Applying evolution strategies to preprocessing EEG signals for brain–computer interfaces

Ricardo Aler [*], Inés M. Galván, José M. Valls

Universidad Carlos III de Madrid, Avda. Universidad 30, 28911 Leganés, Spain

ARTICLE INFO

ABSTRACT

An appropriate preprocessing of EEG signals is crucial to get high classification accuracy for Brain–Computer Interfaces (BCI). The raw EEG data are continuous signals in the time-domain that can be transformed by means of filters. Among them, spatial filters and selecting the most appropriate frequency-bands in the frequency domain are known to improve classification accuracy. However, because of the high variability among users, the filters must be properly adjusted to every user's data before competitive results can be obtained. In this paper we propose to use the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) for automatically tuning the filters. Spatial and frequency-selection filters are evolved to minimize both classification error and the number of frequency bands used. This evolutionary approach to filter optimization has been tested on data for different users from the BCI-III competition. The evolved filters provide higher accuracy than approaches used in the competition. Results are also consistent across different runs of CMA-ES.

Keywords:
Evolution strategies
Brain–computer interfaces
Filter optimization

1. Introduction

The aim of EEG-based[1] Brain–Computer Interfaces (BCI) is to detect patterns on user EEG signals in order to control a computer or an external device [5,22,10,18]. As an example, patterns corresponding to motor imagery (the user imagines that one of his body parts is moving), object rotation imagery, or thinking of words, can be recognized in the EEG signal. The high variability of EEG patterns among different subjects makes machine learning classification techniques the tool of choice. Thus, user-dependent classifiers can be trained from user-generated data by means of supervised machine learning techniques [7]. The classifier can then be used to detect patterns on real-time EEG data.

A large variety of classification algorithms have been tested in BCI research. [25] is an early work where linear and non-linear classifiers are discussed. A complete and recent survey on machine learning techniques for BCIs can be found in [20]. It surveys a large variety of classification techniques, including linear discriminant analysis, support vector machines, neural networks, bayesian classifiers, nearest neighbor classifiers, and ensembles of classifiers. From a more applied point of view, [26] discusses recent machine learning classification and pre-processing techniques for an actual BCI system. Finally, some advanced techniques such as Fuzzy systems [29,23], Gaussian Processes [35], Temporal Models [17], Dynamical Bayesian Networks [34], or Hidden Markov Models [31] are also becoming techniques of interest for BCI's and EEG analysis.

In the context of classification of EEG signals, an important issue for the success of the classifier is to determine the relevant input information. EEG signals are basically time-series that are captured from electrodes. The use of all the instants of the time-series prevents the learning of the classifier for several reasons. First, the number of attributes for classification

[*] Corresponding author.
E-mail addresses: aler@inf.uc3m.es (R. Aler), igalvan@inf.uc3m.es (I.M. Galván), jvalls@inf.uc3m.es (J.M. Valls).

[1] EEG stands for electroencephalogram.

would be too large because they would include all the instants of the time-series. It is known that if the relation between the number of training instances and the number of attributes in the data set is too low, the resulting classifier might overfit the training data [3]. Thus, in the machine learning context, learning the appropriate patterns or avoiding overfitting may become difficult. In addition, it is known that patterns are best detected on the frequency-domain rather than in the time-domain of the original raw data. Therefore, an appropriate preprocessing of the raw signal is acknowledged to be very important in order to get high classification accuracy.

Three kinds of transformations are commonly used [7]: spatial filters, the Fourier Transform (to convert to the frequency domain), and band-pass filters. Spatial filters are useful because the signal detected at one electrode can come from different parts of the brain. Such filters can generate a more localized signal for every electrode. A very simple spatial filter is the Laplacian filter which subtracts the average signal of surrounding electrodes from each electrode, thus sharpening the signal generated just below it. Some other kinds of spatial filters, such as common average referencing, principal component analysis, etc. can also be represented by linear transformations on the original data [7]. The signal must be also transformed to the frequency domain by using, for instance, the Fast Fourier Transform and Power Spectral Density (PSD). This can be achieved by other related possibilities, such as Wavelet Transforms [30] and bispectrum [36]. Moreover, it is known that the patterns of interest are located in particular frequency bands. As an example, motor imagery is accompanied by increase/decrease of amplitude in the frequency band from 8 Hz to 15 Hz and can be easily observed in the frequency-domain signal (this phenomena is called event-related desynchronization (ERD) and event-related synchronization (ERS) [28]). Therefore, it is important to use the most relevant attributes from all the available frequency-bands, bearing in mind that different users will display ERD/ERS phenomena on slightly different bands. Frequency-selection filters are used in this paper for that purpose.

