

# Application of the Evidence Framework to Brain-Computer Interfaces

Ulrich Hoffmann, Gary Garcia, Jean-Marc Vesin, and Touradj Ebrahimi  
Signal Processing Institute  
Swiss Federal Institute of Technology - EPFL, CH-1015 Lausanne Switzerland

*Abstract*—A brain-computer interface (BCI) is a communication system, that implements the principle of "think and make it happen without any physical effort". This means a BCI allows a user to act on his environment only by using his thoughts, without using peripheral nerves and muscles. Nearly all BCIs contain as a core part a machine learning algorithm, which learns from training data a function, that can be used to discriminate different brain activities.

In the present work we use a bayesian framework for machine learning, the evidence framework [1], [2], to develop a variant of linear discriminant analysis for the use in a BCI based on electroencephalographic measurements (EEG). Properties of the resulting algorithm are: a) a continuous probabilistic output is given, b) fast estimation of regularization constants, and c) the possibility to select among different feature sets, the one which is most promising for classification.

The algorithm has been tested on one dataset from the BCI competition 2002 and two datasets from the BCI competition 2003 and provides a classification accuracy of 95%, 81%, and 79% respectively.

*Keywords*—Brain-computer interface, bayesian framework, EEG, evidence framework, linear discriminant analysis

## I. INTRODUCTION

Brain-computer interface is an area of research that is gaining lots of interest recently. The number of different approaches has been rapidly growing during the last years and considerable progress has been made. This progress is resulting from many factors, among them being the development of advanced signal processing and machine learning methods for BCI.

Many different algorithms have been tested on the machine learning side of BCI. These include, among others, support vector machines [7] and kernel methods [6], neural networks [13], hidden markov models [8], and variations of linear discriminant analysis (LDA) [7].

LDA is one of the simplest methods among those mentioned above and yet often gives very good results in the area of BCI. This can be observed in the results of the BCI competition 2003 [10], where 6 out of 10 winning algorithms used LDA or modified versions of LDA for classification.

The success of LDA in the context of BCI has motivated us to study extensions of the basic LDA algorithm, that are desirable for BCI. In particular we concentrate on the evidence framework, originally described in the regression [1] and neural networks context [2]. This framework can also be used for other types of classifiers [3], [4].

The main goal of this work is to show, that the combination of the above mentioned framework and LDA results in a classifier, that has properties which are very useful for BCI. These properties are:

### a) *Probabilistic output*

For a given feature vector the classifier outputs a class label, and also a value between zero and one that indicates how probable it is, that the class label is correct.

### b) *Fast estimation of regularization parameters*

Typically, regularization constants are estimated through a time-consuming cross-validation procedure. The algorithm presented here allows to estimate regularization parameters very quickly, without using cross-validation.

### c) *Feature set selection*

In a BCI, it has to be decided which features are useful for classification. The algorithm presented in this paper allows to rank different feature sets with respect to their discriminative power.

To our best knowledge the combination of the above mentioned properties in one algorithm is new in the context of BCI. The method in [12] is similar in spirit to the approach presented here, but concentrates more on autoregressive model based time-series classification, and online adaptation of classifier parameters. The algorithm in [13] gives probabilistic output but does not have the additional properties of the algorithm presented here.

The outline of the rest of the paper is as follows: In Sec. II, the datasets used in this work are described and the feature extraction method is summarized. In Sec. III, the application of the evidence framework to LDA is described. Section IV presents results obtained with the algorithm.

## II. FEATURE EXTRACTION

Three datasets are analyzed in this work. The first two datasets ("self-paced 2s" and "self-paced 1s") contain multichannel EEG segments recorded during voluntary finger movement. Segments are 1.5 seconds, respectively 0.5 seconds long, and end 0.12, respectively 0.13 seconds before the onset of movement. A more detailed description of these datasets can be found in [7], [10].

To extract features, the discrete Fourier transform (DFT) is applied to each channel of a EEG segment. Feature vectors are obtained by concatenation of the real and imaginary parts of the Fourier coefficients. This is a simple method to represent the event related potentials (ERPs) that are characteristic for the datasets mentioned above. Before applying the DFT, each channel is multiplied by a window  $\omega(s) = 1 - \cos(s\pi/S)$ , where  $S$  is the length of one EEG segment (as in [7]). This is done because the ERPs are more pronounced towards the end of the segments.

The third dataset ("motor-imagery") contains 9-second long EEG segments, recorded during an imagined movement task. Seconds 1-3 contain a preparation phase and are not

used to obtain the results in the following sections. More details about this dataset can be found in [10].

