Adaptive Methods in BCI Research - An Introductory Tutorial

Alois Schlögl, Carmen Vidaurre, and Klaus-Robert Müller

1 Introduction

1.1 Why We Need Adaptive Methods

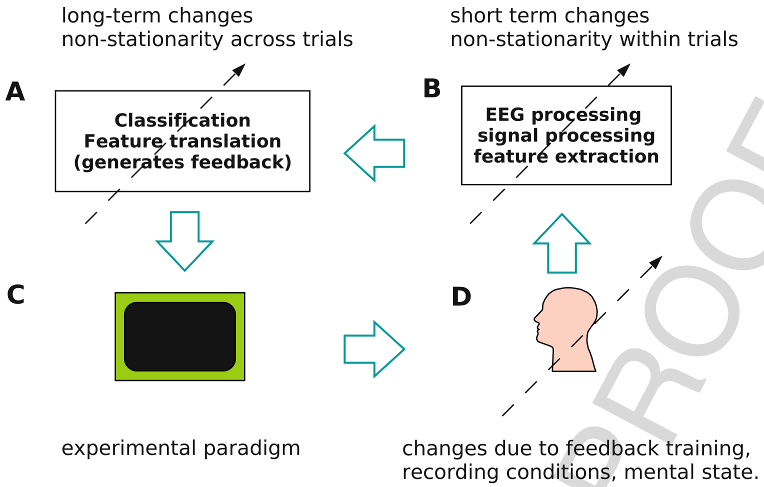
This chapter tackles a difficult challenge: presenting signal processing material to non-experts. This chapter is meant to be comprehensible to people who have some math background, including a course in linear algebra and basic statistics, but do not specialize in mathematics, engineering, or related fields. Some formulas assume the reader is familiar with matrices and basic matrix operations, but not more advanced material. Furthermore, we tried to make the chapter readable even if you skip the formulas. Nevertheless, we include some simple methods to demonstrate the basics of adaptive data processing, then we proceed with some advanced methods that are fundamental in adaptive signal processing, and are likely to be useful in a variety of applications. The advanced algorithms are also online available [30]. In the second part, these techniques are applied to some real-world BCI data.

All successful BCI systems rely on efficient real-time feedback. Hence, BCI data processing methods must be also suitable for online and real-time processing. This requires algorithms that can only use sample values from the past and present but not the future. Such algorithms are sometimes also called causal algorithms. Adaptive methods typically fulfill this requirement, while minimizing the time delay. The data processing in BCIs consists typically of two main steps, (i) signal processing and feature extraction, and (ii) classification or feature translation (see also Fig. 1). This work aims to introduce adaptive methods for both steps; these are also closely related to two types of non-stationarities - namely short-term changes related to different mental activities (e.g. hand movement, mental arithmetic, etc.), and less specific long term changes related to fatigue, changes in the recording conditions, or effects of feedback training.

AQ1

A. Schlögl (✉)
Technische Universität, Krenngasse 37, 8010 Graz, Austria
e-mail: alois.schloegl@tugraz.at

DOI 10.1007/978-3-642-02091-9_18,



This figure will be printed in b/w

Fig. 1 Scheme of a Brain–Computer Interface. The brain signals are recorded from the subject (**d**) and processed for feature extraction (**b**). The features are classified and translated into a control signal (**a**), and feedback is provided to the subject. The arrows indicate a possible variation over time (see also the explanation in the text)

The first type of changes (i.e. short-term changes) is addressed in the feature extraction step (B in Fig. 1). Typically, these are changes within each trial that are mainly due to the different mental activities for different tasks. One could also think of short-term changes unrelated to the task, which are typically the cause for imperfect classification and are often difficult to distinguish from the background noise, so these are not specifically addressed here.

The second type of non-stationarities are long-term changes caused by e.g. a feedback training effect. More recently, adverse long-term changes (e.g. due to fatigue, changed recording conditions) have been discussed. These non-stationarities are addressed in the classification and feature translation step (part a in Fig. 1).

Accordingly, we do see class-related short-term changes (due to the different mental tasks), class-related long-term changes (due to feedback training), and unspecific long-term changes (e.g. due to fatigue). The source of the different non-stationarities are the probands and its brain signals as well as the recording conditions (part d in Fig. 1) [24, 50, 51]. Specifically, feedback training can modify the subjects' EEG patterns, and this might require an adaptation of the classifier which might change again the feedback. The possible difficulties of such a circular relation have been also discussed as the "man–machine learning dilemma" [5, 25]. Theoretically, a similar problem could also occur for short-term changes. These issues will be briefly discussed at the end of this chapter.

Segmentation-type approaches are often used to address non-stationarities. For example, features were extracted from short data segments (FFT-based Bandpower [23, 25, 27], AR-based spectra in [18], slow cortical potentials by [2], or CSP combined with Bandpower [4, 6, 7, 16]). Also classifiers were obtained and retrained from specific sessions (e.g. [4, 25]) or runs. A good overview on various methods is provided in chapter "Digital signal Processing and Machine Learning" in this volume [17].

Segmentation methods may cause sudden changes from one segment to the next one. Adaptive methods avoid such sudden changes, but are continuously updated to the new situation. Therefore, they have the potential to react faster, and have a smaller deviation from the true system state. A sliding window approach (segmentation combined with overlapping segments) can also provide a similar advantage, however, we will demonstrate that this comes with increased computational costs.

In the following pages, some basic adaptive techniques are first presented and discussed, then some more advanced techniques are introduced. Typically, the stationary method is provided first, and then the adaptive estimator is introduced. Later, a few techniques are applied to adaptive feature extraction and adaptive classification methods in BCI research, providing a comparison between a few adaptive feature extraction and classification methods.

A short note about the notation: first, all the variables that are a function of time will be denoted as $f(t)$ until Sect. 1.3. Then, the subindex k will be used to denote sample-based adaptation and n to trial-based adaptation.

1.2 Basic Adaptive Estimators

1.2.1 Mean Estimation

Let us assume the data as a stochastic process $x(t)$, that is series of stochastic variables x ordered in time t; at each instant t in time, the sample value $x(t)$ is observed, and the whole observed process consists of N observations. Then, the (overall) mean value μ_x of $x(t)$ is

$$mean(x) = \mu_x = \frac{1}{N}\sum_{t=1}^{N} x(t) = E\langle x(t)\rangle \tag{1}$$

In case of a time-varying estimation, the mean can be estimated with a sliding window approach using

$$\mu_x(t) = \frac{1}{\sum_{i=0}^{n-1} w_i}\sum_{i=0}^{n-1} w_i \cdot x(t-i) \tag{2}$$

where n is the width of the window and w_i are the weighting factors. A simple solution is using a rectangular window i.e. $w_i = 1$ resulting in

$$\mu_x(t) = \frac{1}{n}\sum_{i=0}^{n-1} x(t-i) \tag{3}$$

For the rectangular window approach ($w_i = const$), the computational effort can be reduced by using this recursive formula

$$\mu_x(t) = \mu_x(t-1) + \frac{1}{n} \cdot (x(t) - x(t-n)) \tag{4}$$

Still, one needs to keep the n past sample values in memory. The following adaptive approach needs no memory for its past sample values

$$\mu_x(t) = (1 - UC) \cdot \mu_x(t-1) + UC \cdot x(t) \tag{5}$$

whereby UC is the update coefficient, describing an exponential weighting window

$$w_i = UC \cdot (1 - UC)^i \tag{6}$$

with a time constant of $\tau = 1/(UC \cdot F_s)$ if the sampling rate is F_s. This means that an update coefficient UC close to zero emphasizes the past values while the most recent values have very little influence on the estimated value; a larger update coefficient will emphasize the most recent sample values, and forget faster the ealiers samples. Accordingly, a larger update coefficient UC enables a faster adaptation. If the update coefficient UC becomes too large, the estimated values is based only on a few samples values. Accordingly, the update coefficient UC can be used to determine the tradeoff between adaptation speed and estimation accuracy.

All mean estimators are basically low pass filters whose bandwidths (or edge frequency of the low pass filter) are determined by the window length n or the update coefficient UC. The relationship between a rectangular window of length n and an exponential window with $UC = \frac{1}{n}$ is discussed in [36] (Sect. 3.1). Thus, if the window length and the update coefficient are properly chosen, a similar characteristic can be obtained.

Table 1 shows the computational effort for the different estimators. The stationary estimator is clearly not suitable for a real-time estimation; the sliding window approaches require memory that is proportional to the window size and are often computationally more expensive than adaptive methods. Thus the adaptive method has a clear advantage in terms of computational costs.

Table 1 Computational effort of mean estimators. The computational and the memory effort per time step are shown by using the O-notation, with respect to the number of samples N and the window size n.[1]

Method	Memory effort	Computational effort
stationary	$O(N)$	$O(N)$
weighted sliding window	$O(N \cdot n)$	$O(N \cdot n)$
rectangular sliding window	$O(N \cdot n)$	$O(N \cdot n)$
recursive (only for rectangular)	$O(N \cdot n)$	$O(N \cdot n)$
adaptive (exponential window)	$O(N)$	$O(N)$

1.2.2 Variance Estimation

The overall variance σ_x^2 of x_t can be estimated with

$$var(x) = \sigma_x^2 = \frac{1}{N}\sum_{t=1}^{N}(x(t)-\mu)^2 = E\langle(x(t)-\mu)^2\rangle \tag{7}$$

$$= \frac{1}{N}\sum_{t=1}^{N}(x(t)^2 - 2\mu x(t) + \mu^2) = \tag{8}$$

$$= \frac{1}{N}\sum_{t=1}^{N}x(t)^2 - \frac{1}{N}\sum_{t=1}^{N}2\mu x(t) + \frac{1}{N}\sum_{t=1}^{N}\mu^2 = \tag{9}$$

$$= \frac{1}{N}\sum_{t=1}^{N}x(t)^2 - 2\mu\frac{1}{N}\sum_{t=1}^{N}x(t) + \frac{1}{N}N\mu^2 = \tag{10}$$

$$= \sigma_x^2 = \frac{1}{N}\sum_{t=1}^{N}x(t)^2 - \mu_x^2 \tag{11}$$

Note: this variance estimator is biased. To obtain an unbiased estimator, one must multiply the result by $N/(N-1)$.