Filters can be adjusted by hand, following a process of trial and error. This requires some experience on building and adjusting filters that at the end, may turn out not to be optimal. There are also approaches to compute filters automatically. Most work have been carried out on computing spatial filters by common spatial patterns (CSP) [9,7]. This approach allows to determine data projections, represented by linear transformations, that maximize the variance of signals of one class and minimize the variance of signals of the other class. There are some extensions to CSP that allow to also learn band-pass (or spectral) filters [19,6,33,2]. Maximizing the variance of one class while minimizing that of the other clearly increases the discriminability of the two classes, so it is expected that most classifiers will obtain high accuracies on the filtered data. But CSP and its variants do not directly optimize the accuracy of the classifier on the data.

In this paper, we propose to optimize simultaneously both spatial and frequency selection filters by means of the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [27,15]. Therefore, given a classifier, our approach tries to find the optimal spatial and frequency-selection filters that maximize directly the accuracy of the classifier. CMA-ES is able to optimize functions without making strong assumptions about them and therefore is specially appropriate for noisy problems such as BCI classification. CMA-ES has several other advantages. For instance, the algorithm does not require to perform extensive parameter setting. Although there are many parameters involved, the designers of the algorithm provide standard values that have been shown to work in many different contexts. Other important parameters, such as the mutation step-size, are self-adapted by the algorithm as the optimization progresses. In this work a brief description is made, but more details about CMA-ES can be found in [15] and an in-depth tutorial in [16].

This paper is structured as follows. Section 2 expands on the key concepts of BCIs. Section 3 includes a description of CMA-ES. Section 4 describes how spatial and frequency-selection filters can be used to preprocess raw EEG data and to generate a set of training instances. Section 5 describes how filters are evolved by CMA-ES. The approach is empirically tested in Section 6. Finally, Section 7 summarizes the conclusions of this work.

2. Architecture of brain–computer interfaces

A BCI is a device designed to translate certain kinds of thoughts into actions. Their main aim is to restore motor function for disabled patients, suffering for instance from amyotrophic lateral sclerosis or the locked-in syndrome [18]. BCI's can be either invasive or non-invasive. The focus of this paper is non-invasive BCI's, where electrodes are attached on the scalp in various locations through which EEG activity is recorded.

In a typical BCI setting, subjects are required to perform specific mental tasks while their EEG is being recorded. The EEG signal is amplified and sent to the computer where it can be analyzed by algorithms. Depending on the kind of BCI, different features can be detected on the EEG signal and transformed into actions (such as moving a cursor on the screen or controlling a wheelchair). These features can be either voluntarily generated by the user (such as slow cortical potentials or sensorimotor rhythms) or elicited by visual or auditory stimulation (event-related potentials or steady-state evoked potentials).

The data used in this paper involves mainly sensorimotor rhythms (SMR), so they will be explained in more detail. SMR can be detected on the somatosensory cortex as changes in frequency amplitude in the range [8–11] Hz (and also around 20 Hz and 40 Hz). Users can control SMRs by voluntarily blocking or desynchronizing them (i.e. the signal amplitude decreases at those frequencies). Desynchronization occurs with movement or preparation for movement, and also with motor imagery (imagination of movements). Synchronization (increase in amplitude) returns with relaxation or after the (imagination of the) movement. Of particular interest for BCI's is the modulation of SMRs with motor imagery because it can be used with patients who cannot move any of their body parts. Just the imagination of the movement of the left or the right hand is enough to desynchronize a particular region of the somatosensory cortex. Motor imagery of different body parts desynchronize different regions of the somatosensory cortex. However, different users have different (de)synchronization patterns so a way of building SMR-based BCI's is to adapt the detection of synchronization/desynchronization EEG features for each user. This can be achieved by means of supervised machine learning classification algorithms.

