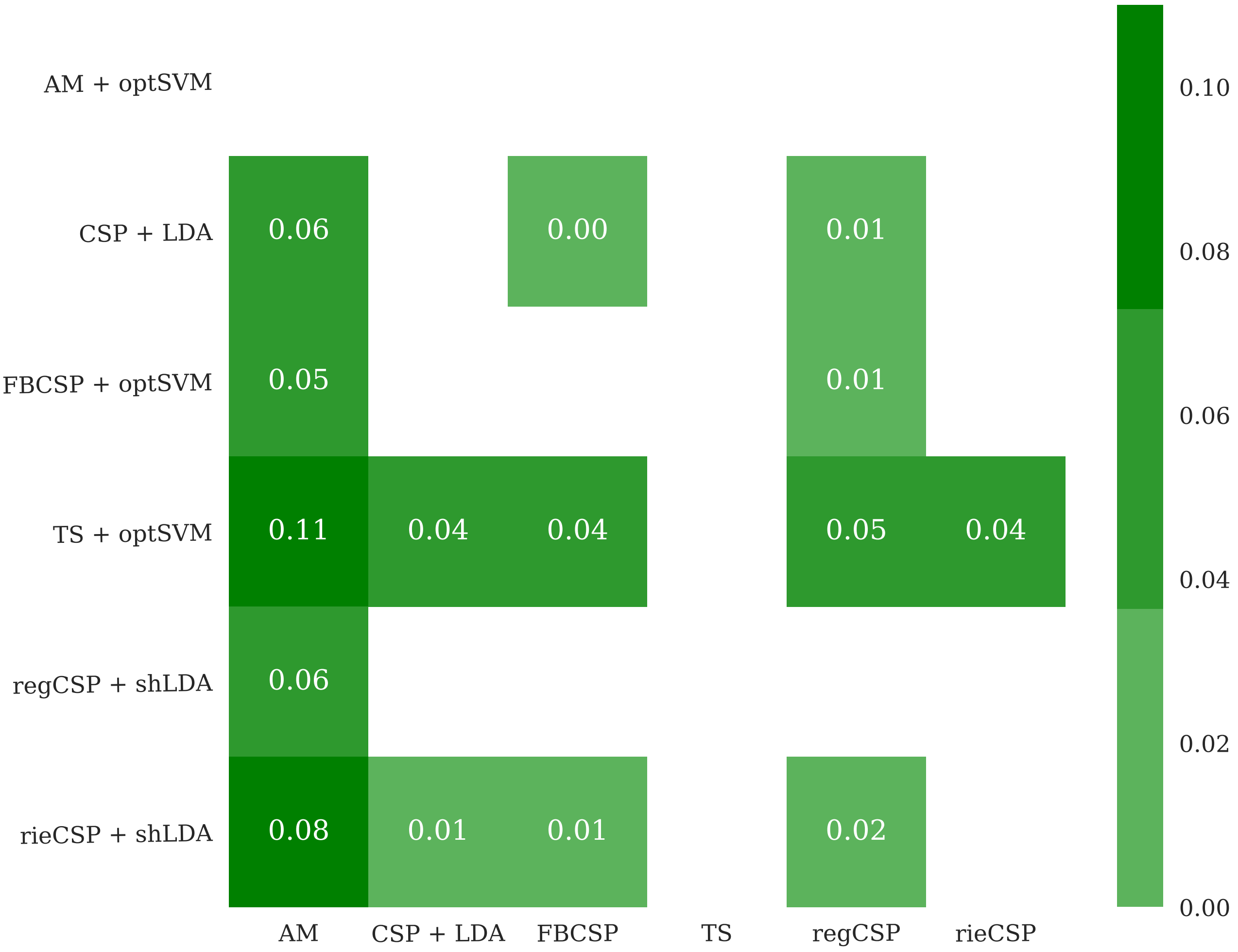

Fig. 2: Ranking of algorithms in performance across all datasets with statistics generated as defined in section III. The numbers correspond to the standardized mean difference of the score in the y-axis minus that in the x-axis; grey boxes represent pairs where the algorithm on the y-axis is not significantly larger than the algorithm on the x.