An adaptive estimator for the variance is this one

$$\sigma_x(t)^2 = (1-UC)\cdot\sigma_x(t-1)^2 + UC\cdot(x(t)-\mu_x(t))^2 \tag{12}$$

Alternatively, one can also compute the adaptive mean square

$$MSQ_x(t) = (1-UC)\cdot MSQ_x(t-1) + UC\cdot x(t)^2 \tag{13}$$

and obtain the variance by

$$\sigma_x(t)^2 = MSQ_x(t) - \mu_x(t)^2 \tag{14}$$

When adaptive algorithms are used, we also need initial values and a suitable update coefficient. For the moment, it is sufficient to assume that initial values and the update coefficient are known. Various approaches to identify suitable values will be discussed later (see Sect. 1.5).

1.2.3 Variance-Covariance Estimation

In case of multivariate processes, also the covariances between the various dimensions are of interest. The (stationary) variance-covariance matrix (short covariance matrix) is defined as

$$cov(x) = \boldsymbol{\Sigma}_x = \frac{1}{N}\sum_{t=1}^{N}(\boldsymbol{x}(t)-\boldsymbol{\mu}_x)^T\cdot(\boldsymbol{x}(t)-\boldsymbol{\mu}_x) \tag{15}$$

whereby ^{T} indicates the transpose operator. The variances are the diagonal elements of the variance-covariance matrix, and the off-diagonal elements indicate the covariance $\sigma_{i,j} = \frac{1}{N}\sum_{t=1}^{N}\left((x_i(t) - \mu_i)\cdot(x_j(t) - \mu_j)\right)$ between the i-th and j-th element. We define also the so-called *extended covariance matrix* (ECM) $\boldsymbol{E}$ as

$$ECM(x) = \boldsymbol{E}_x = \sum_{t=1}^{N_x}[1, \boldsymbol{x}(t)]^T \cdot [1, \boldsymbol{x}(t)] = \left[\begin{array}{c|c} a & \boldsymbol{b} \\ \hline \boldsymbol{c} & \boldsymbol{D} \end{array}\right] = N_x \cdot \left[\begin{array}{c|c} 1 & \boldsymbol{\mu}_x \\ \hline \boldsymbol{\mu}_x{}^T & \boldsymbol{\Sigma}_x + \boldsymbol{\mu}_x^T \boldsymbol{\mu}_x \end{array}\right] \quad (16)$$

One can obtain from the ECM $\boldsymbol{E}$ the number of samples $N = a$, the mean $\boldsymbol{\mu} = \boldsymbol{b}/a$ as well as the variance-covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{D}/a - (\boldsymbol{c}/a)\cdot(\boldsymbol{b}/a)$. This decomposition will be used later.

The adaptive version of the ECM estimator is

$$\boldsymbol{E}_x(t) = (1 - UC)\cdot \boldsymbol{E}_x(t-1) + UC \cdot [1, \boldsymbol{x}(t)]^T \cdot [1, \boldsymbol{x}(t)] \quad (17)$$

where t is the sample time, UC is the update coefficient. The decomposition of the ECM $\boldsymbol{E}$, mean μ, variance σ^2 and covariance matrix $\boldsymbol{\Sigma}$ is the same as for the stationary case; typically is $N = a = 1$.

1.2.4 Adaptive Inverse Covariance Matrix Estimation

Some classifiers like LDA or QDA rely on the inverse $\boldsymbol{\Sigma}^{-1}$ of the covariance matrix $\boldsymbol{\Sigma}$; therefore, adaptive classifiers require an adaptive estimation of the inverse covariance matrix. The inverse covariance matrix $\boldsymbol{\Sigma}$ can be obtained from Eq. (16) with

$$\boldsymbol{\Sigma}^{-1} = a \cdot \left(\boldsymbol{D} - \boldsymbol{c}\cdot a^{-1}\cdot \boldsymbol{b}\right)^{-1}. \quad (18)$$

This requires an explicit matrix inversion. The following formula shows how the inverse convariance matrix $\boldsymbol{\Sigma}^{-1}$ can be obtained without an explicit matrix inversion. For this purpose, the block matrix decomposition [1] and the matrix inversion lemma (20) is used. Let us also define $iECM = \boldsymbol{E}^{-1} = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]^{-1}$ with $S = D - CA^{-1}B$. According to the block matrix decomposition [1]

$$\begin{aligned} \boldsymbol{E}_x^{-1} = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]^{-1} &= \left[\begin{array}{c|c} A^{-1} + A^{-1}BS^{-1}CA^{-1} & -A^{-1}BS^{-1} \\ \hline -S^{-1}CA^{-1} & S^{-1} \end{array}\right] \\ &= \left[\begin{array}{c|c} 1 + \boldsymbol{\mu}_x \boldsymbol{\Sigma}_x^{-1}\boldsymbol{\mu}_x & -\boldsymbol{\mu}_x^T \boldsymbol{\Sigma}_x^{-T} \\ \hline -\boldsymbol{\Sigma}_x^{-1}\boldsymbol{\mu}_x^T & \boldsymbol{\Sigma}_x^{-1} \end{array}\right] \end{aligned} \quad (19)$$

The inverse extended covariance matrix $iECM = \boldsymbol{E}^{-1}$ can be obtained adaptively by applying the matrix inversion lemma (20) to Eq. (17). The matrix inversion lemma (also know as Woodbury matrix identity) states that the inverse $\boldsymbol{A}^{-1}$ of a given matrix $\boldsymbol{A} = (\mathbf{B+UDV})$ can be determined by

$$\begin{aligned} \boldsymbol{A}^{-1} &= (\mathbf{B+UDV})^{-1} = 20 \\ &= \boldsymbol{B}^{-1} + \boldsymbol{B}^{-1}\boldsymbol{U}\left(\boldsymbol{D}^{-1} + \boldsymbol{V}\boldsymbol{B}^{-1}\boldsymbol{U}\right)^{-1}\boldsymbol{V}\boldsymbol{B}^{-1} \end{aligned} \tag{20}$$

To adaptively estimate the inverse of the *extended covariance matrix*, we identify the matrices in (20) as follows:

$$\boldsymbol{A} = \boldsymbol{E}(t) \tag{21}$$

$$\boldsymbol{B}^{-1} = (1 - UC) \cdot \boldsymbol{E}(t-1) \tag{22}$$

$$\boldsymbol{U}^T = \boldsymbol{V} = \boldsymbol{x}(t) \tag{23}$$

$$\boldsymbol{D} = UC \tag{24}$$

where UC is the update coefficient and $\boldsymbol{x}(t)$ is the current sample vector. Accordingly, the inverse of the covariance matrix is:

$$\boldsymbol{E}(t)^{-1} = \frac{1}{(1-UC)} \cdot \left(\boldsymbol{E}(t-1)^{-1} - \frac{1}{\frac{1-UC}{UC} + \boldsymbol{x}(t)\cdot\boldsymbol{v}} \cdot \boldsymbol{v}\cdot\boldsymbol{v}^T\right) \tag{25}$$

with $\boldsymbol{v} = \boldsymbol{E}(t-1)^{-1} \cdot \boldsymbol{x}(t)^T$. Since the term $\boldsymbol{x}(t)\cdot\boldsymbol{v}$ is a scalar, and no explicit matrix inversion is needed.

In practice, this adaptive estimator can become numerically unstable (due to numerical inaccuracies, the iECM can become asymmetric and singular). This numerical problem can be avoided if the symmetry is enforced, e.g. in the following way:

$$\boldsymbol{E}(t)^{-1} = \left(\boldsymbol{E}(t)^{-1} + \boldsymbol{E}(t)^{-1,T}\right)/2 \tag{26}$$

Now, the inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$ can be obtained by estimating the extended covariance matrix with Eq. (25) and decomposing it according to Eq. (19).

Kalman Filtering and the State Space Model

The aim of a BCI is to identify the state of the brain from the measured signals. The measurement itself, e.g. some specific potential difference at some electrode, is not the "brain state" but the result of some underlying mechanism generating different patterns depending on the state (e.g. alpha rhythm EEG). Methods that try to identify the underlying mechanism are called system identification or model identification methods. There are a large number of different systems and different methods in this area. In the following, we'll introduce an approach to identify a state-space model (Fig. 2). A state-space model is a general approach and can be used to describe a large number of different models. In this chapter, an autogregressive model and a linear discriminant model will be used, but a state-state space model can be also used to describe more complex models. Another useful advantage, besides the general

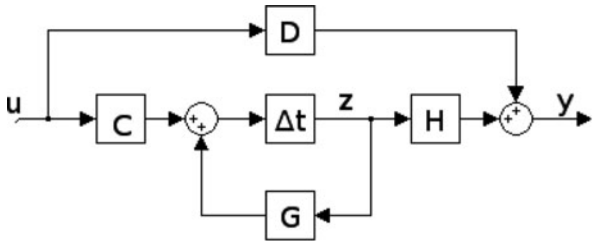


Fig. 2 State Space Model. G is the state transition matrix, H is the measurement matrix, Δt denotes a one-step time delay, C and D describe the influence of some external input u to the state vector and the output y, respectively. The system noise w and observation noise v are not shown

nature of the state-space model, is the fact that efficient adaptive algorithms are available for model identification. This algorithm is called the Kalman filter.

