raxDAWN: Circumventing Overfitting of the Adaptive xDAWN

Mario Michael Krell[1], Hendrik Wöhrle[2] and Anett Seeland[2]
[1]*Robotics Research Group, University of Bremen, Robert-Hooke-Str. 1, Bremen, Germany*
[2]*Robotics Innovation Center, German Research Center for Artificial Intelligence GmbH, Bremen, Germany*
krell@uni-bremen.de, {hendrik.woehrle, anett.seeland}@dfki.de

Keywords: xDAWN, Spatial Filtering, Online Learning, Electroencephalogram, Event-Related Potential, Brain-Computer Interface

Abstract: The xDAWN algorithm is a well-established spatial filter which was developed to enhance the signal quality of brain-computer interfaces for the detection of event-related potentials. Recently, an adaptive version has been introduced. Here, we present an improved version that incorporates regularization to reduce the influence of noise and avoid overfitting. We show that regularization improves the performance significantly for up to 4%, when little data is available as it is the case when the brain-computer interface should be used without or with a very short prior calibration session.

1 INTRODUCTION

In brain-computer interfaces (Blankertz et al., 2011; Zander and Kothe, 2011; van Erp et al., 2012; Kirchner et al., 2013, BCIs), event-related potentials (ERPs) in the electroencephalogram (EEG) are quite often used to *deduce* informations from the human's internal brain state and translate the internal state to informations that are usable by other systems. Examples are P300 and error related potentials (Krusienski et al., 2006; Buttfield et al., 2006). In contrast to the common ERP analysis, many BCIs have to work on single-trial instead of averaged data. The single-trial analysis of EEG is very difficult due to the low signal-to-noise ratio. Here, spatial filtering is a common approach to enhance the signal-to-noise ratio in EEG data. Its concept is to linearly combine data from different sensors to a reduced set of so-called pseudo channels with reduced noise level. For ERP-based BCIs, usually only one pattern is relevant and has to be detected. One approach is to concatenate the ERP data samples and look for some periodic behaviour as done by the PiSF and its variants (Ghaderi and Kirchner, 2013). A different approach is adopted by the xDAWN algorithm (Rivet et al., 2009, further details in Section 2.1). It models the average pattern as a signal hidden by the noise and the (potential) overlay of ERPs due to short time distance. The objective of the filter then is to maximize the signal-to-signal-plus-noise ratio. Recently, the xDAWN algorithm has been enhanced by a new version which enables incremental training at run time (Wöhrle et al., 2015). This is important for BCIs because the calibration phase should be as short as possible and the patterns might change over time. Hence, an additional adaptation is required.

A different group of spatial filters are derived from the common spatial pattern algorithm (Blankertz et al., 2008, CSP). In contrast to the aforementioned algorithms, the objective of CSP filters is to enhance the data for two different classes and focuses on the frequency domain instead of the time domain. To add further properties to this spatial filter, several regularization methods have been suggested as extensions (Samek et al., 2012). Regularization methods have not yet been applied to the xDAWN and this paper is the first to introduce this method.

In Section 2, we introduce xDAWN and axDAWN and subsequently we show how to integrate Tikhonov regularization into the model similar to the approach for the CSP. In Section 3, the algorithm is evaluated on EEG data and it is shown that it can improve the performance, especially when training data is missing. Finally, we conclude in Section 4.

2 METHODS

This Section first briefly introduces xDAWN and its adaptive variant. Based on these descriptions afterwards, we propose the new regularized variant.

2.1 xDAWN

Let $X \in \mathbb{R}^{N_T \times N_S}$ be the matrix of recorded data, where N_T is the total number of single temporal samples and N_S is the number of sensors. So X_{ij} is the recording of the j-th sensor at the i-th time point. The ERP is the typical (averaged) electrophysiological response to a stimulus. This is modeled with the matrix $A \in \mathbb{R}^{N_E \times N_S}$ where N_E is the expected length of the ERP and it is usually chosen between 600 and 1000 milliseconds. To model the data based on A, an additional noise matrix $N \in \mathbb{R}^{N_T \times N_S}$ and a Toeplitz matrix $D \in \mathbb{R}^{N_T \times N_E}$ are required. For every time point, where an ERP pattern is expected to start, a 1 is added to D at the respective time index in the first column. For the other columns the entry is continued to have a diagonal of ones, as it is common for Toeplitz matrices. The summarizing formula of the xDAWN data model then reads

$$X = DA + N\,. \tag{1}$$

The first step is to obtain a least squares estimate of A:

$$\hat{A} = \arg\min_A \|X - DA\|^2 = (D^T D)^{-1} D^T X\,. \tag{2}$$

If there is no overlap of ERPs, $\hat{A}$ is equal to the averaged signal $(D^T X)$. The second step of the xDAWN modeling process is to define the objective of constructing a filter vector $\hat{u}$ which maximizes the signal-to-signal plus noise ratio with the generalized Rayleigh quotient[1]

$$\hat{u} = \arg\max_{u \in \mathbb{R}^{N_S}} \frac{u^T \hat{A}^T D^T D \hat{A} u}{u^T X^T X u}\,. \tag{3}$$

