Minimizing Calibration Time for Brain Reading

Anonymous DAGM submission

Paper ID 006

Abstract. Machine learning is increasingly used to autonomously adapt brain-machine interfaces to user-specific brain patterns. In order to minimize the preparation time of the system, it is highly desirable to reduce the length of the calibration procedure, during which training data is acquired from the user, to a minimum. One recently proposed approach is to reuse models that have been trained in historic usage sessions of the same or other users by utilizing an ensemble-based approach. In this work, we propose two extensions of this approach which are based on the idea to combine predictions made by the historic ensemble with session-specific predictions that become available once a small amount of training data has been collected. These extensions are particularly useful for *Brain Reading Interfaces* (BRIs), a specific kind of brain-machine interfaces. BRIs do not require that user feedback is given and thus, additional training data may be acquired concurrently to the usage session. Accordingly, BRIs should initially perform well when only a small amount of training data acquired in a short calibration procedure is available and allow an increased performance when more training data becomes available during the usage session. An empirical offline-study in a testbed for the use of BRIs to support robotic telemanipulation shows that the proposed extensions allow to achieve this kind of behavior.

1 Introduction

Brain Reading Interfaces (BRIs) are one particular kind of brain-machine interface (BMI) that allow to provide the machine with information about the current mental state and intent of its user such that the machine can optimize its behavior accordingly. In contrast to active Brain-Computer Interfaces (BCIs, see [3,13] for a review of works), BRIs estimate the user's mental state and intent based on passive, external observation of brain activity without requiring any active participation of the user. This observation can, e.g., be based on electroencephalography (EEG). Since no active participation of the user is required, BRIs are well-suited for scenarios like robotic telemanipulation where a sophisticated BMI is expedient but the user needs to be fully immersed in his task.

Like active BCIs, BRIs must be adapted to the current brain patterns of the user since these characteristic patterns vary between different subjects and even change over time within the same subject. This can be achieved by using machine learning (ML) techniques (see, e.g., Blankertz et al. [4] for an example in an active BCI). The common approach for using ML in BCIs is to record labeled training data during a so-called calibration procedure that must be conducted prior to

each usage session. In this calibration procedure, the user acts in a controlled and supervised scenario. The labeled data acquired is then used to adapt the ML-based BCI system to the user's current brain patterns. The drawback of this approach is that the user has to conduct this calibration procedure each time he wants to use the system. Thus, it is highly desirable to keep this calibration procedure as short as possible (or remove its necessity altogether).

Different approaches for reducing the calibration time have been proposed: Krauledat et al. [9] proposed an algorithm targeted at long-term BCI users that allows to skip the calibration procedure. This is accomplished by inferring spatial filters and classifiers that generalize well across sessions based on reusing training data from historic sessions of the same user and clustering of historic spatial filters. Fazli et al. [6] proposed a method that allows to skip the calibration procedure for both long-term and novel users. Their approach is based on an ensemble of historic spatial-filter/classifier combinations that are transferred to the current session and whose individual predictions are combined into a joint prediction by means of a gating function. Both approaches require that a large number of historic sessions be available. Further approaches for reducing calibration time are multi-task learning [2], semi-supervised learning [10], and a hybrid approach that mixes historic data with session-specific data [11].

The main contribution of this paper is to propose two extensions of the "pure" ensemble-based approach of Fazli et al. and to present an empirical comparison of these approaches in a testbed for the use of BRIs to support robotic telemanipulation. The two extensions we propose—Classification Augmentation and Feature Augmentation—are based on the idea of combining the predictions made by the historic ensemble with session-specific predictions that become available once some amount of training data has been collected. We show that these extensions achieve good performance when only a small amount of training data is available and—in contrast to the "pure" ensemble approach—also become increasingly better for more training data. This is particularly important for BRIs, since BRIs allow to interweave the acquisition of training data with the actual usage session. Thus, the system should initially perform well based on a small amount of training data acquired in a short calibration procedure but should also be able to improve performance when increasingly more training data is gathered during the usage session. Furthermore, in contrast to related approaches like [6] and [9], the proposed extensions perform well also when only a small number of historic sessions is available. The paper is structured as follows: In Section 2, a testbed for BRIs in robotic telemanipulation is presented. Subsequently, the baseline BRI as well as different ensemble-based extensions are proposed in Section 3. In Section 4, the experimental setup and a discussion of our results are given and a conclusion is drawn in Section 5.