After applying the DFT to each channel the feature vectors for the third dataset are obtained by concatenation of the squares of the absolute values of the Fourier coefficients. This is a simple way to represent the rhythmic activity which is characteristic for motor-imagery tasks.

For all three datasets we build reduced feature sets containing only the coefficients of one frequency band, of a certain width. The classification algorithm then chooses which frequency band and which width is optimal for classification. All features are normalized to the range  $[-1, 1]$ . This is done because the algorithm presented in the next section gives the best results when all features have the same scale.

### III. LINEAR DISCRIMINANT ANALYSIS IN THE EVIDENCE FRAMEWORK

The algorithm presented in this section learns from training data a function  $f(x_j(i)) = y_j$ , which maps feature vectors to class labels. The training data consists of a set of feature vectors  $x_j(i) \in \mathbb{R}^W$  and class labels  $y_j \in \{1, -1\}$ . Class labels indicate to which brain activity a feature vector corresponds.

It is assumed, that  $f$  is a linear function, i.e. each class label can be expressed as a weighted sum of the features in the corresponding feature vector and is corrupted by a certain amount of noise  $n_j$ :

$$y_j = w^t x_j(i) + n_j, \quad j = 1 \dots N, \quad w \in \mathbb{R}^W. \quad (1)$$

The constant  $N$  denotes the number of feature vectors in the training set,  $W$  indicates the number of features, and the index  $i$  indicates which feature set  $\mathcal{F}_i$  was used to build the feature vectors. It is assumed that the  $n_j$  are independent, identically distributed samples of a zero mean, Gaussian noise process.

In the following a probability distribution over all weight vectors  $w \in \mathbb{R}^W$  is calculated. This probability distribution is then used for classification of feature vectors.

To this end, a hierarchical Bayesian approach to inference, namely the evidence framework, is combined with LDA. On the first level of inference, a parameterized prior and a parameterized likelihood for the weight vector are defined. Then Bayes rule is used to derive the posterior distribution for the weight vector.

On the second level of inference, a prior and a likelihood for the parameters of prior and likelihood on the first level are defined. Again the posterior distribution is derived with the help of Bayes rule.

On the third level of inference, a prior and a likelihood for different feature sets are defined and used to find a posterior distribution for the feature sets.

#### A. First Level of Inference

Since the selection of feature sets is done only on the third level of inference, we drop the dependency on the feature set  $\mathcal{F}_i$  for now, and denote by  $X$  the matrix resulting from the horizontal stacking of all feature vectors, and by  $Y$  the row-vector resulting from the concatenation of all class labels. Given a dataset  $D = \{X, Y\}$  the assumption

of Gaussian noise then leads to the joint likelihood for the inverse variance  $\beta$  and the weight vector  $w$ :

$$p(D|\beta, w) = \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}} \exp\left(-\frac{\beta}{2}\|w^t X - Y\|_2^2\right). \quad (2)$$

In the following, it is assumed that  $\beta$  is fixed to a value which will be inferred on the second level of inference.

The prior for a weight vector  $w$  is again parameterized by a variable  $\alpha$  which is regarded as given on the current level of inference:

$$p(w|\alpha) = \left(\frac{\alpha}{2\pi}\right)^{\frac{W}{2}} \exp\left(-\frac{\alpha}{2}\|w\|_2^2\right). \quad (3)$$

The above prior is equivalent to a widely used regularization technique, known as Tikhonov regularization, or weight decay.

Given likelihood and prior, we can build the posterior:

$$p(w|\beta, \alpha, D) = \frac{p(D|\beta, w)p(w|\alpha)}{\int p(D|\beta, w)p(w|\alpha) dw}. \quad (4)$$

Since both prior and likelihood are Gaussian, the posterior is also Gaussian and its parameters can be derived from the parameters of likelihood and prior. The posterior can be used to find the distribution of class labels for feature vector  $x$ :

$$p(y|\beta, \alpha, x, D) = \int p(y|\beta, x, w)p(w|\beta, \alpha, D) dw$$

$$p(y|\beta, x, w) = \left(\frac{\beta}{2\pi}\right)^{\frac{1}{2}} \exp\left(-\frac{\beta}{2}(w^t x - y)^2\right). \quad (5)$$

Equation 5 allows us to calculate the probability that feature vector  $x$  has class label  $y = 1$  (a similar reasoning is used for  $y = -1$ ):

$$p(y \geq 0|\beta, \alpha, x, D) = \left(\frac{\sigma}{2\pi}\right)^{\frac{1}{2}} \int_0^\infty \exp\left(-\frac{\sigma}{2}(y - \hat{y})^2\right) dy. \quad (6)$$

The parameters of the Gaussian in Eq. 6 are given by:

$$\hat{y} = A^{-1}XYx, \quad \sigma = \frac{\beta}{1 + \beta x^t A^{-1}x}$$

$$A = (\beta XX^t + \alpha I). \quad (7)$$

The output probabilities, given by Eq. 6 can be used in at least two different ways in a BCI. First, EEG segments that can be classified with only low reliability, can be identified and excluded from further processing. This helps to avoid user frustration due to too many wrong decisions. Second, probabilistic output can be used for continuous control, e.g. 1D cursor control as in [11].