The third step to solve the optimization problem is a combination of QR decomposition and singular value decomposition applied to the matrices in the optimization problem. For further details, we refer to (Rivet et al., 2009). Note that the Generalized Eigenvalue Decomposition is a common approach to solve the Rayleigh quotient optimization problem. The result is a set of filters which is sorted by their "quality" where quality is measured by the absolute value of the eigenvalue.

[1]The original definition used a filter matrix U and traces for both parts of the coefficients but the original solution approach refers to the respective eigenvalue problem which would only be appropriate for our definition. For other dimensionality reduction algorithms, the definition is similar (e.g., Fisher's linear discriminant (Mika et al., 2001)).

2.2 axDAWN: Adaptive xDAWN

The axDAWN algorithm tackles the implementation part with a different approach. The main motivation of the axDAWN algorithm (Wöhrle et al., 2015) is that the xDAWN algorithm is not applicable for online learning due to its batch optimization (Wöhrle et al., 2013). It has high memory consumption and it cannot be implemented on a small device with limited resources. Note, that X and D would grow linearly over time.

With each incoming sample, axDAWN updates several matrices, which all have constant dimensions over time. Let t be the new time point and all relevant matrices be already calculated for $t-1$, and let $x(t)$ be a new data sample with the respective row $d(t)$ in D. If there is no overlap of ERPs, $\hat{A}(t)$ can be calculated directly as the running average. Otherwise, $(D^T X)(t) \in \mathbb{R}^{N_E \times N_S}$ is updated by

$$(D^T X)(t) = d(t)^T x(t) + (D^T X)(t-1)\,, \tag{4}$$

the new matrix

$$H(t) := (D^T D)^{-1}(t) \in \mathbb{R}^{N_S \times N_S} \tag{5}$$

is introduced, and the Sherman-Morrison-Woodbury formula (Golub and Van Loan, 1996) is used to update $H(t)$

$$H(t) = H(t-1) \frac{H(t-1) d(t) d^T(t) H(t-1)}{1 + d^T(t) H(t-1) d(t)}\,. \tag{6}$$

Combining both, we get

$$\hat{A}(t) = H(t) \cdot (D^T X)(t). \tag{7}$$

It furthermore holds

$$(D^T D)(t) = (D^T D)(t-1) + d(t)^T d(t) \text{ and} \tag{8}$$

$$R_2(t) := X(t)^T X(t) = R_2(t-1) + x(t)^T x(t) \in \mathbb{R}^{N_S \times N_S}\,. \tag{9}$$

Taking everything into consideration, the formulas can be used to calculate

$$R_1^1(t) := \hat{A}(t)^T (D^T D)(t) \hat{A}(t) \in \mathbb{R}^{N_E \times N_S}. \tag{10}$$

Note, that $R_2(t)$ and $R_1^1(t)$ are part of the original optimization problem

$$\arg\max_u \frac{u^T R_1^1(t) u}{u^T R_2(t) u}\,, \tag{11}$$

but they are calculated incrementally. The inverse of $R_2(t)$ can also be calculated incrementally exactly as for $H(t)$ in Equation (6). The primal eigenvector $u_1(t)$ can now be updated using a recursive least squares approach (Rao and Principe, 2001)

$$\hat{u}_1(t) = \frac{u_1(t-1)^T R_2(t) u_1(t-1)}{u_1(t-1)^T R_1^1(t) u_1(t-1)} R_2(t)^{-1} R_1^1(t) u_1(t-1). \tag{12}$$

