Adaptive SVM-Based Classification Increases Performance of a MEG-Based Brain-Computer Interface (BCI)

Martin Spüler[1], Wolfgang Rosenstiel[1], and Martin Bogdan[1,2]

[1] Wilhelm-Schickard-Institute for Computer Science, University of Tübingen, Germany
[2] Computer Engineering, University of Leipzig, Germany

Abstract. One problem in current Brain-Computer Interfaces (BCIs) is non-stationarity of the underlying signals. This causes deteriorating performance throughout a session and difficulties to transfer a classifier from one session to another, which results in the need of collecting training data every session. Using an adaptive classifier is one solution to keep the performance stable and reduce the amount of training that is needed for a good BCI performance. In this paper we present an approach for an adaptive classifier based on a Support Vector Machine (SVM). We evaluate its advantage on offline BCI data and show its benefits and online feasibility in an online experiment using a MEG-based BCI with 10 subjects.

Keywords: Brain-Computer interface (BCI), adaptive control, Support Vector Machine (SVM).

1 Introduction

The goal of a Brain-Computer Interface (BCI) is to enable the user to communicate or control a computer by means of pure brain activity. While in most non-invasive BCIs electro-encephalography (EEG) is the method of choice, magneto-encephalography (MEG) is another interesting, but rarely used technique for measuring and recording brain activity. It provides a higher spatiotemporal resolution and has been shown to work well for BCI[1].

Nevertheless it suffers from the same problem as EEG-based BCIs, namely the non-stationarity of the recorded signal, which introduces a covariate shift of the data. Reasons for non-stationarity may be caused by the transfer from training without feedback to online usage with feedback, head movements in the MEG, which cause the generating brain areas to be under a different sensor, or changes in the mental state over time. These non-stationaries especially are a problem, when a classifier trained on data of a previous session is used for classification, which is often referred to as the session-transfer problem.

Adaptive classifiers may be one possibility to alleviate non-stationaries[2]. It has been shown that linear discriminant analysis(LDA) classifiers can be adapted

A.E.P. Villa et al. (Eds.): ICANN 2012, Part I, LNCS 7552, pp. 669–676, 2012.

unsupervised and that adaption may improve BCI performance[3,4,5]. But, disregarding this paper, Vidaurre et al. are the only ones who showed that an unsupervised classifier adaption can be used online in a BCI[6].

While previous works use LDA for classification, we focus on SVM-based classifiers in this paper. By incorporating the new continuously available data, the classifier can be tuned to reflect changing data distributions. This should result in increased performance and reduce the need to collect more training data.

But a high accuracy is not the only requirement for a classifier that is used in a BCI. For online feedback the classifier has to be efficient, since the adaption of the classifier must not slow down the BCI or interfere with the response time of the BCI. This is especially challenging when recording MEG, since the amount of data collected with MEG is much larger than with EEG.

In the following we will present an algorithm that adapts itself to changing data and is efficient enough to be used online with high-dimensional MEG data. For the first time, we show in an online experiment that unsupervised adaption of a SVM increases BCI performance.

2 Method

2.1 Adaptive BCI Classifier

For unsupervised adaption of the SVM[7] classifier we used a method that is similar to co-training[8], but can be used online. Assume our initial training data D consists of N samples $\{x_1, ..., x_N\}$ with $x_i \in \mathbb{R}^n$ and given labels $[y_1, ..., y_N]$ with $y_i \in \{-1, 1\}$. D is used to train an initial SVM classifier by

$$\begin{aligned} &\min_{\mathbf{w},\xi,b} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N} \xi_i \\ &\text{subject to} \quad y_i(\langle \mathbf{w}, \mathbf{x_i} \rangle - b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, ..., N \end{aligned} \tag{1}$$

The sign of the decision function $f(x_i) = \langle \mathbf{w}, \mathbf{x}_i \rangle + b$ is used to determine the class label.