Kalman [12] and Bucy [13] presented the original idea of Kalman filtering (KFR). Meinhold et al. [20] provided a Bayesian formulation of the method. Kalman filtering is an algorithm for estimating the state (vector) of a state space model with the system Eq. (27) and the measurement (or observation) Eq. (28).

$$z(t) = \boldsymbol{G}(t,t-1) \cdot z(t-1) + \boldsymbol{C}(t) \cdot \boldsymbol{u}(t) + \boldsymbol{w}(t) \tag{27}$$

$$\boldsymbol{y}(t) = \boldsymbol{H}(t) \cdot z(t) + \boldsymbol{D}(t) \cdot \boldsymbol{u}(t) + \boldsymbol{v}(t) \tag{28}$$

$\boldsymbol{u}(t)$ is an external input. When identifying the brain state, we usually ignore the external input. Accordingly $\boldsymbol{C}(t)$ and $\boldsymbol{D}(t)$ are zero, while $z(t)$ is the state vector and depends only on the past values of $\boldsymbol{w}(t)$ and some initial state z_0. The observed output signal $\boldsymbol{y}(t)$ is a combination of the state vector and the measurement noise $\boldsymbol{v}(t)$ with zero mean and variance $\boldsymbol{V}(t) = \boldsymbol{E}\langle \boldsymbol{v}(t) \cdot \boldsymbol{v}(t)^T \rangle$. The process noise $\boldsymbol{w}(t)$ has zero mean and covariance matrix $\boldsymbol{W}(t) = \boldsymbol{E}\langle \boldsymbol{w}(t) \cdot \boldsymbol{w}(t)^T \rangle$. The state transition matrix $\boldsymbol{G}(t,t-1)$ and the measurement matrix $\boldsymbol{H}(t)$ are known and may or may not change with time.

Kalman filtering is a method that estimates the state $z(t)$ of the system from measuring the output signal $y(t)$ with the prerequisite that $\boldsymbol{G}(t,t-1)$, $\boldsymbol{H}(t)$, $\boldsymbol{V}(t)$ and $\boldsymbol{W}(t)$ for $t > 0$ and z_0 are known. The inverse of the state transition matrix $\boldsymbol{G}(t,t-t)$ exists and $\boldsymbol{G}(t,t-1) \cdot \boldsymbol{G}(t-1,t) = \boldsymbol{I}$ is the unity matrix $\boldsymbol{I}$. Furthermore, $\boldsymbol{K}(t,t-1)$, the a-priori state-error correlation matrix, and $\boldsymbol{Z}(t)$, the a posteriori state-error correlation matrix are used; $\boldsymbol{K}_{1,0}$ is known. The Kalman filter equations can be summarized in this algorithm

$$\begin{aligned}
\boldsymbol{e}(t) &= \boldsymbol{y}(t) - \boldsymbol{H}(t) \cdot \hat{z}(t) \\
\hat{z}(t+1) &= \boldsymbol{G}(t,t-1) \cdot \hat{z}(t) + \boldsymbol{k}(t-1) \cdot \boldsymbol{e}(t) \\
\boldsymbol{Q}(t) &= \boldsymbol{H}(t) \cdot \boldsymbol{K}(t,t-1) \cdot \boldsymbol{H}^T(t) + \boldsymbol{V}(t) \\
\boldsymbol{k}(t) &= \boldsymbol{G}(t,t-1) \cdot \boldsymbol{K}(t,t-1) \cdot \boldsymbol{H}^T(t)/\boldsymbol{Q}(t) \\
\boldsymbol{Z}(t) &= \boldsymbol{K}(t,t-1) - \boldsymbol{G}(t-1,t) \cdot \boldsymbol{k}(t) \cdot \boldsymbol{H}(t) \cdot \boldsymbol{K}(t,t-1) \\
\boldsymbol{K}(t+1,t) &= \boldsymbol{G}(t,t-1) \cdot \boldsymbol{Z}(t) \cdot \boldsymbol{G}(t,t-1)^T + \boldsymbol{W}(t)
\end{aligned} \tag{29}$$

Using the next observation value $\boldsymbol{y}(t)$, the one-step prediction error $\boldsymbol{e}(t)$ can be calculated using the current estimate $\hat{z}(t)$ of the state $z(t)$, and the state vector $z(t+1)$ is updated (29). Then, the estimated prediction variance $\boldsymbol{Q}(t)$ that can be calculated which consists of the measurement noise $\boldsymbol{V}(t)$ and the error variance due to the estimated state uncertainty $\boldsymbol{H}(t) \cdot \boldsymbol{K}(t, t-1) \cdot \boldsymbol{H}(t)^T$. Next, the Kalman gain $\boldsymbol{k}(t)$ is determined. Finally, the a posteriori state-error correlation matrix $\boldsymbol{Z}(t)$ and the a-priori state-error correlation matrix $\boldsymbol{K}(t+1, t)$ for the next iteration are obtained.

Kalman filtering was developed to estimate the trajectories of spacecrafts and satellites. Nowadays, Kalman filtering is used in a variaty of applications, including autopilots, economic time series prediction, radar tracking, satellite navigation, weather forecasts, etc.

1.3 Feature Extraction

Many different features can be extracted from EEG time series, like temporal, spatial, spatio-temporal, linear and nonlinear parameters [8, 19]. The actual features extracted use first order statistical properties (i.e. time-varying mean like the slow cortical potential [2]), or more frequently the second order statistical properties of the EEG are used by extracting the frequency spectrum, or the autoregressive parameters [18, 31, 36]. Adaptive estimation of the mean has been discussed in Sect. 1.2. Other time-domain parameters are activity, mobility and complexity [10], amplitude, frequency, spectral purity index [11], Sigma et al. [49] and brainrate [28]. Adaptive estimators of these parameters have been implemented in the open source software library Biosig [30].

Spatial correlation methods are PCA, ICA and CSP [3, 4, 16, 29]; typically these methods provide a spatial decomposition of the data and do not take into account a temporal correlation. Recently, extensions of CSP have been proposed that can construct spatio-temporal filters [4, 7, 16]. To address non-stationarities, covariate shift compensation approaches [38, 40, 41] have been suggested and adaptive CSP approaches have been proposed [42, 43]. In order to avoid the computational expensive eigendecomposition after each iteration, an adaptive eigenanalysis approaches as suggested in [21, 39] might be useful.

Here, the estimation of the adaptive autoregressive (AAR) parameters is discussed in greater depth. AAR parameters can capture the time-varying second order statistical moments. Almost no a priori knowledge is required, the model order p is not very critical and, since it is a single coefficient, it can be easily optimized. Also, no expensive feature selection algorithm is needed. AAR parameters provide a simple and robust approach, and hence provide a good starting point for adaptive feature extraction.

1.3.1 Adaptive Autoregressive Modeling

A univariate and stationary autoregressive (AR) model is described by any of the following equations

$$\begin{aligned} y_k &= a_1 \cdot y_{k-1} + \cdots + a_p \cdot y_{k-p} + x_k = \\ &= \sum_{i=1}^{p} a_i \cdot y_{k-i} + x_k = \\ &= [y_{k-1}, \ldots, y_{k-p}] \cdot [a_1, \ldots a_p]^T + x_k = \\ &= \boldsymbol{Y}_{k-1} \cdot \boldsymbol{a} + x_k 30 \end{aligned} \tag{30}$$

with innovation process $x_k = N(\mu_x = 0, \sigma_x^2)$ having zero mean and variance σ_x^2. For a sampling rate f_0, the spectral density function $P_y(f)$ of the AR process y_k is

$$P_y(f) = \frac{\sigma_x^2/(2\pi f_0)}{|1 - \sum_{k=1}^{p}(a_i \cdot exp^{-jk2\pi f/f_0})|^2} \tag{31}$$

There are several estimators (Levinson-Durbin, Burg, Least Squares, geometric lattice) for the stationary AR model. Estimation of adaptive autoregressive (AAR) parameters can be obtained with Least Mean Squares (LMS), Recursive Least Squares (RLS) and Kalman filters (KFR) [36]. LMS is a very simple algorithm, but typically performs worse (in terms of adaptation speed and estimation accuracy) than RLS or KFR [9, 31, 36]. The RLS method is a special case of the more general Kalman filter approach. To perform AAR estimation with the KFR approach, the AAR model needs to be adapted – in a suitable way – to the state space model.

The aim is to estimate the time-varying autoregressive parameters; therefore, the AR parameters become state vectors $\boldsymbol{z}_k = \boldsymbol{a}_k = [a_{1,k}, \cdots, a_{p,k}]^T$. Assuming that the AR parameters follow a mutltivariate random walk, the state transition matrix becomes the identity matrix $\boldsymbol{G}_{k,k-1} = \boldsymbol{I}_{p\times p}$ and the system noise w_k allows for small alterations. The observation matrix $\boldsymbol{H}_k$ consists of the past p sampling values $y_{k-1}, \ldots, y_{k-p}$. The innovation process $v_k = x_k$ with $\sigma_x^2(k) = V_k$. The AR model (30) is translated into the state space model formalism (27-28) as follows:

$$\begin{aligned} &\textit{State Space Model} \Leftrightarrow \textit{Autoregressive Model} \\ \boldsymbol{z}_k &= \boldsymbol{a}_k = [a_{1,k}, \ldots a_{p,k}]^T \\ \boldsymbol{H}_k &= \boldsymbol{Y}_{k-1} = [y_{k-1}, \ldots, y_{k-p}]^T \\ \boldsymbol{G}_{k,k-1} &= \boldsymbol{I}_{p\times p} \\ \boldsymbol{V}_k &= \sigma_x^2(k) \\ \boldsymbol{Z}_k &= E\langle(\boldsymbol{a}_k - \hat{a}_k)^T \cdot (\boldsymbol{a}_k - \hat{\boldsymbol{a}}_k)\rangle \\ \boldsymbol{W}_k &= \boldsymbol{A}_k = E\langle(\boldsymbol{a}_k - \boldsymbol{a}_{k-1})^T \cdot (\boldsymbol{a}_k - \boldsymbol{a}_{k-1})\rangle \\ \boldsymbol{v}_k &= x_k \end{aligned} \tag{32}$$