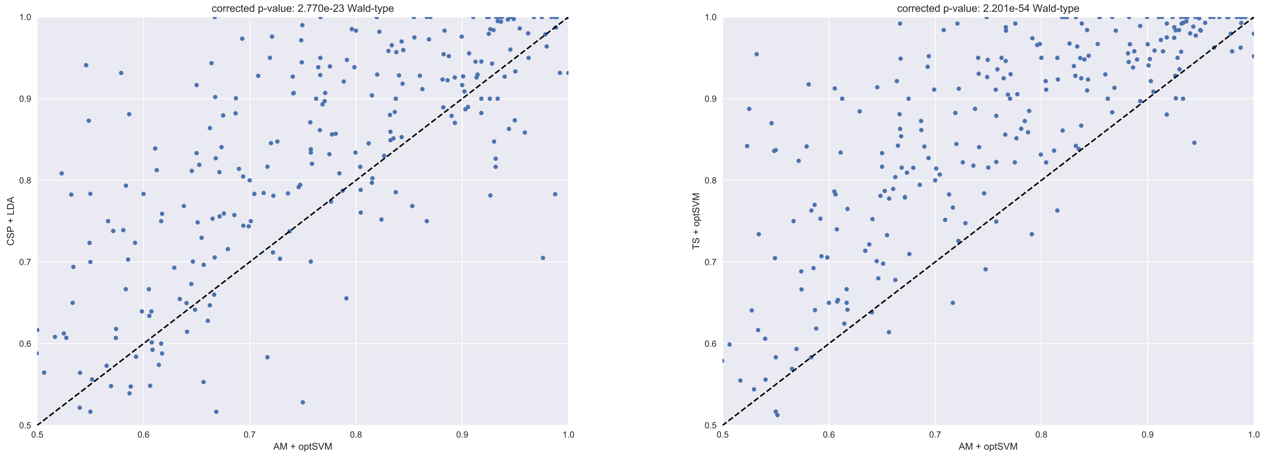

Fig. 3: Paired plots of log variance versus basic CSP and the tangent space projection. It does consistently worse than both CSP and the tangent space method.