#### B. Second Level of Inference

To find the parameters  $\alpha$  and  $\beta$  which were regarded as given in the previous section, we proceed as before by defining a prior, a likelihood and then calculating the posterior distribution. The maximum of the posterior is then used as value for  $(\beta, \alpha)$  on the first level of inference, i.e. a maximum a posteriori (MAP) estimate of the regularization parameters is calculated.

Independence of  $\alpha$  and  $\beta$  is assumed and the following prior, which is uniform over  $\ln(\beta)$  and  $\ln(\alpha)$  is assigned:

$$p(\beta, \alpha) = p(\beta)p(\alpha) = \frac{1}{\beta\alpha}. \quad (8)$$

This prior is commonly used for scale parameters and expresses the fact that we have no a priori information about the scale of the variables involved in our experiment (for a very good discussion of this topic see [9]).

The likelihood is the normalizing integral in Eq. 4, i.e.:

$$p(D|\beta, \alpha) = \int p(D|\beta, w)p(w|\alpha) dw. \quad (9)$$

This is also called the *evidence* for  $(\beta, \alpha)$ . The posterior is:

$$p(\beta, \alpha|D) = \frac{p(D|\beta, \alpha)p(\beta, \alpha)}{\int p(D|\beta, \alpha)p(\beta, \alpha) d\beta d\alpha}. \quad (10)$$

Strictly, the posterior defined by the above equation would have to be used for all calculations involving  $(\beta, \alpha)$ , but this would pose problems in computing the integrals on the first level of inference. As in [1], [2], [5] the posterior is approximated by a Dirac function at its mode:

$$p(\beta, \alpha|D) \approx \delta(\hat{\beta}, \hat{\alpha})p(D|\beta, \alpha)p(\beta, \alpha). \quad (11)$$

To find the mode  $\hat{\beta}, \hat{\alpha}$  of the posterior, Eqs. 2, 3, and 9 are used to expand Eq. 10:

$$\begin{aligned} p(\beta, \alpha|D) &= c \int \exp(-\frac{\beta}{2}\|w^t X - Y\|_2^2 - \frac{\alpha}{2}\|w\|_2^2) dw \\ c &= \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}} \left(\frac{\alpha}{2\pi}\right)^{\frac{W}{2}} \frac{1}{\beta\alpha}. \end{aligned} \quad (12)$$

The integral in Eq. 12 is evaluated and the logarithm is taken:

$$\begin{aligned} \ln(p(\beta, \alpha|D)) &= \frac{N}{2} \ln\left(\frac{\beta}{2\pi}\right) + \frac{W}{2} \ln\left(\frac{\alpha}{2\pi}\right) - \ln(\beta\alpha) \\ &+ \frac{W}{2} \ln(2\pi) - \frac{1}{2} \ln(\det(A)) + \frac{1}{2}\beta^2 Y X^t A^{-1} X Y^t \end{aligned} \quad (13)$$

Whereas in [1], [2], [5] a heuristic is used to find the maximum of Eq. 13 we calculate derivatives and use a standard optimization algorithm. The MATLAB optimization toolbox is used to find the maximum.

The fast estimation of regularization constants as presented above, can be of considerable importance in a BCI. Indeed often training is performed in several steps and parameters have to be updated regularly. Unlike cross-validation the method presented above uses no test set, which means that all the available data can be used to estimate regularization constants.

### C. Third Level of Inference

Up to here it has been shown, how to derive a distribution for the weight vector  $w$  given a prior, a likelihood and the training data. Additionally it has been shown, how to find a MAP estimate of the parameters for the first level prior and likelihood. Now different sets of features  $\mathcal{F}_i$  will be compared. A flat prior over different feature sets is specified ( $F$  denotes the number of feature sets):

$$p(\mathcal{F}_i) = \frac{1}{F}. \quad (14)$$

The likelihood for a given feature set is given by the normalizing integral in Eq. 10:

$$p(D|\mathcal{F}_i) = \int p(D|\beta, \alpha, \mathcal{F}_i)p(\beta, \alpha) d\beta d\alpha. \quad (15)$$

The posterior is proportional to the product of likelihood and prior:

$$p(\mathcal{F}_i|D) \propto p(D|\mathcal{F}_i)p(\mathcal{F}_i) \quad (16)$$

The method proposed in [1], [2], [5] is used to evaluate the integral in Eq. 15 and to find the posterior distribution.