2 Scenario

Labyrinth Oddball The empirical evaluation was conducted on an EEG dataset recorded in the Labyrinth Oddball scenario (see Figure 1), a testbed for the

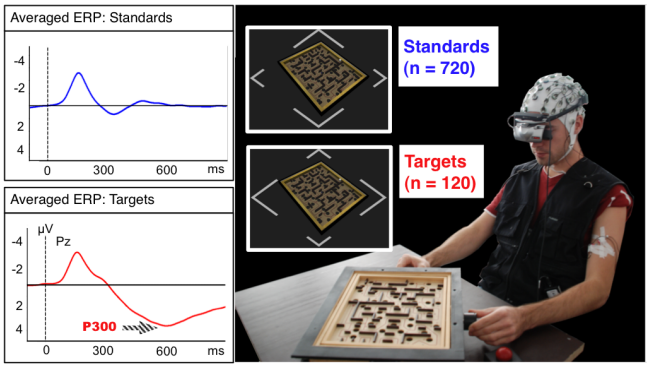

Fig. 1. Labyrinth Oddball: The subject plays a physical simulation of the BRIO® labyrinth and has to respond to rare 'target' stimuli by pressing a buzzer. Event-related potentials (ERPs) evoked by 'target' and more frequent 'standard' stimuli are depicted.

use of BRIs in robotic telemanipulation. In this testbed, the operator has to simultaneously execute a manipulation task (playing the Labyrinth game) and to distinguish two different kinds of stimuli presented to him while playing the game. The BRI only needs to passively monitor whether the operator of the Labyrinth game correctly recognized and distinguished these stimuli. Since no user feedback is given, the testbed is well suited for evaluation of BRIs (for more details we refer to [1] and the video in the supplementary material). The task for the BRI is to discriminate between the EEG patterns evoked by recognizing so-called 'standard' and 'target' stimuli[1]. While 'standard' stimuli are frequent (720 presentations per run) but irrelevant for the user, 'target' stimuli are rare (120 presentations per run) and require him to press a buzzer. Such a scenario is called "oddball discrimination paradigm" and the successful recognition of the rare 'target' stimuli is known to elicit an event-related potential (ERP) called P300 [12]. In contrast to many active BCIs (e.g., [13]), the classification has to be made based on the individual instance and not on an average over several repetitions of the same condition. To avoid differences in early visual brain activity and to make sure that differences in the EEG recorded and classified after the presentation of both stimuli types are actually due to higher cognitive processing, the visual presentation (shape and color) of standard and target stimuli was kept very similar. Note that neither during the calibration procedure nor during evaluation runs feedback was given to the subject.

Data Acquisition EEG data was acquired in 12 sessions from 6 male subjects; each subject performed 2 sessions. Sessions were recorded on different days; accordingly, the EEG cap was fitted onto the subject's head for each session

[1] This is a kind of proxy-task for the actual task of distinguishing between recognized and missed target stimuli (see [1] for a discussion).

anew. Each of these sessions consisted of five repetitions (called "runs") of the Labyrinth Oddball paradigm. After each of the five runs there was a short break of 10 minutes. The EEG was recorded and stored along with information about which stimulus was presented at what time and whether the buzzer was pressed afterwards. EEG was recorded continuously from 64 electrodes (extended 10–20 system with reference at electrode FCz), using an actiCap system (Brain Products GmbH, Munich, Germany). Two of the 64 channels (replacing the electrodes TP7 and TP8) were used to record electromyography signals of muscles of the lower arm and have been discarded in this study. EEG signals were amplified by two 32 channel BrainAmp DC amplifiers (Brain Products GmbH, Munich, Germany) and were sampled at 1000 Hz. The impedance was kept below 5 kΩ.