For numerical reasons $\hat{u}_i$ has to be normalized to

$$u_i(t) = \frac{\hat{u}_i(t)}{\|\hat{u}_i(t)\|_2} \tag{13}$$

and is later on denormalized. For the lower order filters a deflation technique is used which basically projects the matrix R_1 to a subspace which is invariant to the higher filters:

$$R_1^i(t) = \left(I - \frac{R_1^{i-1}(t) u_{i-1}(t) u_{i-1}(t)^T}{u_{i-1}(t)^T R_1^{i-1}(t) u_{i-1}(t)} \right) R_1^{i-1}(t), \tag{14}$$

where $I \in \mathbb{R}^{N_S \times N_S}$ denotes the identity matrix. The respective formula for the filter update is

$$\hat{u}_i(t) = \frac{u_i(t-1)^T R_2(t) u_i(t-1)}{u_i(t-1)^T R_1^i(t) u_i(t-1)} R_2(t)^{-1} R_1^i(t) u_i(t-1). \tag{15}$$

Note, that the resulting filters are not the solutions of the original optimization problem but they show a very fast convergence (Rao and Principe, 2001) and so usually result in approximately the same filters as for the original xDAWN (Wöhrle et al., 2015).

The remaining step is the initialization of parameters. Rao et al. provide no information about the initialization. Woehrle et al. initialized the filters with small random numbers, $\hat{A}(0)$, $R_1^1(0)$ and $R_2(0)$ with zero entries and in the implementation, $R_2(0)^{-1}$ was initialized with $\frac{1}{4}I$. Note, that $R_2(0)^{-1}$ is not the exact inverse of $R_2(0)$.

2.3 raxDAWN: Regularized axDAWN

In contrast to the xDAWN, the CSP is defined as the filter maximizing

$$\hat{u} = \arg\max_u \frac{u^T \Sigma_1 u}{u^T (\Sigma_1 + \Sigma_2) u} \tag{16}$$

where Σ_k is the covariance matrix of data belonging to class k. So $(\Sigma_1 + \Sigma_2)$ in the denominator can be seen as the counterpart to $R_2(t)$, modeling the total signal variance. But in the nominator the variance related to one single class is optimized in contrast to the signal estimate for the xDAWN, which is related to the ERP class. Adding the Tikhonov regularization

$$\lambda \|u\|^2 \tag{17}$$

to the denominator in the CSP optimization problem is supposed to come with a "mitigation of the influence of artifacts and a reduced tendency to overfitting as filters with large norm are avoided." (Samek et al., 2012). Due to the model similarities, it is reasonable to apply the same scheme to the xDAWN model definition to obtain a regularized version

$$\hat{u} = \arg\max_u \frac{u^T \hat{A}^T D^T D \hat{A} u}{u^T X^T X u + \lambda \|u\|^2} = \arg\max_u \frac{u^T R_1^1(t) u}{u^T (R_2(t) + \lambda I) u}. \tag{18}$$

Similar approaches have also been used for other filters like Discriminative Spatial Patterns (Liao et al., 2007, DSP) and Kernel Fisher Discriminant Analysis (Mika, 2003, KFDA). The xDAWN algorithm cannot be used to implement the regularized variant, because it utilizes the QR decomposition of X and it is not based on $X^T X$. But the modification of the axDAWN algorithm is straightforward: $R_1^1(0)$ has to be initialized with λI instead of zeros and

$$R_1^1(0)^{-1} = \lambda^{-1} I. \tag{19}$$

Consequently, modifying the initialization of axDAWN in the open source implementation in pySPACE (Krell et al., 2013) provides an implementation of raxDAWN. Another direct advantage of this new algorithm is, that the original initialization problem of R_2 is solved with the regularization approach because a very low regularization weight can be used.

3 EVALUATION

This section describes different experiments on EEG data to show some properties of raxDAWN and to compare it with xDAWN and axDAWN.

3.1 Data

For the offline evaluation, we used the same data as in (Wöhrle et al., 2015). Six subjects participated in the study on two different days (two sessions). On each day subjects repeated an oddball experiment five times (five sets). Each recording contains data from 120 rare and important stimuli which elicit an ERP (P300) and around 720 irrelevant stimuli which were used for the noise and as the second class for the respective classification task. Further details are provided in (Kirchner et al., 2013).

For the evaluation, we took the first of the five recordings of each day and subject for training an the remaining four jointed sets for testing. No online learning was used in the testing phase.