If a new sample x_j is available, it is classified by $z_j = sign(f(x_j))$ and the posterior class probability $p_j = Pr(y_j = z_j|x_j)$ is calculated. Platt's probabilistic output[9] is used to estimate p_j by computing the parameters (A, B) for a sigmoid function that approximates the posterior class probabilities

$$Pr(y_i = 1|x_i) \approx P_{A,B}(f) \equiv \frac{1}{1 + exp(Af + B)}, \quad \text{where} f = f(x) \tag{2}$$

Since the true class label is unknown only the predicted class label z_j can be used for incorporating x_j and adapt the classifier. As a wrong prediction of z_j could lead to an inferior classifier when used for adaption, p_j is taken into account to ensure a high probability of z_j being correct. A threshold p_t is defined and if $p_j > p_t$ holds, x_j is added to the training data set D with $y_j = z_j$.

Since D has changed, (1) needs to be solved again. Although this could be done by using SMO[10], it wouldn't be feasible when used online, because the solution for (1) would be computed from scratch each time D changes. Therefore the method described by Diehl and Cauwenberghs[11] is used to train a SVM and incrementally update the solution every time a new sample is added. All code was written in Matlab[1].

The choice of a proper p_t is crucial. Choosing a smaller p_t can lead to suboptimal results due to bad class information, while a higher p_t only allows very few samples to be used for adaption. In the following $p_t = 0.8$ is used, which worked as a good tradeoff. This value has been chosen empirically based on the data described in the next section. Detailed results from the parameter selection are also presented later in this paper.

2.2 Offline Analysis

Data. To evaluate the possible benefit of the adaptive classifier, an offline analysis was performed on previously recorded data[12] from 10 subjects. The data consisted of 2 classes: a subtraction task and a motor imagery task.

Each subject participated in 2 sessions, with each session consisting of 51 trials per task. Recording was done with a 275-channel whole-head MEG-system (VSM MedTech Ltd.) at a sampling rate of 586 Hz. Each trial lasted 4.05 s with about 6 s of break between the trials. Instructions were given on a screen and a fixation cross was displayed during trials to minimize eye movement. This was only a recording of data and no BCI feedback was given to the subjects.

Signal Processing and Classification. The signals were filtered and resampled to 200 Hz. For spatial filtering a small laplacian derivation was applied. To reduce the number of channels only the 185 inner channels were used, which should also reduce the influence of possible artefacts, which are most prominent on the outer channels. After the preprocessing, the power spectrum was estimated by an autoregressive model of order 16 computed with the Burg Method and the frequency range from 1 to 40 Hz with a bin width of 2 Hz was used. The logarithm function was applied to each value. Before classification r^2-ranking[13] was used for feature selection. We did not estimate the optimal number of features by cross-validation, which introduced overfitting in previous experiments, but instead used a fixed number of 1000 features. Each feature was normalized to have zero mean and unit variance for the training dataset. The test dataset was scaled according to the mean and standard deviation of the training dataset.

For classification we always used the first session for training and the second session to test the classifier. This procedure was chosen to specifically address the problem of non-stationarities and the session-transfer problem.

To evaluate the proposed adaptive method, we compared it to other SVM-based approaches. For all methods the SVM parameters were kept constant with

[1] A small Matlab library for incremental SVM classification including multiclass-support and Platt's probabilistic output can be downloaded at `http://www-ti.informatik.uni-tuebingen.de/~spueler/mcpIncSVM/`

$C = 1$ and a linear kernel was used. We decided against a parameter estimation by gridsearch and cross-validation because it introduced overfitting in previous experiments.

As baseline classifier a non-adaptive SVM was used. Apart from the method proposed in section 2.1, a supervised adaption was performed: after classification of trial x_j the classifier was updated by adding x_j with the true class label y_j. To compare the presented adaptive algorithm with other adaptive SVM-based methods, the approaches by Tamura et al.[14] and Li et al.[15] were used. For the method by Tamura et al. different ζ and η were tested and the best parameters used.

2.3 Online Experiment

To confirm the results from the offline analysis (shown in section 3), an online experiment was designed with 10 subjects. The same two classes were used as in the data for the offline analysis: a motor imagery task and a subtraction task. To explicitly evaluate the proposed adaptive algorithm in context of the session-transfer problem, each subject should participate in two sessions. In the first session 200 trials training data were recorded and a non-adaptive classifier was tested afterwards with online feedback. In the second session the adaptive classifier (initially trained on the training data from the first session) was used for online feedback in 200 trials. Since two of the subjects did not achieve significant BCI control in the first session, they did not participate in the second session.