3 Methods

Baseline BRI As a first step of the baseline BRI system used for discrimination of the 'standard' and the 'target' condition, rectangular time windows starting 0 ms and ending 1000 ms after stimulus presentation are extracted from the continuous signal recorded during the experiment. Thereupon, the extracted time windows are normalized so that the mean value of each channel becomes 0 within this window. Subsequently, the signal is low-pass filtered (cutoff frequency 12 Hz), downsampled from 1000 Hz to 25 Hz, and again low-pass filtered for a cutoff frequency of 4 Hz in order to focus on slow ERPs like the P300.

After this, the signal is spatially filtered. Spatial filtering denotes a mapping of the original n channels $x(t)$ (that correspond one-to-one to the n electrodes) onto new pseudo-channels $\tilde{x}(t) = W^T x(t)$ that are a (linear) mixture of the signals recorded at different electrodes (see Blankertz et al. [5] for a discussion of why spatial filtering is an important step). In this work, we have generated spatial filters based on the common spatial patterns (CSP) algorithm [8]. CSP maps the data onto axes such that the variance for instances of the first class is maximized and the variance for the second class is minimized (or vice versa). This is achieved by a simultaneous diagonalization of the two empirical intra-class covariance matrices $\Sigma_1 = n_1^{-1} \sum_{i=1}^{n_1} (x_i^{(1)} - \mu^{(1)})(x_i^{(1)} - \mu^{(1)})^T$ and $\Sigma_2 = n_2^{-1} \sum_{i=1}^{n_2} (x_i^{(2)} - \mu^{(2)})(x_i^{(2)} - \mu^{(2)})^T$, i.e., by solving $\Sigma_1 W = \Lambda \Sigma_2 W$ where Λ is the vector of generalized eigenvalues and W is the matrix of generalized eigenvectors corresponding to the learned filters.

The values of the resulting pseudo-channels, i.e., the 26×62 samples of the 62 pseudo-channels that fall into the time window from 0 to 1000 ms, are used as features. Thereupon, each feature dimension is normalized such that its 2.5th percentile on the training data is mapped onto 0 and the 97.5th percentile is mapped onto 1. The resulting feature vectors are classified using a support vector machine (SVM) with linear kernel and complexity 0.01. Since the ratio of standard and target class instances in the dataset is highly unbalanced due to the oddball paradigm, the weight for class 'target' has been set to 2.0, while the weight of class 'standard' was set to 1.0. The feature set and all mentioned parameters have been chosen based on a preliminary investigation conducted on

a hold-out dataset. The implementation of the data processing system is based on the "Modular toolkit for Data Processing" [14].

Ensemble approach The baseline BRI outlined above adapts to the specific user by supervised training of subject- (and session)-specific spatial filters, feature normalization, and classifiers. Once trained, these three components form a subject- and session-specific classification system c_s (subsequently called a *classification flow*) that maps preprocessed time series x onto the scalar classifier prediction $c_s(x) \in \mathbb{R}$. Unfortunately, training of a classification flow requires a large training dataset that needs to be recorded at the start of each session. In order to reduce the required amount of training data (possibly even to zero), Fazli et al. [6] proposed to reuse classification flows trained on N historic sessions from the same and other subjects; such a set $h = (c_{h_1}, \ldots, c_{h_N})$ of historical classification flows c_{h_i} is called an *ensemble*. An ensemble can be used to generate a vector of class predictions $h(x) = (c_{h_1}(x), \ldots, c_{h_N}(x)) \in \mathbb{R}^N$ for a given time series x.

Thereupon, a so-called *gating function* g combines the ensemble's predictions $h(x) \in \mathbb{R}^N$ into a joint prediction $g(h(x)) \in \mathbb{R}$ (in the linear case $g(x) = \sum_{i=1}^{N} w_i c_{h_i}(x)$). A gating function can be defined without requiring session-specific training data by, e.g., training it on historic data (compare Fazli et al. [6]) or, alternatively, without any training by predicting according to the equally-weighted mean of the ensemble's predictions ($w_i = 1/N$). Furthermore, in situations where a small amount of session-specific training data is available, it is possible to train a gating function such that higher weights w_i are assigned to historic flows c_{h_i} that have high predictive performance for the current session. In this paper, we focus on the latter approach since it can be combined naturally with the proposed augmentation approaches (see below). We use an SVM with linear kernel for learning the gating function's parameters w_i since this SVM-based gating function achieved superior performance on hold-out test data of the given scenario compared to other common methods for learning gating functions. The outlined "pure" ensemble approach is depicted as the middle layer in Figure 2.