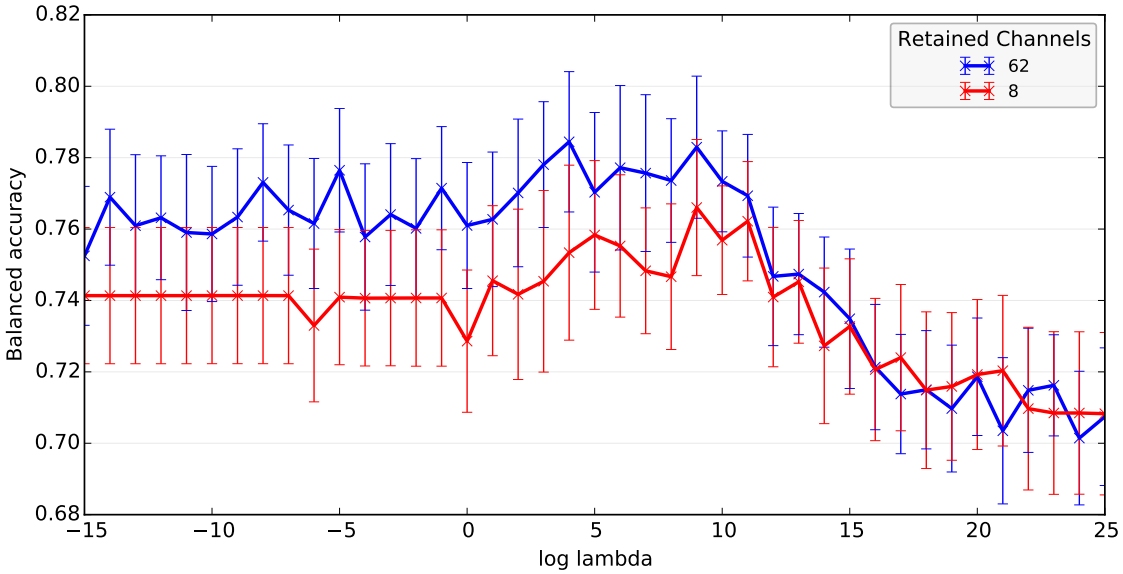

Figure 1: **Mean performance traces (with standard error) of raxDAWN for** 8 **(red) and** 62 **(blue) retained pseudo channels dependent on the regularization parameter** ($2^{\log\ \text{lambda}}$)**.**

3.2 Processing

The general processing scheme was taken from (Wöhrle et al., 2015). The open source software pySPACE (Krell et al., 2013) was again used for implementation. The data was cut into segments of one second after the stimuli. For the first noise cancellation before the application of the spatial filter, we performed a z-score standardization, decimation to 25 Hz, and a lowpass filter with a cutoff frequency of 4 Hz. After the spatial filter, straight lines were fitted every 120 ms with a size of 400 ms and the slopes were used as features. The features were standardized again and the standard support vector machine (SVM) from the LIBSVM package was used (Chang and Lin, 2011). The SVM regularization constant was optimized using a stratified 5-fold cross validation on the training data

$$C \in \left\{10^0, 10^{-1}, \ldots, 10^{-5}\right\}. \tag{20}$$

Finally the decision threshold was optimized. As performance measure, we used the balanced accuracy which is the arithmetic mean of true positive rate and true negative rate.

For statistical tests, we used the Wilcoxon signed-rank test.

3.3 Influence of the Regularization

For the first evaluation, we only used the first 24 ERPs and the respective noise data from the irrelevant stimuli for training, since the regularization is expected to pay off with few data.

In Figure 1, the effect of the regularization parameter λ is displayed for the case of no dimensionality reduction (62 retained channels) and for the reduction to the most relevant 8 pseudo channels[2]. To obtain a substantial effect, λ should be chosen larger than 1. The curves show first an increase in performance due to the regularization but then the performance drops drastically because the regularization suppresses the reduction of the noise and there is only a focus on signal enhancement. Furthermore, the choice of λ is very specific for the respective dataset and should be optimized separately with a logarithmic scaling.

3.4 Influence of the Amount of Training Data

If there is sufficient data available, the algorithm is not expected to overfit to much to the noise data. Hence for a comparison between the filters the number of used samples needs to be considered.

In this setting, we reduce the dimensionality to 8 pseudo channels and we compare the raxDAWN with axDAWN and xDAWN for different numbers of training instances. The data was used till a predefined number of ERP samples have been reached in the stream and the respective samples from the irrelevant

[2]The large standard error results from the differences between the 10 evaluations (6 subjects with 2 recoding session) and the different optimal λ values.