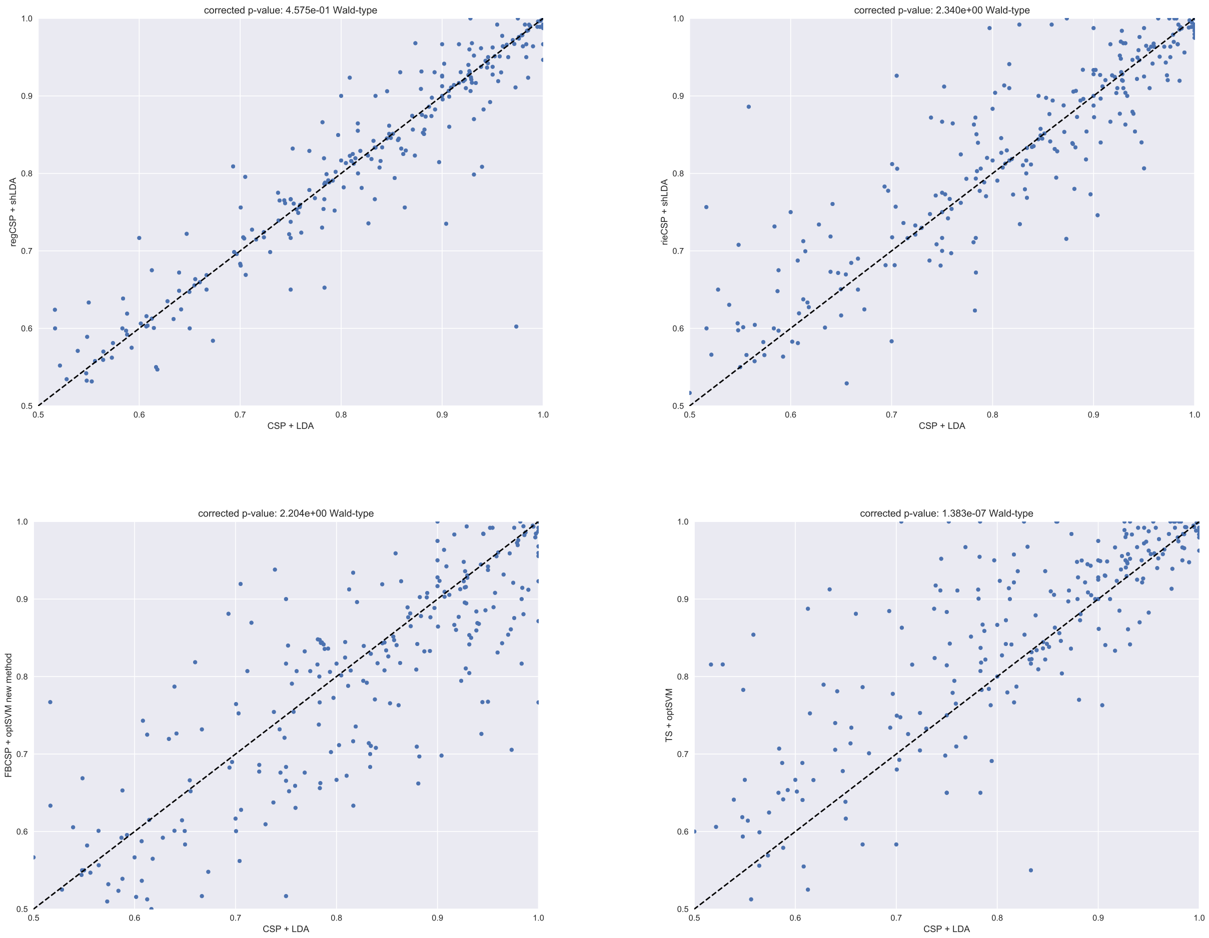

Fig. 4: Paired pots of CSP versus the two regularized methods, FBCSP, and the tangent space SVM. Interestingly, across datasets neither of the regularization methods does significantly better in within-session classification. However, the tangent space method does reliably out-perform it

Acknowledgements

We would like to extend our thanks to Professor Marco Congedo for his valuable input regarding the appropriate statistical procedure for this analysis.

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

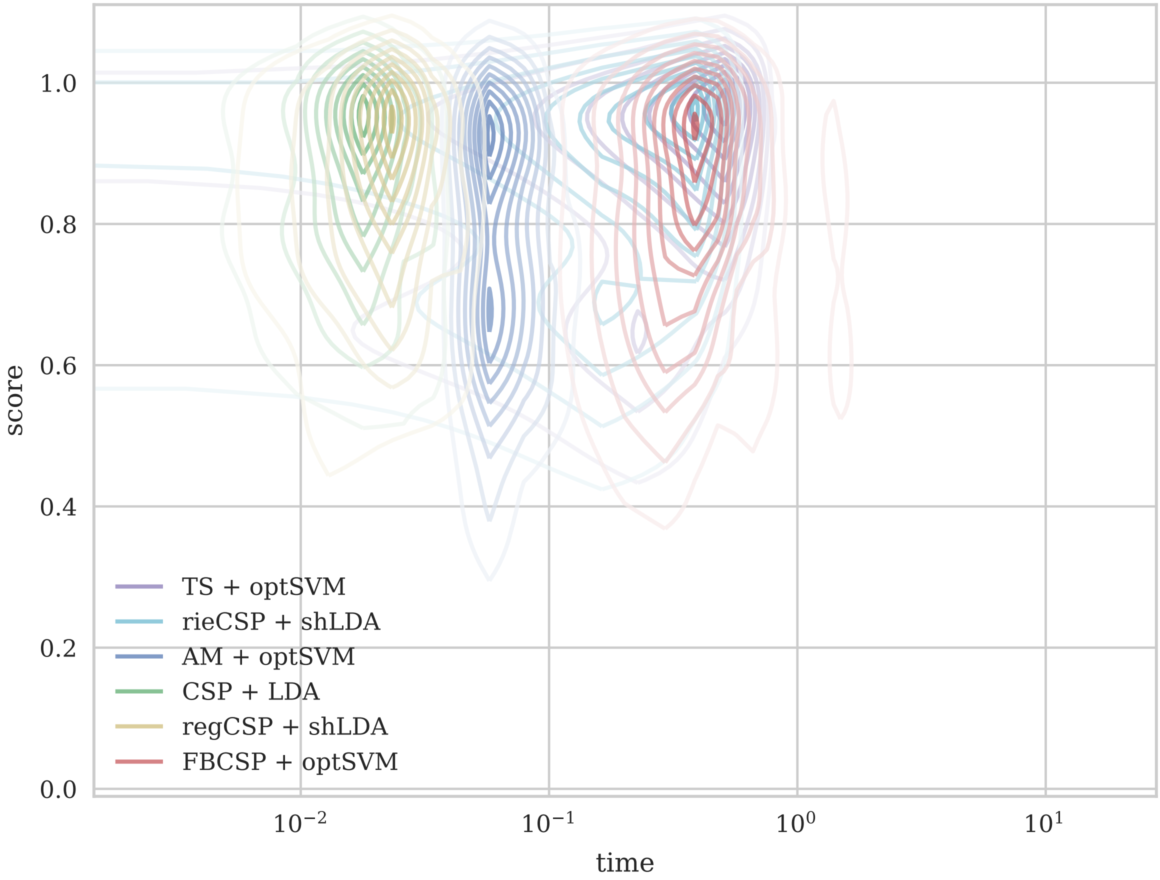

Fig. 5: Plot of the distribution of score versus computation time over all datasets, for all pipelines.