The architecture of a SMR machine learning-based BCI can be seen in Fig. 1. The EEG is recorded from the electrodes, then the signal is preprocessed using different kind of filters, then classified, and finally used to control the computer or some other device such as a wheelchair.

The learning of the classifier and the adjustment of the filters is carried out off-line. The user goes through an acquisition session where EEG data is recorded while the user is instructed to perform certain mental tasks (or achieve certain states). In SMR-based BCIs these are usually related to motor imagery. The acquisition session is usually divided into periods or epochs, each one devoted to a different mental task, such as the imagination of the movement of the left hand, of the right hand, of the feet, etc. At the end of the session, EEG data is available for every mental task. Once the data has been acquired, a preprocessing mechanism is used to filter the signals (transformation to the frequency domain, spatial filtering, etc.). Adjusting filters is key for obtaining good classification rates. After that, machine learning classification algorithms can be used to build the classifier that will be later available for on-line use.

3. The Covariance Matrix Adaptation Evolution Strategy (CMA-ES)

The algorithm CMA-ES (Covariance Matrix Adaptation Evolution Strategy) was proposed by Hansen and Osterman [14,15]. It is an evolution strategy for difficult optimization problems in continuous domains, characterized by the use of a covariance matrix to guide the search process. In this Section, we will give a short overview of a (μ,λ) CMA-ES, where μ is the number of parents and λ the number of offspring. The CMA tutorial could be consulted by the interested reader for a complete description of the algorithm [16].

CMA-ES estimates a probability distribution from the best performing samples in order to guide the search towards promising regions of the search space. In CMA-ES, the probability distribution to be estimated is a multivariate normal $N(m,\delta^2 \cdot C)$, where m is the mean and $\delta^2 \cdot C$ is the covariance matrix, decomposed into matrix C and scalar step-size δ. The mean represents the current location of the search and it moves towards better locations as search progresses. The covariance matrix controls mutations and is used to guide the search. In some sense, the covariance matrix "points" towards better solutions and is estimated from past samples x_i that performed well with respect to fitness function f. Let's suppose that we desire to minimize a fitness function $f(x)$: $\mathbf{R}^p \rightarrow \mathbf{R}$, where p is the dimensionality of the problem, the basic steps of CMA-ES algorithm are the following:

1. Initialize distribution parameters m, δ, and C.
2. For generation (iteration) $t = 0, 1, 2, \ldots$:
   (a) Sample λ points from $N(m,\delta^2 \cdot C)$: $x_i \leftarrow N(m,\delta^2 \cdot C)$
   (b) Evaluate fitness f of $x_1, \ldots, x_\lambda$
   (c) Update distribution parameters m, δ, and C based on the best performers $x_1, \ldots, x_\mu$

In CMA-ES algorithm the distribution parameters (m, δ, and C) are updated every generation. Next, the mechanism to update them is briefly described.

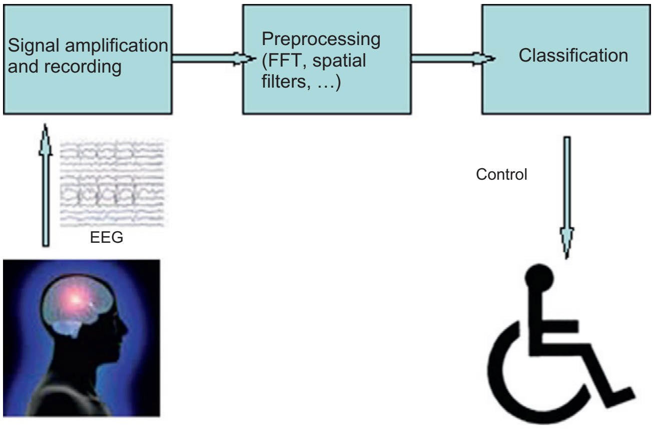

Signal amplification and recording
Preprocessing (FFT, spatial filters, ...)
Classification
EEG
Control

Fig. 1. Architecture of a BCI system.