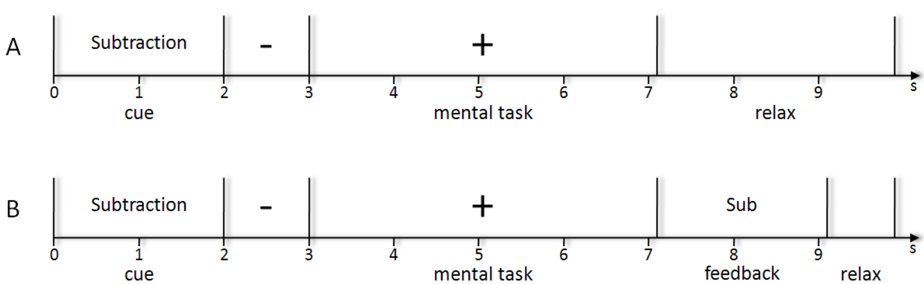

Fig. 1. Online paradigm for the MEG-based BCI. A) training phase without feedback. B) test phase with feedback.

A 275-channel whole-head MEG-system (VSM MedTech Ltd.) at a sampling rate of 586 Hz was used to record the data. During measurement the head position was continuously recorded. During the test phase feedback was given after every trial, which indicated the result of the classifier. The paradigm as well as the time intervals used in the online experiment can be seen in figure 1.

To evaluate the benefit of the adaptive classifier we simulated the online experiment by applying the non-adaptive baseline classifier (as previously described in 2.2) offline to the data.

3 Results

3.1 Parameter Selection

Results from the offline analysis, where the proposed adaptive method was tested with different thresholds p_t, are shown in figure 2. The highest accuracy was obtained for $p_t = 0.75$ with 90.3 %. Due to the steep drop of the accuracy when using a $p_t < 0.75$ or a $p_t \geq 0.85$, we decided to use $p_t = 0.8$.

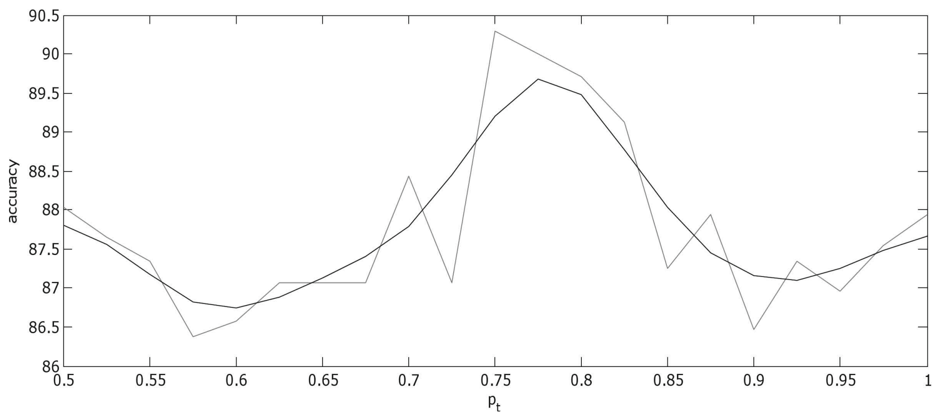

Fig. 2. Accuracy depending on the choice of the threshold p_t. The gray curve shows the actual values, while the black curve shows the smoothed accuracy values.

3.2 Offline Analysis

The results for the offline analysis are shown in table 1. With a mean accuracy of 89.7 % our adaptive classifier performed better than the non-adaptive baseline classifier, which achieved an average accuracy of 87.9 %. The method by Li et al. gave worse results (86.2 %) than the non-adaptive classifier, while the method by Tamura et al. gave exactly the same results as the non-adaptive classifier (87.9 %).