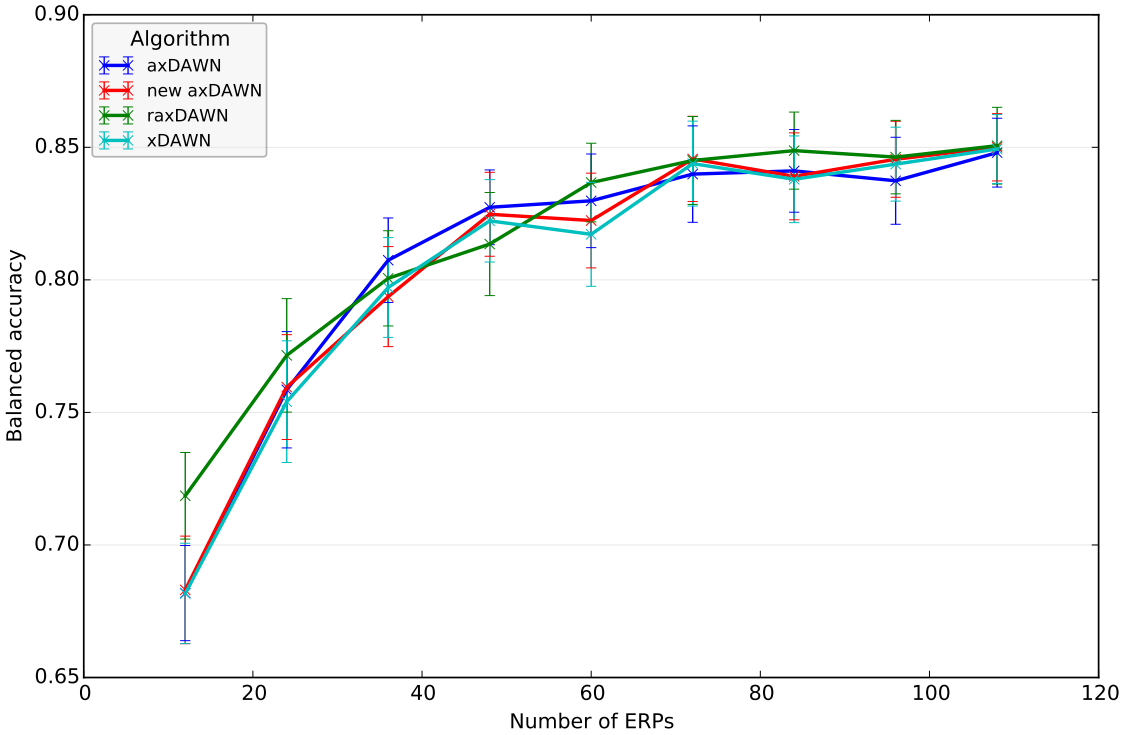

Figure 2: **Comparison of spatial filters dependent on the number of training ERP samples (mean performance and standard error).**

stimuli were used for the second class and the noise. For optimizing λ, we used the same 5-fold cross validation as for the SVM regularization parameter C, but with two repetitions to better filter out random effects

$$\lambda \in \left\{2^{-5}, 2^{-4}, \ldots, 2^{15}\right\}. \tag{21}$$

So the optimization of the parameter seems a bit more difficult and dataset specific than the C parameter.

The results are shown in Figure 2. "New axDAWN" denotes the raxDAWN with a small regularization parameter of 2^{-15}. As expected for all algorithms, performances increase with increasing training size and axDAWN and xDAWN show approximately the same performance. Interestingly, the performance of the "new axDAWN" is very close to the xDAWN due to the improved initialization. If the complete dataset is used for training, raxDAWN performs similar to (a)xDAWN but for small sizes of the training set (12 or 24 samples of the ERP class) it clearly outperforms the other spatial filters by 4 or 1% (xDAWN: $p = 0.009$, aXDAWN: $p = 0.003$, and new axDAWN: $p = 0.02$ for both numbers of samples). This result is expected, because for a larger amount of the data the noise should not have such a high influence anymore. Further, the result is consistent with the findings in (Lotte and Guan, 2010), where the highest performance increase due to regularization of CSP was achieved when the amount of available training data was very low.

In Figure 4, the chosen lambda values in the parameter optimization of the raxDAWN are shown. The values are diverse and depend on the number of used ERPs as well as on the dataset. This parameter behavior is unexpected and needs further investigation. A more sophisticated parameter optimization might result in a more stable choice and even better performance. The problem of parameter optimization can be also observed when Figure 2 is compared with Figure 3. Figure 3 displays the best performance value in the cross validation cycle for the parameter optimization. Here, the raxDAWN shows slightly better performance in the cross validation for every number of used ERPs and not only for the low number. This difference indicates a parameter overfitting.