Accordingly, the Kalman filter algorithm for the AAR estimates becomes

$$\begin{aligned} e_k &= y_k - \boldsymbol{Y}_{k-1} \cdot \hat{\boldsymbol{a}}_{k-1} \\ \hat{\boldsymbol{a}}(k) &= \hat{\boldsymbol{a}}_{k-1} + \boldsymbol{k}_{k-1} \cdot e_k \\ \boldsymbol{Q}_k &= \boldsymbol{Y}_k \cdot \boldsymbol{A}_{k-1} \cdot \boldsymbol{Y}_k^T + V_k \\ \boldsymbol{k}_k &= A_{k-1} \cdot \boldsymbol{Y}_{k-1}/\boldsymbol{Q}_k \\ \boldsymbol{Z}_k &= \boldsymbol{A}_{k-1} - \boldsymbol{k}_k \cdot \boldsymbol{Y}_k^T \cdot A_{k-1} \\ \boldsymbol{A}_k &= \boldsymbol{Z}_k + \boldsymbol{W}_k \end{aligned} \tag{33}$$

$\boldsymbol{W}_k$ and $\boldsymbol{V}_k$ are not determined by the Kalman equations, but must be known. In practice, some assumptions must be made which result in different algorithms

[36]. For the general case of KFR, the equation with explicit $\boldsymbol{W}_k$ is used $\boldsymbol{A}_k = \boldsymbol{Z}_k + \boldsymbol{W}_k$ with $\boldsymbol{W}_k = q_k \cdot I$. In the results with KFR-AAR below, we used $q_k = UC \cdot trace(\boldsymbol{A}_{k-1})/p$. The RLS algorithm is characterized by the fact that $\boldsymbol{W}_k = \boldsymbol{UC} \cdot \boldsymbol{A}_{k-1}$. Numerical inaccuracies can cause instabilities in the RLS method [36]; these can be avoided by enforcing a symmetric state error correlation matrix $\boldsymbol{A}_k$. For example, Eq. (33) can be choosen as $\boldsymbol{A}_k = (1 + UC) \cdot \left((\boldsymbol{Z}_k + \boldsymbol{Z}_k^T\right)/2$. The AAR parameters calculated using this algorithm are referred as RLS-AAR. In the past, KFR was usually used for stability reasons. With this new approach, RLS-AAR performs best among the various AAR estimation methods, as shown by results below.

Typically, the variance of the prediction error $\boldsymbol{V}_k = (1 - UC) \cdot \boldsymbol{V}_{k-1} + UC \cdot e_k^2$ is adaptively estimated from the prediction error (33) according to Eq. (12).

Kalman filters require initial values, namely the initial state estimate $z_0 = \boldsymbol{a}_0$, the initial state error correlation matrix $\boldsymbol{A_0}$ and some guess for the variance of innovation process $\boldsymbol{V}_0$. Typically, a rough guess might work, but can also yield a long lasting initial transition effect. To avoid such a transition effect, a more sensible approach is recommended. A two pass approach was used in [33]. The first pass was based on some standard initial values, these estimates were used to obtain the initial values for the second pass $\boldsymbol{a}_0 = \boldsymbol{\mu}_a$, $\boldsymbol{A_0} = cov(\boldsymbol{a}_k)$, $\boldsymbol{V}_0 = var(e_k)$. Moreover, $\boldsymbol{W}_k = \boldsymbol{W} = cov(\boldsymbol{\alpha}_k)$ with $\boldsymbol{\alpha}_k = \boldsymbol{a}_k - \boldsymbol{a}_{k-1}$ can be used for the KFR approach.

For an adaptive spectrum estimation (31), not only the AAR parameters, but also the variance of the innovation process $\sigma_x^2(k) = V_k$ is needed. This suggests that the variance can provide additional information. The distribution of the variance is χ^2-distribution. In case of using linear classifiers, this feature should be "linearized" (typically with a logarithmic transformation). Later, we will show some experimental results comparing AAR features estimates with KFR and RLS. We will further explore whether including variance improves the classification.

1.4 Adaptive Classifiers

In BCI research, discriminant based classifiers are very popular because of their simplicity and the low number of parameters needed for their computation. For these reasons they are also attractive candidates for on-line adaptation. In the following, linear (LDA) and quadratic (QDA) discriminant analysis are discussed in detail.

1.4.1 Adaptive QDA Estimator

The classification output $D(\boldsymbol{x})$ of a QDA classifier in a binary problem is obtained as the difference between the square root of the Mahalanobis distance to the two classes i and j as follows:

$$D(\boldsymbol{x}) = d_{\{j\}}(\boldsymbol{x}) - d_{\{i\}}(\boldsymbol{x}) \tag{34}$$

where the Mahalanobis distance is defined as:

$$d_{\{i\}}(\boldsymbol{x}) = ((\boldsymbol{x} - \boldsymbol{\mu}_{\{i\}})^T \cdot \boldsymbol{\Sigma}_{\{i\}}^{-1} \cdot (\boldsymbol{x} - \boldsymbol{\mu}_{\{i\}}))^{1/2} \tag{35}$$

where $\boldsymbol{\mu}_{\{i\}}$ and $\boldsymbol{\Sigma}_{\{i\}}$ are the mean and the covariance, respectively, of the class samples from class i. If $D(\boldsymbol{x})$ is greater than 0, the observation is classified as class i and otherwise as class j. One can think of a minimum distance classifier, for which the resulting class is obtained by the smallest Mahalanobis distance $argmin_i(d_{\{i\}}(\boldsymbol{x}))$. As seen in Eq. (35), the inverse covariance matrix (16) is required. Writing the mathematical operations in Eq. (35) in matrix form yields:

$$d_{\{i\}}(\boldsymbol{x}) = ([1;\ \boldsymbol{x}] \cdot \boldsymbol{F}_{\{i\}} \cdot [1;\ \boldsymbol{x}]^T)^{1/2} \tag{36}$$

with

$$\boldsymbol{F}_{\{i\}} = \left[\frac{-\boldsymbol{\mu}_{\{i\}}^T}{\boldsymbol{I}}\right] \cdot \boldsymbol{\Sigma}_{\{i\}}^{-1} \cdot \left[-\boldsymbol{\mu}_i \middle| \boldsymbol{I}\right] = \left[\begin{array}{c|c} \boldsymbol{\mu}_{\{i\}}^T \boldsymbol{\Sigma}_{\{i\}}^{-1} \boldsymbol{\mu}_{\{i\}} & -\boldsymbol{\mu}_{\{i\}}^T \boldsymbol{\Sigma}_{\{i\}}^{-T} \\ \hline -\boldsymbol{\Sigma}_{\{i\}}^{-1} \boldsymbol{\mu}_{\{i\}} & \boldsymbol{\Sigma}_{\{i\}}^{-1} \end{array}\right] \tag{37}$$

Comparing Eq. (19) with (37), we can see that the difference between $\boldsymbol{F}_{\{i\}}$ and $\boldsymbol{E}_{\{i\}}^{-1}$ is just a 1 in the first element of the matrix, all other elements are equal. Accordingly, the time-varying Mahalanobis distance of a sample $\boldsymbol{x}(t)$ to class i is

$$d_{\{i\}}(\boldsymbol{x}_k) = \left\{[1, \boldsymbol{x}_k] \cdot \left(\boldsymbol{E}_{\{i\},k}^{-1} - \begin{bmatrix} 1 & 0_{1\times M} \\ 0_{M\times 1} & 0_{M\times M} \end{bmatrix}\right) \cdot [1, \boldsymbol{x}_k]^T\right\}^{1/2} \tag{38}$$

where $\boldsymbol{E}_{\{i\}}^{-1}$ can be obtained by Eq. (25) for each class i.

1.4.2 Adaptive LDA Estimator

Linear discriminant analysis (LDA) has linear decision boundaries. This is the case when the covariance matrices of all classes are equal; that is, $\boldsymbol{\Sigma}_{\{i\}} = \boldsymbol{\Sigma}$ for all classes i. Then, all observations are distributed in hyperellipsoids of equal shape and orientation, and the observations of each class are centered around their corresponding mean $\boldsymbol{\mu}_{\{i\}}$. The following equation is used in the classification of a two-class problem:

$$D(\boldsymbol{x}) = \boldsymbol{w} \cdot (\boldsymbol{x} - \boldsymbol{\mu}_x)^T = [b, \boldsymbol{w}] \cdot [1, \boldsymbol{x}]^T \tag{39}$$

$$\boldsymbol{w} = \Delta\boldsymbol{\mu} \cdot \boldsymbol{\Sigma}^{-1} = (\boldsymbol{\mu}_{\{i\}} - \boldsymbol{\mu}_{\{j\}}) \cdot \boldsymbol{\Sigma}^{-1} \tag{40}$$

$$\boldsymbol{b} = -\boldsymbol{\mu}_x \cdot \boldsymbol{w}^T = -\frac{1}{2} \cdot (\boldsymbol{\mu}_{\{i\}} + \boldsymbol{\mu}_{\{j\}}) \cdot \boldsymbol{w}^T \tag{41}$$

where $D(\boldsymbol{x})$ is the difference in the distance of the feature vector $\boldsymbol{x}$ to the separating hyperplane described by its normal vector $\boldsymbol{w}$ and the bias b. If $D(\boldsymbol{x})$ is greater than 0, the observation x is classified as class i and otherwise as class j.