Augmentation approaches While ensemble approaches have been successful in achieving good performance when only a limited amount (or even no) training data from the current session is available (see, e.g., [6]), it is unlikely that they can achieve competitive results when more session-specific training data becomes available since they can not exploit novel patterns or shifts present in the current session that have not been observed in any of the historic sessions. We propose to use the ensemble approach presented above not instead but in addition to the training of a session-specific flow c_s, i.e., to *augment* the session-specific flow c_s by the predictions of the ensemble h. In this approach, the available training data is used for two purposes: training of a session-specific flow c_s and training of the gating function g which determines the final classification based on the ensemble's predictions and the session-specific information. We propose and

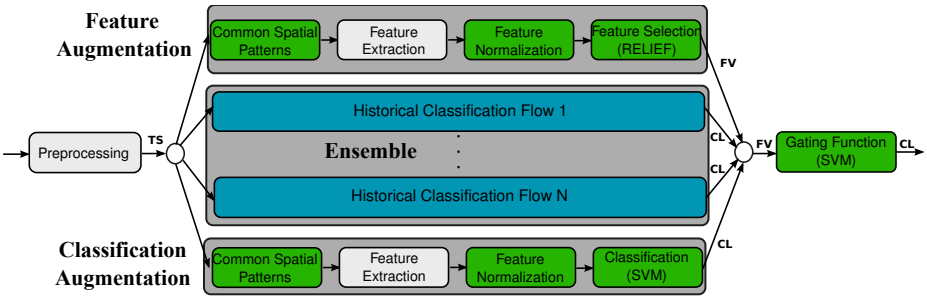

Fig. 2. Different ensemble and augmentation approaches. Feature Augmentation and Classification Augmentation are two alternative approaches for augmenting the ensemble's predictions by session-specific information. TS denotes a time-series, FV a feature vector, and CL a scalar classifier prediction.

compare two alternative approaches: *Classification Augmentation* and *Feature Augmentation* (see Figure 2).

In the classification augmentation approach, the prediction of the session-specific classification flow $c_s(x)$ is treated like any of the ensemble flow's predictions $c_{h_i}(x)$: An augmented ensemble $\tilde{h} = (c_{h_1}, \ldots, c_{h_N}, c_s)$ is generated and the gating function g chooses the joint prediction $g(\tilde{h}(x))$ based on $\tilde{h}$'s output ($\tilde{h}(x) \in \mathbb{R}^{N+1}$). Both c_s and g need to be trained based on data acquired in the current session; using the same data for both tasks, however, would result in a too strong reliance of the gating function on c_s since the predictive performance of c_s would be evaluated on its own training data. Thus, the available training data needs to be split into two parts. Empirically, we have found that using $2/3$ for training of c_s and $1/3$ for training of g is a good compromise.

In contrast, in the feature augmentation approach, the session-specific information added to the ensemble's predictions is not the classifier's prediction $c_s(x)$ but the values of the n most informative features $f_1(x), \ldots, f_n(x)$, i.e., $\tilde{h}(x) = (c_{h_1}(x), \ldots, c_{h_N}(x), f_1(x), \ldots, f_n(x)) \in \mathbb{R}^{N+n}$. Thus, $\tilde{h}(x)$ consists of two very different kinds of values: classifier predictions and CSP-pseudo-channel values (the selected features). However, this does not impose a problem and has the advantage that the available training data can be used more efficiently than in classification augmentation (note that while in principle feature selection and training of the gating function should be done on disjoint training sets, we have found empirically that it is favorable to train both on the same data). The choice of n is one additional parameter of this approach. The determination of the most informative features is made using the RELIEF feature selection algorithm [7].