Interestingly the adaptive classifier even marginally exceeded the accuracy for a supervised adaption, that resulted in an average accuracy of 89.4 %.While for the approach by Tamura et al. different ζ and η were tested, $\zeta = 0.7, \eta = 0.1$ worked best for this data, which are the same parameters as used in [14].

3.3 Online Experiment

The results from the online experiment are shown in figure 2. While the online accuracy averaged over all 8 subjects was 84.6 %, the mean accuracy for the non-adaptive classifier was 80.5 %. There was no correlation between the number of days between the sessions and the benefit by the adaptive algorithm.

Table 1. Offline accuracies for different methods of adaptive classification. Best results are printed bold, ignoring the condition with supervised adaption.

subject	adaptive	Li et al.	Tamura et al.	non-adaptive	supervised
A01	**78.4 %**	77.5 %	76.5 %	76.5 %	71.6 %
A02	**95.1 %**	**95.1 %**	92.2 %	92.2 %	93.1 %
A03	**95.1 %**	94.1 %	**95.1 %**	**95.1 %**	97.1 %
A04	84.3 %	83.3 %	**90.2 %**	**90.2 %**	90.2 %
A05	**89.2 %**	88.2 %	83.3 %	83.3 %	93.1 %
A06	**87.3 %**	63.7 %	81.4 %	81.4 %	86.3 %
A07	97.1 %	**98.0 %**	**98.0 %**	**98.0 %**	97.1 %
A08	**98.0 %**	95.1 %	96.1 %	96.1 %	96.1 %
A09	**75.5 %**	73.5 %	74.5 %	74.5 %	71.6 %
A10	**97.1 %**	93.1 %	92.2 %	92.2 %	98.0 %
mean	**89.7 %**	86.2 %	87.9 %	87.9 %	89.4 %

Table 2. Accuracies for the presented adaptive SVM algorithm during the online experiment, simulated online accuracies with the non-adaptive baseline classifier and the number of days between session 1 and session 2

subject	online adaptive	non-adaptive	days between S1 and S2
B01	**92.5 %**	68.0 %	2
B02	**92.0 %**	90.5 %	1
B03	**95.5 %**	94.0 %	1
B04	84.0 %	**84.5 %**	1
B05	**97.5 %**	**97.5 %**	1
B06	**83.5 %**	82.5 %	6
B07	**59.5 %**	59.0 %	1
B08	**72.0 %**	68.0 %	1
mean	**84.6 %**	80.5 %	1.75

When pooling the results from the offline analysis and the online experiment, the proposed adaptive algorithm has a significantly better accuracy than the non-adaptive baseline algorithm ($p < 0.05$, paired t-test).

4 Discussion

The offline analysis has shown that the adaptive algorithm presented in this paper increases accuracy compared to the non-adaptive baseline algorithm and has better performance than the methods by Li et al. and Tamura et al.. In addition its average performance is better than a supervised classifier, which is stunning on first sight. The reason for this is unknown, but one explanation could be erroneously labeled data. If the subject performed the wrong mental task during a trial, the label would be wrong. Despite the wrong label, the adaptive classifier could detect the correct class and use the data with a correct label for training.

With the online experiment we have verified the results from the offline analysis, that the presented adaptive algorithm increases performance. We have also shown the presented algorithm to be efficient enough to be used online. Since we have shown it to work with high-dimensional MEG-data, it will also work with EEG or ECoG data.

We used some principles of co-training and expanded it to suit the special needs for online adaption of a BCI system. In addition we used incremental training of the SVM to avoid a complete retraining each iteration. This allows the adaptive classifier to handle high-dimensional data, as they occur when using MEG, fast enough to be used in an online environment.

Due to limited space in this paper and the importance of experiments with real BCI data, results from experiments with artifically generated data are not presented here. But it should be mentioned that this method copes well with highly non-stationary data (mean changes continously by more than 10 times the standard deviation), if old data is removed by decremental unlearning[11].

A potential improvement of the classifier could be made by additionally incorporating the information by error-related potentials (ErrPs), which can be measured if the user detects an erroneous response by the BCI system. It has been shown that ErrPs can be detected with sufficient accuracy during BCI usage[16] and be used to correct errors. If an ErrP is detected, the result of the classifier is wrong with a high probability and therefore should not be used for adaption.