3.5 Influence of the Number of Retained Channels

In this evaluation, we used a reduced number of samples as in Section 3.3 but varied the number of retained pseudo channels. Again the regularization parameter of the raxDAWN was optimized. The results

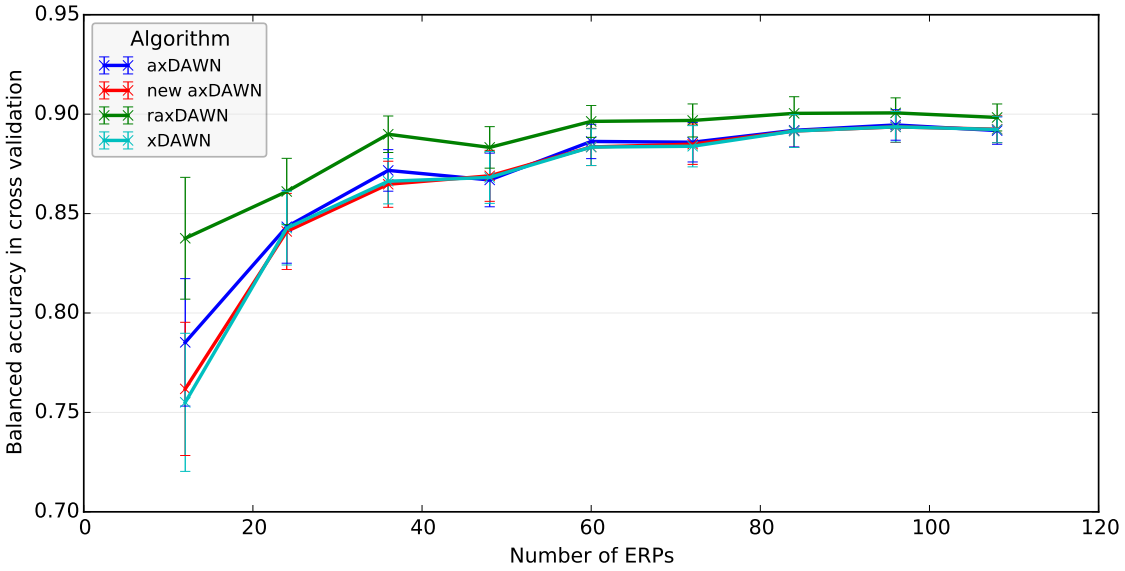

Figure 3: **Comparison of spatial filters dependent on the number of training ERP samples (mean performance and standard error).**

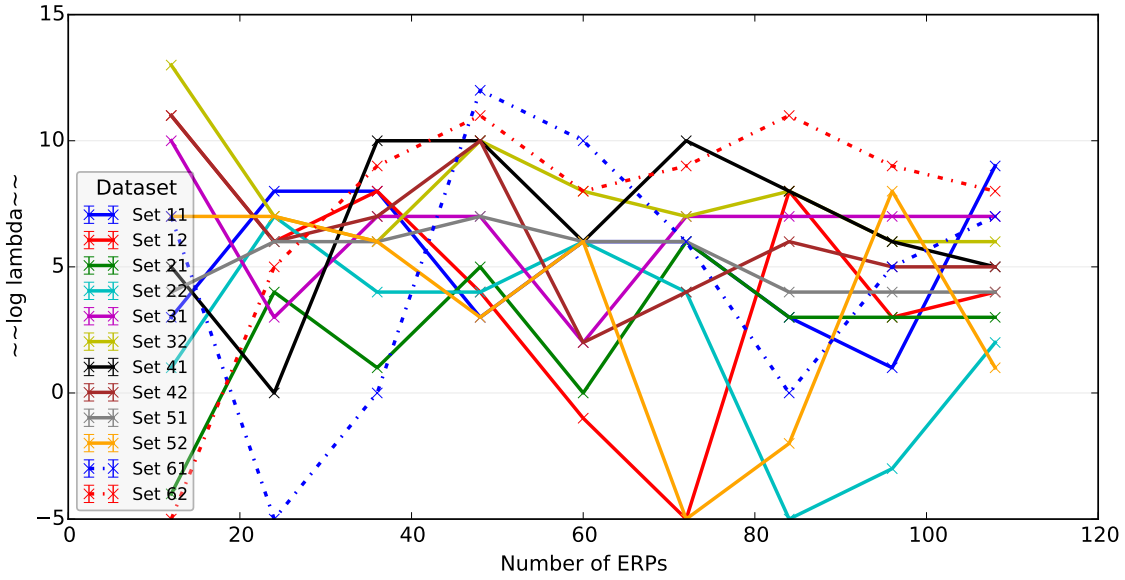

Figure 4: **Lambda values chosen by the parameter optimization ($2^{\text{log lambda}}$). The first set index corresponds to the subject number and the second index corresponds to the session number.**

are shown in Figure 5. For a number of 4, there is no large difference between the algorithms because the noise has possibly less influence. For 62 channels the raxDAWN performs slightly worse. For the group of 8, 16, and 32 retained channels, the raxDAWN outperforms the other filters by $1-3\%$ (xDAWN: $p = 0.04$, axDAWN: $p = 0.02$). The other filters show no difference in performance ($p = 0.49$).