4 Evaluation

Experimental Setup One historic classification flow has been trained for each historic session, resulting in 12 historic classification flows. Each of the 12 sessions

has been used once as evaluation session with the remaining 11 sessions being considered accordingly as historic sessions. Two different settings have been compared: In the "LeaveOneSessionOut" setting, the classification flows belonging to all but the current evaluation session have been used in the ensemble (resulting in ensembles of $N = 11$ flows), while in the "LeaveOneSubjectOut" setting, all classification flows that have not been generated from usage sessions of the current subject are used in the ensemble (resulting in ensembles of $N = 10$ flows). For each evaluation session, the data recorded in the first run has been used as training data and each of the remaining four runs has been used once as test dataset (intra-session setup), resulting in $4 * 12 = 48$ performance samples per method. Training datasets of six different sizes $t \in \{42, 84, 168, 252, 420, 840\}$ have been randomly sampled from the 840 labeled instances of the first run, where $t = 840$ corresponds to a calibration time of approximately 16 minutes. We refer to "experimental_design.pdf" in the supplementary material for more details.

Parameters of the SVM gating function have been selected using 5-fold internal cross-validation on the training data (complexity $C \in \{0.001, 0.01, 0.1, 1.0\}$ and target class weight $w_t \in \{1, 2, 5, 10\}$ for standard class weight 1). The parameter n of the feature-augmentation approach has been linearly increased from $n = 2$ for $t = 42$ to $n = 50$ for $t = 840$ to account for a stronger influence of the session-specific information when more training data becomes available. The scalar output of the gating function g is mapped onto the binary classes by choosing a threshold that maximizes the performance on the training data. For comparison, the results of the "zero-training" gating function that predicts according to the equally-weighted ensemble mean are given for the pure ensemble for $t = 0$. The performance of the session-specific flow c_s is given as "baseline". No value for classification augmentation is given for $t = 42$ since not enough target class training examples were available for the two-stage training procedure.

Because of the large class-skew of the classification task, standard measures such as accuracy are not well suited as performance metric. Instead, performance is measured according to the *mutual information* metric $I(T;Y) = H(T) - H(T|Y)$ with $H(T) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i)$ being the Shannon entropy of the class label T and $H(T|Y)$ the conditional entropy of the class label T given the classifier's prediction Y. The values of the metric correspond to the bits of information about the true class label conveyed by the classifier. The main advantage of this metric is that any kind of random classifier has mutual information 0. Note that the class label's entropy (and thus $I(T;Y)$) is upper bounded by $H(T) \approx 0.533$ for the given class ratio of 6 : 1. The optimally achieved performance (mutual information of 0.22) corresponds roughly to 94% correct classifications.

Results and Discussion We compared the four different approaches (factor e) for different training set sizes (factor t) by repeated measures ANOVA with t and e as within-subjects factors. This statistic model was separately performed for each setting $s \in$ {"LeaveOneSessionOut", "LeaveOneSubjectOut"} because of the different ensemble sizes N for the two settings. Whenever the results of

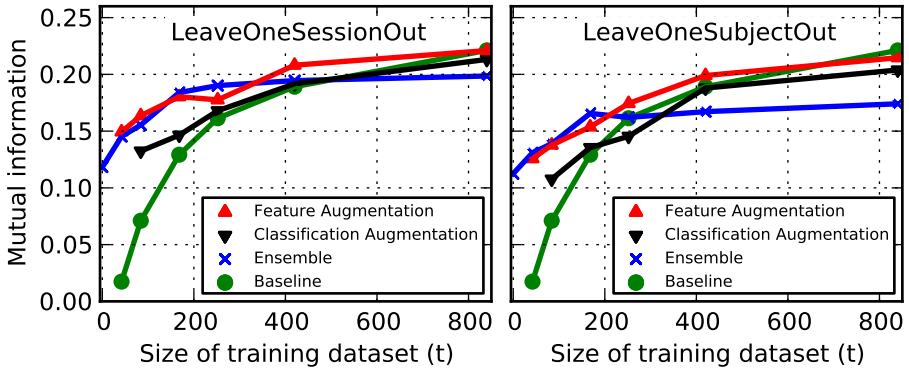

Fig. 3. Effect of training set size. Comparison of baseline, ensemble, and augmentation approaches for maximal N (LeaveOneSessionOut: $N = 11$, LeaveOneSubjectOut: $N = 10$) and for different training set sizes t.