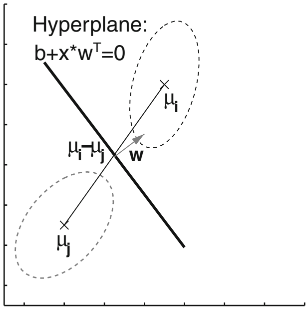
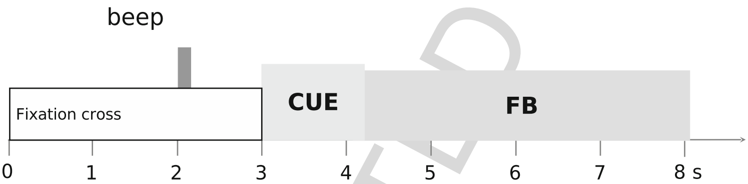


Fig. 3 Concept of classification with LDA. The two classes are reprensented by two ellipsoids (the covariance matrices) and the respective class mean values. The hyperplane is the boundary of decision, with $D(\boldsymbol{x}) = b + \boldsymbol{x} \cdot \boldsymbol{W}^T = 0$. A new observation $\boldsymbol{x}$ is classified as follows: if $D(\boldsymbol{x})$ is greater than 0, the observation x is classified as class i and otherwise as class j. The normal vector to the hyperplane, $\boldsymbol{w}$, is in general not in the direction of the difference between the two class means



heightheight

Fig. 4 Paradigm of cue-based BCI experiment. Each trial lasted 8.25 s. A cue was presented at $t = 3s$, feedback was provided from **t**=4.25 to 8.25 s

AQ4 The methods to adapt LDA can be divided in two different groups. First, using the estimation of the covariance matrices of the data, for which the speed of adaption is fixed and determined by the update coefficient. The second group is based on Kalman Filtering and has the advantage of having a variable adaption speed depending on the properties of the data.

Fixed Rate Adaptive LDA Using (19), it can be shown that the distance function (Eq. 39) is

$$D(\boldsymbol{x}_k) = [b_k, \boldsymbol{w}_k] \cdot [1, \boldsymbol{x}_k]^T \tag{42}$$

$$= b_k + \boldsymbol{w}_k \cdot \boldsymbol{x}_k^T \tag{43}$$

$$= -\Delta\boldsymbol{\mu}_k \cdot \boldsymbol{\Sigma}_k^{-1} \cdot \boldsymbol{\mu}_k^T + \Delta\boldsymbol{\mu}_k \cdot \boldsymbol{\Sigma}_k^{-1} \cdot \boldsymbol{x}_k^T \tag{44}$$

$$= [0, \boldsymbol{\mu}_{\{i\},k} - \boldsymbol{\mu}_{\{j\},k}] \cdot \boldsymbol{E}_k^{-1} \cdot [1, \boldsymbol{x}_k] \tag{45}$$

with $\Delta\boldsymbol{\mu}_k = \boldsymbol{\mu}_{\{i\},k} - \boldsymbol{\mu}_{\{j\},k}$, $b = -\Delta\boldsymbol{\mu}(t) \cdot \boldsymbol{\Sigma}(t)^{-1} \cdot \boldsymbol{\mu}(t)^T$ and $\boldsymbol{w} = \Delta\boldsymbol{\mu}(t) \cdot \boldsymbol{\Sigma}^{-1}$.

Accordingly, the adaptive LDA can be estimated with Eq. (45) using (25) for estimating $\boldsymbol{E}_k^{-1}$ and (5) for estimating the class-specific adaptive mean $\boldsymbol{\mu}_{\{i\},k}$ and $\boldsymbol{\mu}_{\{j\},k}$. The adaptation speed is determined by the update coefficient UC used in the Eq. (5) and (25). For a constant update coefficient, the adaptation rate is also constant.

Variable Rate Adaptive LDA This method is based in Kalman Filtering and its speed of adaptation depends on the Kalman Gain, shown in Eq. (29), which varies with the properties of the data. The state space model for the classifier case is summarized in (46), where c_k is the current class label, z_k are the classifier weights, the measurement matrix $\boldsymbol{H}_k$ is the feature vector with a one added in the front $[1;\ \boldsymbol{x}_k]$, and $D_k(\boldsymbol{x})$ is the classification output.

$$
\begin{aligned}
&\mathit{State\ Space\ Model} \Leftrightarrow \mathit{LinearCombiner}\\
\boldsymbol{y}_k &= c_k\\
\boldsymbol{z}_k &= [b_k, \boldsymbol{w}_k]^T\\
\boldsymbol{H}_k &= [1; \boldsymbol{x}_k]^T\\
\boldsymbol{G}_{k,k-1} &= \boldsymbol{I}_{M\times M}\\
\boldsymbol{Z}_k &= E\langle(\boldsymbol{w}_k - \hat{w}_k)^T \cdot (\boldsymbol{w}_k - \hat{w}_k)\rangle\\
\boldsymbol{W}_k &= \boldsymbol{A}_k = E\langle(\boldsymbol{w}_k - \boldsymbol{w}_{k-1})^T \cdot (\boldsymbol{w}_k - w_{k-1})\rangle\\
\boldsymbol{e}_k &= D_k(\boldsymbol{x}) - c_k
\end{aligned}
\tag{46}
$$

Then, the Kalman filter algorithm for the adaptive LDA classifier is

$$
\begin{aligned}
e_k &= y_k - \boldsymbol{H}_k \cdot \boldsymbol{z}_{k-1}^T 47\\
z_k &= z_{k-1} + \boldsymbol{k}_k \cdot e_k\\
Q_k &= \boldsymbol{H}_k \cdot \boldsymbol{A}_{k-1} \cdot \boldsymbol{H}_k^T + V_k\\
\boldsymbol{k}_k &= A_{k-1} \cdot \boldsymbol{H}_k^T / Q_k\\
\boldsymbol{Z}_k &= \boldsymbol{A}_{k-1} - \boldsymbol{k}_k \cdot \boldsymbol{H}_k^T \cdot A_{k-1}\\
\boldsymbol{A}_k &= \boldsymbol{Z}_k + \boldsymbol{W}_k 47
\end{aligned}
\tag{47}
$$

The variance of the prediction error $\boldsymbol{V}_k$ was estimated adaptively from the prediction error (47) according to Eq. (12). The RLS algorithm was used to estimate $\boldsymbol{A}_k$.

As the class labels are bounded between 1 and -1, it would be convenient to also bound the product $\boldsymbol{H}_k \cdot z_{k-1}^T$ between these limits. Hence, a transformation in the estimation error can be applied, but then the algorithm is not a linear filter anymore:

$$
e_k = y_k + 1 - \frac{2}{(1 + \exp(-\boldsymbol{H}_k \cdot z_{k-1}^T))} \tag{48}
$$

1.5 Selection of Initial Values, Update Coefficient and Model Order

All adaptive algorithms need some initial values and must select some update coefficients. Some algorithms like adaptive AAR need also a model order p. Different approaches are available to select these parameters. The initial values can be often obtained by some a priori knowledge. Either it is known that the data has zero mean (e.g. because it is low pass filtered), or a reasonable estimate can be obtained from previous recordings, or a brief segment in the beginning of the record is used to estimate the initial values. If nothing is known, it is also possible to use some random initial values (e.g. zero) and wait until the adaptive algorithm eventually converges to the proper range. For a state space model [9], we recommend starting with a diagonal matrix weighted by the variance of previous data and multiplied by a factor δ, which can be very small or very large $\boldsymbol{\Sigma}_0 = \delta\sigma^2\boldsymbol{I}$.

Of course one can also apply more sophisticated methods. For example, to apply the adaptive classifier to new (untrained) subjects, a general classifier was estimated from data of seven previous records from different subjects. This had the advantage that no laborious training sessions (i.e. without feedback) were needed, but the new subjects could work immediately with BCI feedback. Eventually, the adaptive classifier adapted to the subject specific pattern [44–48].

A different approach was used in an offline study using AAR parameters [33]. Based on some preliminary experiments, it became obvious that setting the initial values of the AAR parameters to zero can have some detrimental influence on the result. The initial transient took several trials, while the AAR parameters were very different than the subsequent trials. To avoid this problem, we applied the AAR estimation algorithm two times. The first run was initialized by $\vec{a}_0 = [0, ..., 0]$, $\boldsymbol{A}_0 = I_{pp}$, $V_k = 1 - UC$, $\boldsymbol{W}_k = \boldsymbol{I} \cdot UC \cdot trace(\boldsymbol{A}_{k-1})/p$. The resulting AAR estimates were used to estimate more reasonable initial values $\vec{a}_0 = mean(\vec{a}_t)$, $\boldsymbol{A}_0 = cov\vec{a}_t$, $\boldsymbol{V}_k = vare_t$, $\boldsymbol{W}_k = cov\Delta\vec{a}_t$ with $\Delta\vec{a}_t = \vec{a}_t - \vec{a_{t-1}}$.

The selection of the update coefficient is a trade-off between adaptation speed and estimation accuracy. In case of AAR estimation in BCI data, a number of results [31, 35, 36] suggest, that there is always a global optimum to select the optimum update coefficient, which makes it rather easy to identify a reasonable update coefficient based on some representative data sets. In case of adaptive classifiers, it is more difficult to identify a proper update coefficient from the data; therefore we determined the update coefficient based on the corresponding time constant. If the classifier should be able to adapt to a new pattern within 100 trials, the update coefficient was chosen such that the corresponding time constant was about 100 trials.

The order p of the AAR model is another free parameter that needs to be determined. Traditional model order selection criteria like the Akaike Information Criterion (AIC) and similar ones are based on stationary signal data, which is not the case for AAR parameters. Therefore, we have developed a different approach to select the model order which is based on the one-step prediction error [35, 36] of the AAR model. These works were mostly motivitated by the principle of uncertainty between time and frequency domain suggesting model orders in the range from 9 to 30. Unfortunately, the model order obtained with this approach was not necessarily the best for single trial EEG classification like in BCI data, often much smaller orders gave much better results. We have mostly used model orders of 6 [27, 31, 34] and 3 [33, 44, 45, 47]. These smaller orders are prefered by the classifiers, when the number of trials used for classification is rather small. A simple approach is the use the rule of thumb that the nunber of features for the classifier should not exceed a 1/10 of the number of trials. So far the number of studies investigating the most suitable strategy for selecting model order, update coefficient and initial values are rather limited, future studies will be needed to address this open issues.