4 CONCLUSION

In this paper we successfully applied the regularization concept for spatial filters to the axDAWN algorithm and introduced the new raxDAWN algorithm. We evaluated the algorithm on data from a BCI experiment and showed that it improves xDAWN and axDAWN especially in the initialization when only few training data is available.

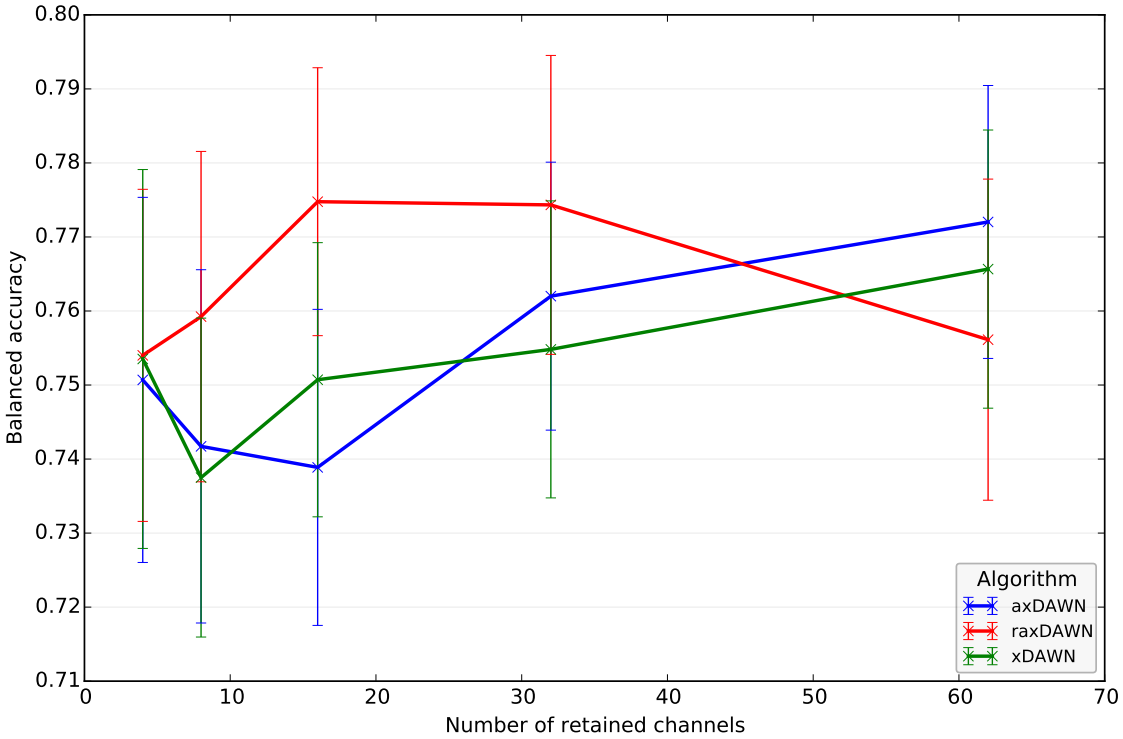

Figure 5: **Comparison of spatial filters dependent on the number of retained channels (mean performance and standard error).**

In the future, we would like to analyze other regularization methods. For example, the first filter from a previous session or a different subject could be used for the regularization in a zero training setup instead of using the filter for initialization as done in (Wöhrle et al., 2015). Another point is a deeper analyses of the optimal choice of the regularization parameter to speed up the optimization. One possibility might be an online optimization which combines some models weighted by their accuracy.

ACKNOWLEDGEMENTS

This work was supported by the Federal Ministry of Education and Research (BMBF, grant no. 01IM14006A).

We thank Marc Tabie, Yohannes Kassahun and our anonymous reviewers for giving useful hints to improve the paper. We thank Su Kyoung Kim for providing the statistics.