the two different settings were compared, the additional factor s was added to the statistic model. Furthermore, in order to avoid that the different values of N for the two settings s affect these comparisons, one randomly selected session of another subject was removed from the "LeaveOneSessionOut" setting such that $N = 10$ in both cases. If needed, the Greenhouse-Geisser correction was applied. For pairwise comparisons, Bonferroni correction was applied. All tests have been performed for a significance level of $p < 0.05$ (see "statistics.pdf" in supplementary material for more detailed results).

Figure 3 summarizes the results of the study. In the "LeaveOneSessionOut" setting, the ensemble approach is significantly better than the baseline for $t \leq 252$ and worse for $t = 840$. This supports the hypothesis that historic predictors provide good performance when only a small amount of training data is available but are outperformed by session-specific predictors when larger amounts of training data have been acquired. Among the two augmentation approaches, feature augmentation is clearly better with statistical significance for $t \in \{42, 84, 168, 420\}$. This may be attributed to the inefficient usage of training data in the classification augmentation approach where it is necessary to split the training data into two disjoint parts (see Section 3). Furthermore, feature augmentation can be considered to be superior to both the ensemble and the baseline approach since performance is never significantly worse than any of the two, but significantly better than the ensemble for $t \geq 420$ and better than the baseline for $t \in \{42, 84, 168, 420\}$. This indicates that feature augmentation provides an efficient way of combining historic and session-specific information by adaptively learning which source of information should be trusted more.

Results in the "LeaveOneSubjectOut" setting are qualitatively similar, with the notable difference that the ensemble's performance is significantly worse than in the "LeaveOneSessionOut" setting for all t. This shows that a historic session

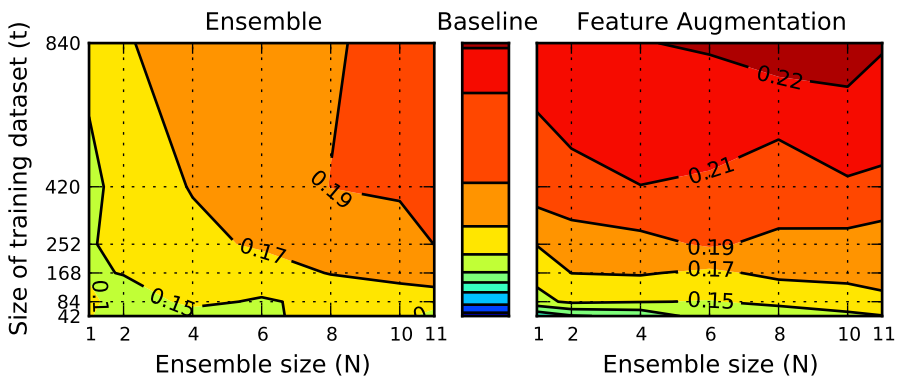

Fig. 4. Effect of the ensemble size. Mutual influence of ensemble size N and the training set size t onto performance (mutual information) in the LeaveOneSessionOut setting. For comparison, the baseline performance is shown for the same values of t.

of the same user helps to increase the performance of the ensemble approach. As a result, in the "LeaveOneSubjectOut" setting, the ensemble is significantly better than the baseline only for $t \leq 168$ but worse for $t = 840$. Performance of the feature augmentation approach deteriorates significantly as well in the "LeaveOneSubjectOut" setting for all $t \neq 252$; however, this deterioration is less strong since the session-specific flow compensates partly for the missing historic session of the same user. Accordingly, the feature augmentation approach is still never significantly worse than the baseline but significantly better for $t \leq 168$.