1.6 Experiments with Adaptive QDA and LDA

Traditional BCI experiments use a block-based design for training the classifiers. This means that some data must be recorded first before a classifier can be estimated; and the classifier can be only modified after a "run" (which is typically about 20 or 40 trials) is completed. Typically, this procedure also involve a manual decision whether the previous classifier should be replaced by the new one or not. Adaptive classifiers overcome this limitation, because the classifier is updated with every trial. Accordingly, an adaptive classifier can react much faster to a change in recording conditions, or when the subject modifies its brain patterns. The aim of the study was to investigate whether such adaptive classifiers can be applied in practical BCI experiments.

Experiments were carried out with 21 able-bodied subjects without previous BCI experience. They performed experiments using the "basket paradigm" [15]. At the bottom of a computer screen, a so-called basket was presented either on the left side or the right side of the screen. A ball moved from the top of the screen to the bottom at a constant velocity. During this time (typically 4 s), the subject controls the horizontal (left-right) position with the BCI system. The task was to control the horizontal position of the ball to move the ball into the displayed basket. Each subject conducted three different sessions, with 9 runs per session and 40 trials per run. 1080 trials were available for each of them (540 trials for each class). Two bipolar channels, C3 and C4, were recorded.

The system was a two-class cue-based and EEG-based BCI, and the subjects had to perform motor imagery of the left or right hand depending on the cue. More specifically, they were not instructed to imagine any specific movement, but they were free to find their own strategy. Some of them reported that the imagination of movements that involve several parts of the arm were more successful. In any case, they were asked to maintain their strategy for at least one run.

In the past, KFR-AAR was the best choice because it was a robust and stable method; other methods were not stable and required periodic reinitialization. With the enforcing of a symetric system matrix (Eq. 1.3), RLS could be stabilized. Moreover, based on the compostion of AR spectra, it seems reasonable to also include the variance of the innovation process as a feature. To compare these methods, Kalman based AAR parameters (KFR-AAR) (model order $p = 6$), RLS-AAR ($p = 6$) parameters, RLS-AAR ($p = 5$) combined with the logarithm of the variance (RLS-AAR+V) and the combination of RLS-AAR($p = 4$) and band power estimates (RLS-AAR+BP) are compared. The model order p was varied to maintain 6 features per channels. The classifier was LDA without adaptation, and

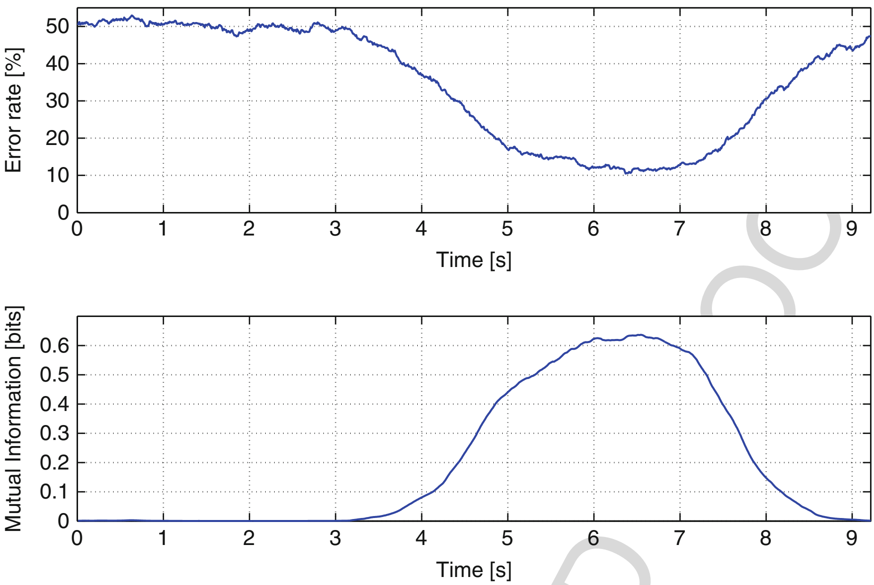


Fig. 5 Time course of the performance measurements. These changes are caused by the short-term nonstationarities of the features; the classifier was constant within each trial

a leave-one-out cross-validation procedure was used selecting the order of the AR model.

The performance results were obtained by single trial analysis of the online classification output of each experimental session. For one data set, the time courses of the error rate and the mutual information (MI are shown in Fig. 5). The mutual information *MI* is a measure the transfered information and is defined $MI = 0.5 \cdot log_2(1+SNR)$ with the signal-to-noise ratio $SNR = \frac{\sigma^2_{\text{signal}}}{\sigma^2_{\text{noise}}}$. The signal part of the BCI output is the variance from the class-related differences, and the noise part is the variance of the background activity described by the variability within one class. Accordingly, the mutual information can be determined from the total variability (variance of the BCI output among all classes), and the average within-class variability by [32, 34, 36]

$$MI = 0.5 \cdot log_2 \frac{\sigma^2_{\text{total}}}{\sigma^2_{\text{within}}} \tag{49}$$

For further comparison, the minimum error rate and the maximum mutual information are used. Figure 6 depicts the performance scatter plots of different AAR-based features against KFR-AAR. The first row shows ERR and the second MI values. For ERR (MI), all values under the diagonal show the superiority of the

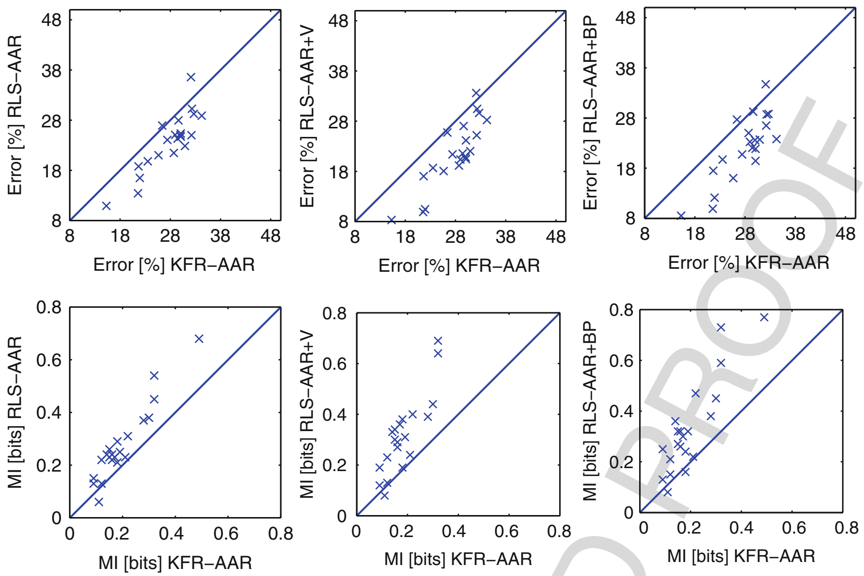


Fig. 6 The first row shows the scatter plots of error rates of different adaptive feature types using an LDA classifier and cross-validation based on leave-one (trial)-out procedure. The second row are scatter plots for mutual information results. The used methods are (i) bandpower values for the bands 8–14 and 16–28 Hz (BP) with a 1-second rectangular sliding window, (ii) AAR estimates using Kalman filtering (KFR), (iii) AAR estimates using the RLS algorithm, RLS-based AAR estimates combined with the logarithm of the variance V (AAR+V), and RLS-based AAR estimates combined with bandpower (AAR+BP). In the first row, values below the diagonal show the superiority of the method displayed in the y-axis. In the second row (MI values), the opposite is true. This figure shows that all methods outperform AAR-KFR

method displayed in the y-axis. Looking at these scatter plots, one can see that KFR-AAR is clearly inferior to all other methods. For completition of the results, and to compare the performance of each feature type, the mean value and standard error of each investigated feature were computed and presented in Table 2.

Table 2 Summary results from 21 subjects. The mean and standard error of the mean (SEM) of minimum ERR and maximum MI are shown. The AAR-based results are taken from the results shown in Fig. 6. Additionally, results from standard bandpower (10–12 and 16–24 Hz) and the bandbower of selected bands (8–14 and 16–28 Hz) are included

Feature	ERR[%]	MI[bits]
BP-standard	26.16±1.90	0.258±0.041
BP	25.76±1.86	0.263±0.041
KFR-AAR	27.85±1.04	0.196±0.021
RLS-AAR	23.73±1.27	0.277±0.031
RLS-AAR+V	21.54±1.45	0.340±0.041
RLS-AAR+BP	22.04±1.44	0.330±

The results displayed in Table 2, with a threshold p-value of 1.7%, show similar performance in ERR and MI for RLS-AAR+V and RLS-AAR+BP; both were found significantly better than KFR-AAR and RLS-AAR. Also RLS-AAR was significantly better than KFR-AAR. The bandpower values are better then the Kalman filter AAR estimates, but are worse compared to RLS-AAR and RLS-AAR+V.

Using the features RLS-AAR+V, we tested then several adaptive classifiers. To simulate a causal system, the time point when the performance of the system was measured was previously fixed, and the ERR and MI calculated at these previously defined time-points. The set of parameters for the classifiers in the first session were common to all subjects and computed from previously recorded data from 7 subjects during various feedback sessions [26]. The set of parameters in the second session were found by subject specific optimization of the data of the first session. The same procedure was used for the parameters selected for the third session.