REFERENCES

Blankertz, B., Lemm, S., Treder, M., Haufe, S., and Müller, K.-R. (2011). Single-Trial Analysis and Classification of ERP Components–a Tutorial. *NeuroImage*, 56(2):814–825.

Blankertz, B., Tomioka, R., Lemm, S., Kawanabe, M., and Müller, K.-R. (2008). Optimizing Spatial filters for Robust EEG Single-Trial Analysis. *IEEE Signal Processing Magazine*, 25(1):41–56.

Buttfield, A., Ferrez, P. W., and Millán, J. d. R. (2006). Towards a robust BCI: error potentials and online learning. *IEEE transactions on neural systems and rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society*, 14(2):164–8.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27.

Ghaderi, F. and Kirchner, E. A. (2013). Periodic Spatial Filter for Single Trial Classification of Event Related Brain Activity. In *Biomedical Engineering*, Calgary,AB,Canada. ACTAPRESS.

Golub, G. H. and Van Loan, C. F. (1996). *Matrix computations*. Johns Hopkins University Press.

Kirchner, E. A., Kim, S. K., Straube, S., Seeland, A., Wöhrle, H., Krell, M. M., Tabie, M., and Fahle, M.

(2013). On the applicability of brain reading for predictive human-machine interfaces in robotics. *PloS ONE*, 8(12):e81732.

Krell, M. M., Straube, S., Seeland, A., Wöhrle, H., Teiwes, J., Metzen, J. H., Kirchner, E. A., and Kirchner, F. (2013). pySPACE a signal processing and classification environment in Python. *Frontiers in Neuroinformatics*, 7(40):1–11.

Krusienski, D. J., Sellers, E. W., Cabestaing, F., Bayoudh, S., McFarland, D. J., Vaughan, T. M., and Wolpaw, J. R. (2006). A comparison of classification techniques for the P300 Speller. *Journal of neural engineering*, 3(4):299–305.

Liao, X., Yao, D., Wu, D., and Li, C. (2007). Combining spatial filters for the classification of single-trial EEG in a finger movement task. *IEEE transactions on biomedical engineering*, 54(5):821–31.

Lotte, F. and Guan, C. (2010). Spatially Regularized Common Spatial Patterns for EEG Classification. In *2010 20th International Conference on Pattern Recognition (ICPR)*, pages 3712–3715.

Mika, S. (2003). *Kernel Fisher Discriminants*. PhD thesis, Technische Universität Berlin.

Mika, S., Rätsch, G., and Müller, K.-R. (2001). A mathematical programming approach to the kernel fisher algorithm. *Advances in Neural Information Processing Systems 13 (NIPS 2000)*, pages 591–597.

Rao, Y. and Principe, J. (2001). An RLS type algorithm for generalized eigendecomposition. In *Neural Networks for Signal Processing XI: Proceedings of the 2001 IEEE Signal Processing Society Workshop (IEEE Cat. No.01TH8584)*, pages 263–272. IEEE.

Rivet, B., Souloumiac, A., Attina, V., and Gibert, G. (2009). xDAWN Algorithm to Enhance Evoked Potentials: Application to Brain-Computer Interface. *IEEE Transactions on Biomedical Engineering*, 56(8):2035–2043.

Samek, W., Vidaurre, C., Müller, K.-R., and Kawanabe, M. (2012). Stationary common spatial patterns for brain-computer interfacing. *Journal of neural engineering*, 9(2):026013.

van Erp, J., Lotte, F., and Tangermann, M. (2012). Brain-Computer Interfaces: Beyond Medical Applications. *Computer*, 45(4):26–34.

Wöhrle, H., Krell, M. M., Straube, S., Kim, S. K., Kirchner, E. A., and Kirchner, F. (2015). An Adaptive Spatial Filter for User-Independent Single Trial Detection of Event-Related Potentials. *IEEE transactions on biomedical engineering*, PP(99):1.

Wöhrle, H., Teiwes, J., Krell, M. M., Kirchner, E. A., and Kirchner, F. (2013). A Dataflow-based Mobile Brain Reading System on Chip with Supervised Online Calibration - For Usage without Acquisition of Training Data. In *Proceedings of the International Congress on Neurotechnology, Electronics and Informatics*, pages 46–53, Vilamoura, Portugal. SciTePress.

Zander, T. O. and Kothe, C. (2011). Towards passive brain-computer interfaces: applying brain-computer interface technology to human-machine systems in general. *Journal of Neural Engineering*, 8(2):025005.