Figure 4 shows how the size N ($N \in \{1, 2, 4, 6, 8, 10, 11\}$) of the historic ensemble and the size of the training dataset t mutually affect the performance of the pure ensemble and the feature augmentation approach (in the "LeaveOneSessionOut" setting). These results have been separately analyzed for each setting by repeated measures ANOVA with the within-subjects factors N, t, and e. The performance of the pure ensemble approach depends strongly on the ensemble's size: Even for large t, no performance above 0.17 is achieved for $N \leq 2$ and no performance above 0.19 for $N \leq 6$. This dependence on N is even stronger in the "LeaveOneSubjectOut" setting (see "ensemble_size_LOSubjectO.pdf" in supplementary material). On the other hand, the feature augmentation approach depends less strongly on N, outperforming the baseline for small t significantly even when N is very small ($t < 84$ for $N = 1; t < 168$ for $N \in \{2, 4\}$) but never being significantly worse than the baseline.

5 Conclusion

We have presented two alternative approaches for combining predictions made by an ensemble trained on historic sessions with a flow that has been trained on data acquired in the current usage session. This hybrid approach allows to achieve a

better performance than the session-specific predictor when only small amounts of training data are available and a better performance than the historic ensemble when more training data becomes available. The proposed approach performs well for subjects for which historic sessions exist but also for novel subjects for which no historic sessions have been conducted. Furthermore, in contrast to related approaches like [6] and [9], the proposed method also achieves good performance when only a small number of historic sessions is available, where it still outperforms the session-specific predictor for small training datasets. Future work is to conduct online studies in which the acquisition of training data is performed concurrently to the usage session.

References

1. Supplementary Material I. (supplementary_material_I.pdf)
2. Alamgir, M., Grosse-Wentrup, M., Altun, Y.: Multi-task learning for Brain-Computer Interfaces. In: Proceedings of the 13th International Conference on Artificial Intelligence and Statistics. vol. 9 of JMLR: W&CP 9 (2010)
3. Birbaumer, N.: Breaking the silence: Brain-Computer Interfaces (BCI) for communication and motor control. Psychophysiology 43(6), 517–532 (Nov 2006)
4. Blankertz, B., Dornhege, G., Lemm, S., Krauledat, M., Curio, G., Müller, K.R.: The Berlin Brain-Computer Interface: Machine learning based detection of user specific brain states. Journal of Universal Computer Sciene 12(6), 581–607 (2006)
5. Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., Müller, K.R.: Optimizing spatial filters for robust EEG Single-Trial analysis. Signal Processing Magazine, IEEE 25(1), 41–56 (2008)
6. Fazli, S., Popescu, F., Danóczy, M., Blankertz, B., Müller, K., Grozea, C.: Subject-independent mental state classification in single trials. Neural Networks 22(9), 1305–1312 (Nov 2009)
7. Kira, K., Rendell, L.A.: The feature selection problem: Traditional methods and a new algorithm. In: AAAI. pp. 129–134 (1992)
8. Koles, Z.J.: The quantitative extraction and topographic mapping of the abnormal components in the clinical EEG. Electroencephalography and Clinical Neurophysiology 79, 440–447 (1991)
9. Krauledat, M., Tangermann, M., Blankertz, B., Müller, K.: Towards zero training for Brain-Computer interfacing. PLoS ONE 3(8), e2967 (2008)
10. Li, Y., Guan, C., Li, H., Chin, Z.: A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system. Pattern Recognition Letters 29(9), 1285–1294 (Jul 2008)
11. Lotte, F., Guan, C.: Learning from other subjects helps reducing Brain-Computer interface calibration time. In: International Conference on Audio Speech and Signal Processing (ICASSP) (2010)
12. Squires, N.K., Squires, K.C., Hillyard, S.A.: Two varieties of long-latency positive waves evoked by unpredictable auditory stimuli. Electroencephalography and Clinical Neurophysiology 38(4), 387–401 (April 1975)
13. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-computer interfaces for communication and control. Clinical Neurophysiology 113(6), 767–791 (Jun 2002)
14. Zito, T., Wilbert, N., Wiskott, L., Berkes, P.: Modular toolkit for data processing (MDP): a python data processing framework. Front. Neuroinform. 2, 8 (2008)