Table 3 shows that all adaptive classifiers outperform the no adaptation setting. The best performing classifier was aLDA, which outperformed the Adaptive QDA and Kalman LDA. Kalman LDA also was found statistically better than Adaptive QDA.

This figure will be printed in b/w

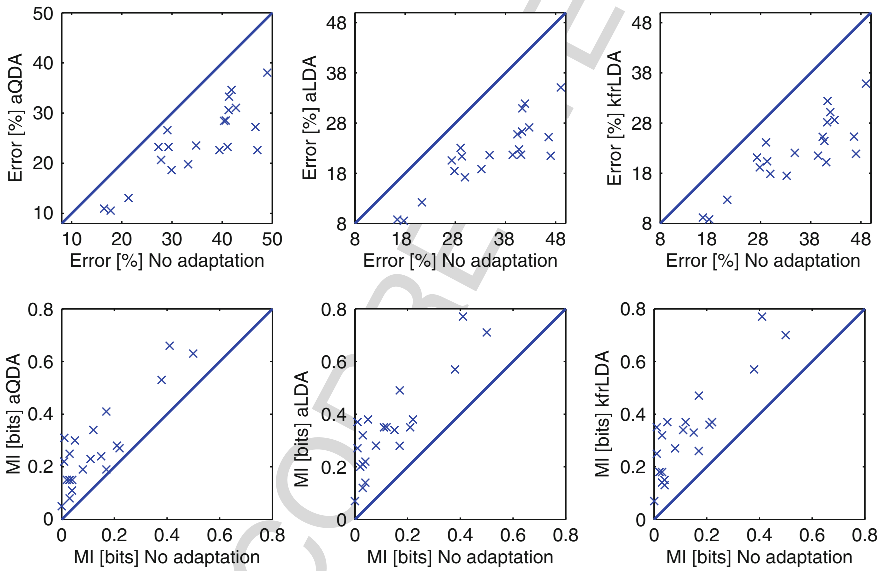


Fig. 7 Scatter plots of performance (error rates in first row and mutual information in second row) of different adaptive classifiers using RLS-AAR+V feature. "No adaptation" uses the initial classifier computed from previously recorded data from 7 different subjects. "aQDA" is the adaptive QDA approach, aLDA is the adaptive LDA approach with fixed update rate, and kfrLDA is the (variable rate) adaptive LDA using Kalman filtering. In the first row, values below the diagonal show the superiority of the method displayed in the y-axis. In the second row (MI values), the contrary is true. These results suggest the superiority of adaptive classification versus no adaptation

Table 3 Average and SEM of ERR and MI at predefined time points. Error rate values were taken from results shown also in Fig. 7

Classifier	ERR[%]	MI[bits]
No adapt	35.17±2.08	0.132±0.031
Adaptive QDA (aQDA)	24.30±1.60	0.273±0.036
Adaptive LDA (aLDA)	21.92±1.48	0.340±0.038
Kalman LDA (kfrLDA)	22.22±1.51	0.331±0.039

Figure 8 depicts how the weights of the adaptive classifier change in time, and we can see a clear long-term change in their average value. This change can be largely explained by the improved separability due to feedback training. To present the changes in the feature space, the features were projected into a two-dimensional subspace defined by the optimal separating hyperplanes similar to [14, 37]. Figure 9 shows how the distributions (means and covariances of the features) change from session 1 to 2 and from session 2 to 3. In this example, the optimal projection changes and some common shift of both classes can be observed. The change of the optimal projection can be explained by the effect of feedback training. However, the common shift of both classes indicates also other long-term changes (e.g. fatigue, new electrode montage, or some other change in recording conditions).

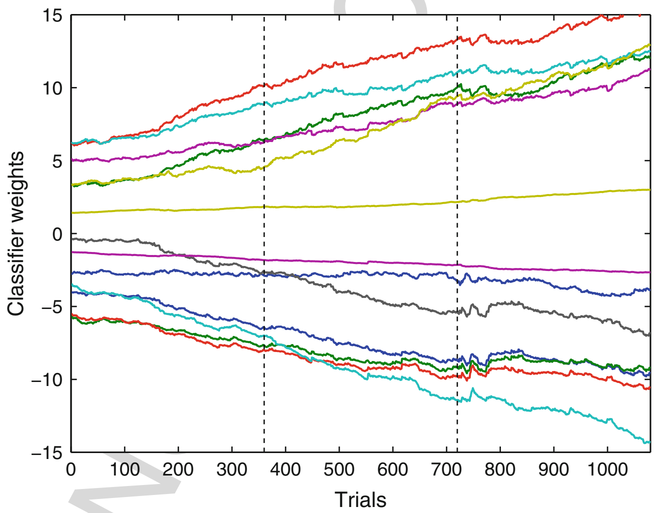


Fig. 8 Classifier weights changing in time of subject S11. These changes indicate consistent long-term changes caused by an improved separability due to feedback training. The data is from three consecutive sessions, each session had 360 trials. The changes after trial 720 probably indicate some change in the recording conditions (e.g. due to the new electrode montage)

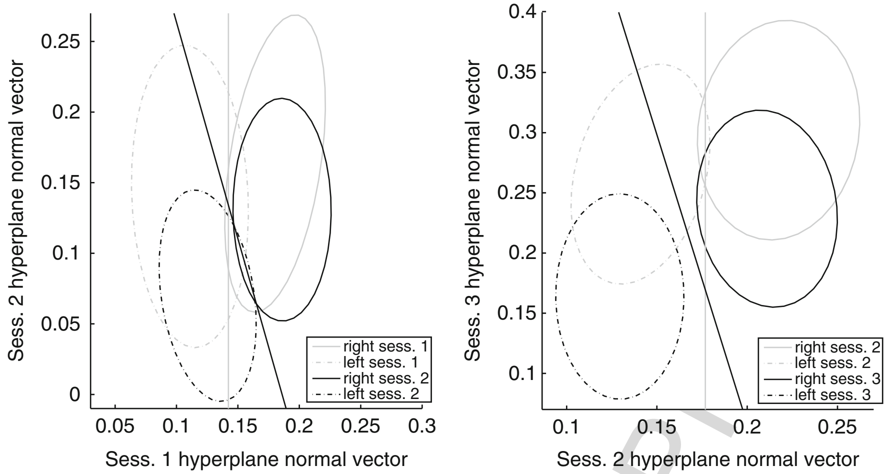


Fig. 9 Changes in feature distributions from session 1 to session 2 (*left*) and session 2 to session 3 (*right*). The separating hyperplanes of two sessions were used to find an orthonormal pair of vectors, and the features were projected in this subspace. The averages and covariances of each class and each session are projected into a common 2-dimensional subspace defined by the two separating hyperplanes

1.7 Discussion

Compensating for non-stationarities in complex dynamical systems is an important topic in data analysis and pattern recognition in EEG and many other analysis. While we have emphasized and discussed the use of adaptive algorithms for BCI, there are further alternatives to be considered when dealing with non-stationarities: (a) segmentation into stationary parts where each stationary system is modeled separately (e.g. [22]), (b) modeling invariance information, i.e. effectively using an invariant feature subspace that is stationary for solving the task, (c) modeling the non-stationarity of densities, which can so far be remedied only in the covariate shift setting, where the conditional $p(y|x)$ stays stable and the densities $p(x)$ exhibit variation [40, 41].

An important aspect when encountering non-stationarity is to measure and quantify the degree of non-stationary behavior, e.g. as done in [37]. Is non-stationarity behavior caused by noise fluctuations, or is it a systematic change of the underlying system? Depending on the answer, different mathematical tools are suitable [33, 36, 40, 47].

Several adaptive methods have been introduced and discussed. The differences between rectangular and exponential windows are exemplified in the adaptive mean estimation. The advantage of an exponential window is shown in terms of computational costs, the memory requirements and the computational efforts are independent of the window size and the adaptation time constant. This advantage holds not only for the mean but also for all other estimators.

The extended covariance matrix was introduced, which makes the software implementation more elegant. An adaptive estimator for the inverse covariance matrix was introduced; the use of the matrix inversion lemma enables avoiding an explicit (and computational costly) matrix inversion. The resulting algorithm was suitable for adaptive LDA and adaptive QDA classifiers. The Kalman filtering method for the general state-space model was explained and applied to two specific models, namely (i) the autoregressive model and (ii) the linear combiner (adaptive LDA) in the translation step.

All techniques are causal (that is, they use samples only from the past and present but not from the future) and are therefore suitable for online and real-time application. This means that no additional time delay is introduced, but the total response time is determined by the window size (update coefficient) only. The presented algorithms have been implemented and tested in M-code (available in Biosig for Octave and Matlab [30]), as well as in the real-time workshop for Matlab/Simulink. These algorithms were used in several BCI studies with real-time feedback [46–48].

The use of subsequent adaptive steps can lead, at least theoretically, to an unstable system. To avoid these pitfalls, several measures were taken in the works described here. First, the feature extraction step and the classification step used very different time scales. Specifically, the feature extraction step takes into account only changes within each trial, and the classification step takes into account only the long-term changes. A more important issue might be the simultaneous adaptation of the subject and the classifier. The results of [46, 47, 48] also demonstrate that the used methods provide a robust BCI system, since the system did not become unstable. This was also supported by choosing conservative (i.e. small) update coefficients. Nevertheless, there is no guarantee that the BCI system will remain stable under all conditions. Theoretical analyses are limited by the fact that the behavior of the subject must be considered. But since the BCI control is based on deliberate actions of the subject, the subject's behavior can not be easily described. Therefore, it will be very difficult to analyse the stability of such a system from a theoretical point of view.

The present work did not aim to provide a complete reference for all possible adaptive methods, but it provides a sound introduction and several non-trivial techniques in adaptive data processing. These methods are useful for future BCI research. A number of methods are also available from BioSig - the free and open source software library for biomedical signal processing [30].