5 Conclusion

In this paper we have proposed an adaptive SVM-based algorithm and have shown in an offline analysis of BCI data recorded with MEG, that it is superior to other adaptive and non-adaptive classifiers. To verify this results, we performed an online experiment with 8 subjects (10 intended) and proved the proposed adaptive algorithm to achieve higher accuracies than the non-adaptive baseline algorithm. We have shown unsupervised online adaption of a SVM classifier in a BCI for the first time and could significantly increase performance.

Acknowledgments. This study was partly granted by the German Federal Ministry of Education and Research (BMBF, Grant UTü 01GQ0831) and the *Deutsche Forschungsgemeinschaft* (DFG, Grant RO 1030/15-1, KOMEG).

References

1. Mellinger, J., Schalk, G., Braun, C., Preissl, H., Rosenstiel, W., Birbaumer, N., Kübler, A.: An MEG-based brain-computer interface (BCI). NeuroImage 36(3), 581–593 (2007)
2. Shenoy, P., Krauledat, M., Blankertz, B., Rao, R.P.N., Müller, K.-R.: Towards adaptive classification for BCI (2006)

3. Blumberg, J., Rickert, J., Waldert, S., Schulze-Bonhage, A., Aertsen, A., Mehring, C.: Adaptive classification for brain computer interfaces. In: 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS 2007, pp. 2536–2539 (August 2007)
4. Vidaurre, C., Schlögl, A., Blankertz, B., Kawanabe, M., Müller, K.: Unsupervised adaptation of the lda classifier for brain-computer interfaces. In: Proceedings of the 4th International Brain-Computer Interface Workshop and Training Course, pp. 122–127 (2008)
5. Liu, G., Zhang, D., Meng, J., Huang, G., Zhu, X.: Unsupervised adaptation of electroencephalogram signal processing based on fuzzy c-means algorithm. International Journal of Adaptive Control and Signal Processing (2011)
6. Vidaurre, C., Kawanabe, M., von Bünau, P., Blankertz, B., Müller, K.R.: Toward Unsupervised Adaptation of LDA for Brain-Computer Interfaces. IEEE Transactions on Biomedical Engineering 58(3), 587–597 (2011)
7. Vapnik, V.N.: Statistical Learning Theory, 1st edn. Wiley-Interscience (September 1998)
8. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, pp. 92–100. ACM, New York (1998)
9. Lin, H.-T., Lin, C.-J., Weng, R.: A note on Platt's probabilistic outputs for support vector machines. Machine Learning 68, 267–276 (2007)
10. Chen, P.H., Fan, R.E., Lin, C.J.: A study on SMO-type decomposition methods for support vector machines. IEEE Transactions on Neural Networks 17, 893–908 (2006)
11. Diehl, C.P., Cauwenberghs, G.: SVM Incremental Learning, Adaptation and Optimization. In: Proceedings of the 2003 International Joint Conference on Neural Networks, pp. 2685–2690 (2003)
12. Bensch, M., Mellinger, J., Bogdan, M., Rosenstiel, W.: A multiclass BCI using MEG. In: Proceedings of the 4th Int. Brain-Computer Interface Workshop, Graz, pp. 191–196 (September 2008)
13. Spüler, M., Rosenstiel, W., Bogdan, M.: A fast feature selection method for high-dimensional MEG BCI data. In: Proceedings of the 5th Int. Brain-Computer Interface Conference, Graz, pp. 24–27 (September 2011)
14. Tamura, H., Kawano, S., Tanno, K.: Unsupervised learning method for a support vector machine and its application to surface electromyogram recognition. Artificial Life and Robotics 14, 362–366 (2009), doi:10.1007/s10015-009-0682-1
15. 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 (2008)
16. Spüler, M., Bensch, M., Kleih, S., Rosenstiel, W., Bogdan, M., Kübler, A.: Online use of error-related potentials in healthy users and people with severe motor impairment increases performance of a P300-BCI. Clinical Neurophysiology 123(7), 1328–1337 (2012)