Examples for feature sets that can be selected on the third level of inference are the frequency band in classification based on event related synchronisation (ERS) and event related desynchronisation (ERD), the order of an AR-Model in classification based on autoregressive modeling, or the number of common spatial patterns that are used to detect a mental activity.

## IV. RESULTS

In this section, first, we report on the classification accuracy achieved with the algorithm, then we give a description of the feature sets selected by the classifier.

### A. Classification Accuracy

To allow for a direct comparison with the results obtained in the BCI competitions, the algorithm was trained only with the competition training sets and tested on the competition test sets. The column "Comp." in Tab. I contains, the results of the competition winners, the column "Test set" contains the result obtained in the present work.

Since an estimate of classification accuracy based on only one test set is not very reliable, the algorithm was also tested in a 50-fold cross-validation loop, i.e. with 50 randomly chosen test sets of the same size as the competition test set. Results are shown in Tab. I in the column "CV".

For the cross-validation loop as well as for the competition test sets all parameters, including parameters for first level prior and likelihood, constants used for normalization to the range [-1,1], and the selected feature set were estimated solely from the training data.

The results show that the algorithm presented in this paper classifies single-trial EEG with high accuracy, comparable to or better than the state of the art. For the third dataset a direct comparison is not possible, since a different evaluation criterion was used in the competition.

To test how useful the probabilistic output is, we have plotted the classification accuracy and the percentage of rejected trials, against a threshold on the confidence level given by our classifier (see Fig. 1). More precisely, the classification accuracy in Fig. 1 was obtained by only taking into account trials for which the classifier was "sure", i.e. for which the maximum of the probabilities given by Eq. 6 was larger than the threshold  $p_t$ . The remaining trials were rejected. It can be seen that experimentally the confidence level given by the classifier is a lower bound on the probability of error. This is a very useful feature of our algorithm.

TABLE I

CLASSIFICATION ACCURACY FOR THE THREE DATASETS. SEE TEXT FOR DETAILS.

Dataset	Comp.	Test set	CV
self-paced 2s (2002)	96.0	97.0	95.6 $\pm$ 1.7
self-paced 1s (2003)	84.0	80.0	81.3 $\pm$ 3.9
motor-imagery (2003)	n/a	81.4	79.5 $\pm$ 2.9

### B. Selected feature sets

To assess the selected feature sets the algorithm was used in a 50-fold cross-validation loop and the number of times each feature was selected was recorded. The results in Fig. 2 show that the algorithm has chosen to represent the ERPs in the two self-paced datasets with low frequency components only. This correlates well with the notion of ERPs in the self-paced paradigm - the ERPs are relatively slow potential shifts in the EEG before the onset of the movement. For the motor-imagery dataset, most of the time a part of the mu-band (8-12 Hz) was chosen. Note that for a small number of times the band 20-22.5 Hz was chosen. Again, this correlates with what is known about the physiology of motor-imagery tasks.

It has to be stressed here, that all of the above results were obtained solely from the data. The only a priori specification was, that the algorithm should search for frequency bands that are promising for classification. Thus feature set selection can be used to adapt to users. In addition, it could be used in settings where only a very small amount of physiological knowledge exists and everything has to be learnt from the data.

### V. CONCLUSION AND FUTURE WORK

In this work, a probabilistic version of LDA was developed and tested on datasets from BCI competitions 2002 and 2003. The algorithm does not rely on user-defined constants and learns all parameters, including regularization parameters and a feature set from the data. Probabilistic output is used to reject trials that cannot be classified with certainty.

To assess the performance in a real BCI, the algorithm has to be tested in an online setting. In addition, we want to explore if prior information, gathered in preceding experiments, can be used to reduce training times and achieve better classification accuracy.