Acknowledgments This work was supported by the EU grants "BrainCom" (FP6-2004-Mobility-5 Grant No 024259) and "Multi-adaptive BCI" (MEIF-CT-2006 Grant No 040666). Furthermore, we thank Matthias Krauledat for fruitful discussions and tools for generating Fig. 5.

[1] (1). 1 [2]@lastnameuse@12 [2]@last nameusel @star2 [2]1 22, 1 gobble.

References

AQ5

1. Block matrixdescompositions http://ccrma-www.stanford.edu/~jos/[note]selattice/Block_matrix_decompositions.html
2. N. Birbaumer et al., The thought-translation device (TTD): Neurobehavioral mechanisms and clinical outcome. IEEE Trans Neural Syst Rehabil Eng, 11(2), 120–123, (2003).

3. B. Blankertz et al., The BCI competition. III: Validating alternative approaches to actual BCI problems. IEEE Trans Neural Syst Rehabil Eng, 14(2), 153–159, (2006).
4. B. Blankertz et al., Optimizing spatial filters for robust EEG single-trial analysis. IEEE Signal Proc Mag, 25(1), 41–56, (2008).
5. del R. Millán and Mouriño J.del, R. Millán and J. Mouriño, Asynchronous BCI and local neural classifiers: An overview of the adaptive brain interface project. IEEE Trans Neural Syst Rehabil Eng 11(2), 159–161, (2003).
6. G. Dornhege et al., Combining features for BCI. In S. Becker, S. Thrun, and K. Obermayer (Eds.), *Advances in neural information processing systems,* MIT Press, Cambridge, MA, pp. 1115–1122, (2003).
7. G. Dornhege et al., Combined optimization of spatial and temporal filters for improving brain-computer interfacing. IEEE Trans Biomed Eng, 53(11), 2274–2281, (2006).
8. G. Dornhege et al., *Toward brain-computer interfacing*, MIT Press, Cambridge, MA, (2007).
9. S. Haykin, *Adaptive filter theory*, Prentice Hall International, New Jersey, (1996).
10. B. Hjorth, “EEG analysis based on time domain properties.” Electroencephalogr Clin Neurophysiol, 29(3), 306–310, (1970).
11. I.I. Goncharova and J.S. Barlow, Changes in EEG mean frequency and spectral purity during spontaneous alpha blocking. Electroencephalogr Clin Neurophysiol, 76(3), 197–204; Adaptive methods in BCI research – an introductory tutorial 25, (1990).
12. R. Kalman and E. Kalman, A new approach to Linear Filtering and Prediction Theory. J Basic Eng Trans ASME, 82, 34–45, (1960).
13. B.R.E. Kalman, and R.S. Bucy, New results on linear filtering and prediction theory. J Basic Eng, 83, 95–108, (1961).
14. K. M. Krauledat, Analysis of nonstationarities in EEG signals for improving Brain-Computer Interface performance, Ph. D. diss. Technische Universität Berlin, Fakultät IV – Elektrotechnik und Informatik, (2008).
15. G. Krausz et al., Critical decision-speed and information transfer in the ‘graz brain-computer-interface’. Appl Psychophysiol Biofeedback, 28, 233–240, (2003).
16. S. Lemm et al. Spatio-spectral filters for robust classification of single-trial EEG. IEEE Trans Biomed Eng, 52(9), 1541–48, (2005).
17. A. Li, G. Yuanqing, K. Li, K. Ang, and C. Guan, Digital signal processing and machine learning. In B. Graimann, B. Allision, and G. Pfurtscheller (Eds.), *Advances in neural information processing systems*, Springer, New York, (2009).
18. J. McFarland, A.T. Lefkowicz, and J.R. Wolpaw, Design and operation of an EEG-based brain-computer interface with digital signal processing technology. Behav Res Methods, Instruments Comput, 29, 337–345, (1997).
19. D.J. McFarland et al., BCI meeting 2005-workshop on BCI signal processing: feature extraction and translation.” IEEE Trans Neural Syst Rehabil Eng 14(2), 135–138, (2006).
20. R.J. Meinhold and N.D. Singpurwalla, Understanding Kalman filtering. Am Statistician, 37, 123–127, (1983).
21. E. Oja and J. Karhunen, On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. J Math Anal Appl, 106, 69–84, (1985).
22. K. Pawelzik, J. Kohlmorgen and K.-R. Muller, Annealed competition of experts for a segmentation and classification of switching dynamics. Neural Comput, 8, 340–356, (1996).
23. G. Pfurtscheller et al., On-line EEG classification during externally-paced hand movements using a neural network-based classifier. Electroencephalogr Clin Neurophysiol, 99(5), 416–25, (1996).

24. N.G. Pfurtscheller and C. Neuper, Motor imagery and direct brain-computer communications, Proceedings IEEE 89, 1123–1134, (2001).
25. G. Pfurtscheller et al., EEG-based discrimination between imagination of right and left hand movement. Electroencephalogr Clin Neurophysiol, 103, 642–651, (1997).
26. G. Pfurtscheller et al., Current trends in Graz brain-computer interface (BCI) research. IEEE Trans Rehabil Eng, 8, 216–219, (2000).
27. G. Pfurtscheller et al., "Separability of EEG signals recorded during right and left motor imagery using adaptive autoregressive parameters." IEEE Trans Rehabil Eng, 6(3), 316–25, (1998).
28. N. Pop-Jordanova and J. Pop-Jordanov, Spectrum-weighted EEG frequency (Brainrate) as a quantitative indicator of arousal Contributions. Sec Biol Med Sci XXVI(2), 35–42, (2005).
29. A. Schlogl "Optimal spatial filtering of single trial EEG during imagined hand movement. IEEE Trans Rehabil Eng. 8(4), 441–446, (2000).
30. A. Schlogl, BioSig – an open source software library for biomedical signal processing, http://biosig.sf.net, 2003–2008
31. A. Schlogl, D. Flotzinger and G. Pfurtscheller, Adaptive autoregressive modeling used for single-trial EEG classification. Biomedizinische Technik, 42, 162–167, (1997).
32. A. Schlögl et al. "Information transfer of an EEG-based brancomputer interface." First international IEEE EMBS conference on neural engineering, 2003, 641–644, (2003).
33. A. Schlögl et al. "Characterization of four-class motor imagery eeg data for the bci-competition 2005. J Neural Eng, 2(4), L14–L22, (1997).
34. C. Neuper and G. Pfurtscheller, Estimating the mutual information of an EEG-based brain-computer-interface. Biomedizinische Technik, 47(1–2), 3–8, (2002).
35. S.J. Roberts and G. Pfurtscheller, A criterion for adaptive autoregressive models. Proceedings Ann Int Conf IEEE Eng Med Biol, 2, 1581–1582, (2000).
36. A. Schlögl, *The electroencephalogram and the adaptive autoregressive model: theory and applications*, Shaker Verlag, Aachen, Germany, (2000).
37. M.P. Shenoy, R. P. N. Rao, M. Krauledat, B. Blankertz and K.-R. Müller, Towards adaptive classification for BCI. J Neural Eng, 3(1), R13–R23, (2006).
38. H. Shimodaira, Improving predictive inference under covariate shift by weighting the log likelihood function. J Stat Plan Inference, 90, 227–244, (2000).
39. V. Solo and X. Kong, Performance analysis of adaptive eigenanalysis algorithms, IEEE Trans Signal Process, 46(3), 636–46, (1998).
40. M. Sugiyama, M. Krauledat and K.-R. Müller, Covariate shift adaptation by importance weighted cross validation. J Mach Learning Res 8, 1027–1061, (2007).
41. M.M. Sugiyama and K.-R. Müller, Input-dependent estimation of generalization error under covariate shift. Stat Decis, 23(4), 249–279, (2005).
42. S. Sun and C. Zhang, Adaptive feature extraction for EEG signal classification. Med Bio Eng Comput, 44(2), 931–935, (2006).
43. R. Tomioka et al., Adapting spatial filtering methods for nonstationary BCIs. Proceedings of 2006 Workshop on Information-Based Induction Sciences (IBIS2006), 2006, 65–70, (2006).
44. C. Vidaurre et al., About adaptive classifiers for brain computer interfaces. Biomedizinische Technik, 49(Special Issue 1), 85–86, (2004).
45. C. Vidaurre et al., A fully on-line adaptive brain computer interface. Biomedizinische Technik, 49(Special Issue 2), 760–761, (2004).
46. C. Vidaurre et al., Adaptive on-line classification for EEG-based brain computer interfaces with AAR parameters and band power estimates. Biomedizinische Technik, 50, 350–354. Adaptive methods in BCI research – an introductory tutorial, 27, (2005).
47. C. Vidaurre et al., A fully on-line adaptive BCI, IEEE Trans Biomed Eng, 53, 1214–1219, (2006).
48. C. Vidaurre et al., Study of on-line adaptive discriminant analysis for EEG-based brain computer interfaces. IEEE Trans Biomed Eng, 54, 550–556, (2007).
49. J. Wackermann, Towards a quantitative characterization of functional states of the brain: from the non-linear methodology to the global linear descriptor. Int J Psychophysiol, 34, 65–80, (1999).

50. J.R. Wolpaw et al., Brain-computer interface technology: A review of the first international meeting. IEEE Trans Neural Syst Rehabil Eng, 8(2), 164–173, (2000).
51. J.R. Wolpaw et al., Brain-computer interfaces for communication and control." Clin Neurophysiol, 113, 767–791, (2002).

Chapter 18

Q. No.	Query
AQ1	“A. Schlog” has been set as a corresponding author Is this ok? and also check edited affiliation and insted e-mail address
AQ2	“Chp. 3.1” has been changed to sect. 3.1. please check
AQ3	Please provide footnote text for footnote citation 1
AQ4	Please provide citation for Figures 3, 4, and 5
AQ5	Please provide year for the Ref.1.