### ACKNOWLEDGMENT

This work was funded in part by Swiss National Science Foundation grant no. 2153-067852.02. We thank Yannick

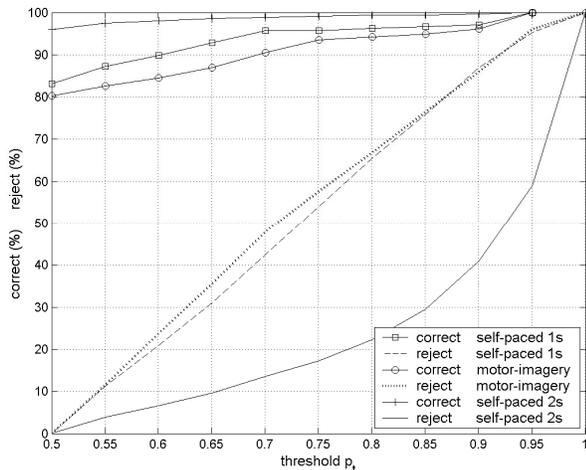


Fig. 1. Plot of classification accuracy and rejection rate vs. probability threshold.

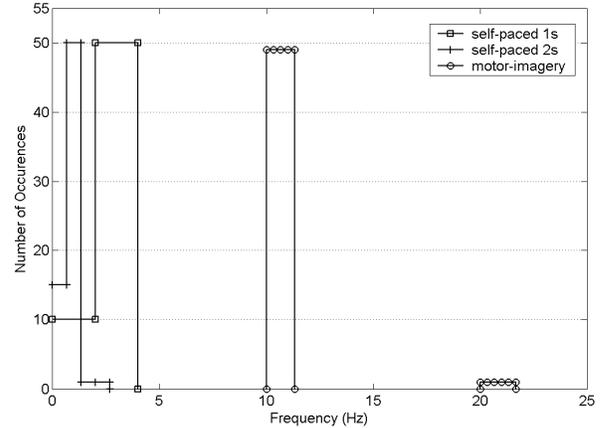


Fig. 2. Feature sets selected by the algorithm.

Maret and Markus Flierl for fruitful discussions and insights.

### REFERENCES

- [1] D.J.C. MacKay, "Bayesian Interpolation," *Neural Comp.*, vol. 4, no. 3, pp. 415-447, 1992
- [2] D.J.C. MacKay, "The evidence framework applied to classification networks," *Neural Comp.*, vol. 4, no. 5, pp. 720-736, 1992
- [3] M. Tipping, "Sparse Bayesian Learning and the Relevance Vector Machine," *Journal of Machine Learning Research*, vol. 1, pp. 211-244, 2001
- [4] T. Van Gestel, J. A. K. Suykens, G. Lanckriet, A. Lambrechts, B. De Moor, and J. Vandewalle, "Bayesian framework for least-squares support vector machine classifiers, Gaussian processes, and kernel fisher discriminant analysis," *Neural Comp.*, vol. 4, no. 5, pp. 1115-1147, 2002
- [5] C.M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995
- [6] G.N. Garcia, U. Hoffmann, T. Ebrahimi, and J.-M. Vesin, "Direct Brain-Computer Communication through EEG Signals," To appear in the IEEE EMBS Book Series on Neural Engineering, 2004
- [7] B. Blankertz, G. Curio, and K.-R. Müller, "Classifying single trial EEG: Towards brain computer interfacing," *Advances in Neural Inf. Proc. Systems (NIPS 01)*, vol. 14, pp. 157-164, 2002
- [8] B. Obermaier, C. Guger, C. Neuper, and G. Pfurtscheller, "Hidden Markov models for online classification of single trial EEG data," *Pattern Recognition Letters*, vol. 22, pp. 1299-1309, 2001
- [9] E.T. Jaynes, *Probability Theory - The Logic of Science*, Cambridge University Press, 2003
- [10] B. Blankertz, K.-R. Müller, G. Curio, T. M. Vaughan, G. Schalk, J. R. Wolpaw, A. Schlögl, C. Neuper, G. Pfurtscheller, T. Hinterberger, M. Schröder, and N. Birbaumer, "The BCI competition 2003: Progress and perspectives in detection and discrimination of EEG single trials," *IEEE Trans. Biomed. Eng.*, to appear, 2004
- [11] D.J. McFarland, A.T. Lefkowitz, and J.R. Wolpaw, "Design and operation of an EEG-based brain-computer interface (BCI) with digital signal processing technology," *Behav. Res. Meth., Instru. & Comput.* vol. 29, pp. 337-345, 1997
- [12] P. Sykacek, S. Roberts, M. Stokes, E. Curran, M. Gibbs and L.C. Pickup, "Probabilistic methods in BCI research," *IEEE Trans. Neur. Sys. & Rehab. Eng.*, pp. 192-195, 2003.
- [13] J.d.R. Millán, J. Mourião, M. Franzé, F. Cincotti, M. Varsta, J. Heikkinen, and F. Babiloni, "A Local Neural Classifier for the Recognition of EEG Patterns Associated to Mental Tasks," *IEEE Trans. Neur. Net.*, vol.13, no.3, pp. 678-686, 2002