Let us now suppose that index i has been sorted according to fitness f and that the best μ performers according to f are selected. The updating of the mean m is given by the following equation (where the $\sum w_i = 1$ are weights chosen by the user).

$$m \leftarrow \sum_{i=1}^{\mu} w_i x_i \tag{1}$$

CMA-ES updates the step-size δ by taking into account the correlation of the best performing individuals x_i during a number of generations. The sequence of successive x_i is used to compute an evolution path, which is represented by a vector s_δ. When the x_i in the path are correlated – all x_i are going in a similar direction – the same distance could be covered in fewer but longer steps. Therefore, δ should be increased. On the other hand, when the evolution path is anticorrelated – the x_i go in contrary directions and therefore cancel each other – the step-size δ should be made smaller so that region of the search space can be explored with a finer grain. In short, it is desirable that there are no correlations among samples along an evolution path. For this reason, the updating method is based on the principle that the adaptation of the step-size should reduce the difference between the distributions of the current evolution and an evolution path under random selection. The expression to update the step-size is given by

$$\delta \leftarrow \delta \cdot \exp\left\{\beta \cdot \left(\frac{\|s_\delta\|}{\|E[N(0,I)]\|} - 1\right)\right\} \tag{2}$$

where β is a parameter that determines the step size variation between successive generations, $\|E[N(0,I)]\|$ is the expectation of the length of a $N(0,I)$ distributed random vector in $\mathbf{R}^p$, and $\|s_\delta\|$ is the length of the current evolutionary path. Thus, Eq. (2) updates δ by comparing the length of the current evolutionary path $\|s_\delta\|$ and the expected length of an evolutionary path under random selection $\|E[N(0,I)]\|$.

Finally, the adaptation of the covariance matrix C is described. An evolution path s_c is computed in a similar way to s_δ (see [16]). Then, C is updated according to

$$C \leftarrow (1 - c_{cov}) \cdot C + \alpha_1 \cdot \mathbf{Cov}(s_c) + \alpha_2 \cdot \mathbf{Cov}(x_{i=1..\mu}) \tag{3}$$

Eq. (3) shows that the old covariance matrix C is updated by adding two components: $\mathbf{Cov}(s_c)$ and $\mathbf{Cov}(x_{i=1...\mu})$. The former is related to the covariance due to the evolutionary path s_c (i.e. the best steps x_i found during the history of the search). The latter is the covariance due to the best μ samples x_i in the current generation. The authors call them the Rank-1-update and the Rank-μ-update, respectively. The idea here is to take advantage both of the global evolution history (s_c) and the information provided by best individuals in the new generation. $(1 - c_{cov})$ is a learning rate and measures the importance given to the previous generation C, whereas $\alpha_1 + \alpha_2 = c_{cov}$ summarize a combination of parameters that weight $\mathbf{Cov}(s_c)$ and $\mathbf{Cov}(x_{i=1...\mu})$.

The matrix C is initialized to the identity matrix, and initial values of m and δ are problem dependent and the authors recommend to set them so that the optimum x is within a cube $m \pm 3 \cdot \delta$. It can be seen that CMA-ES has many parameters. However, the authors provide some settings that are robust for many different problems, and usually they need not be changed [16]. The most important are $\beta = 1/p,\ c = 1/\sqrt{p}$ and $c_{cov} = 2/p^2$. The CMA-ES implementation also increases automatically the population size if automatic restarts are allowed.

4. Preprocessing EEG signals

The aim of this paper is to automatically compute spatial and band-selection filters in order to improve the classification rate of EEG signals. In this Section it will be explained in detail how EEG signals can be preprocessed (filtered) as a preliminary step for training machine learning classifiers.

As explained in Section 1, classification can be improved if new attributes in the frequency domain with high information content can be generated from the original time series. In this paper, the preprocessing stage used is made of three steps:

1. Apply the Fast Fourier Transform (**FFT**) to the raw data to transformed signals from the time-domain to the frequency-domain.
2. Apply a spatial filter to the frequency domain data. The spatial filter can be seen as a linear transform represented by a matrix. In our case, this matrix will be called S.
3. Apply a frequency-band selection filter by which some of the frequency bands in the frequency domain representation of the EEG signal are removed. This frequency selection filter will be referred to as B.

The effect of combining the spatial filter S and the frequency domain transform can be represented by Eq. (4), where matrix M contains the time series for all the electrodes and **FFT** is the Fast Fourier Transform. **FFT** is a linear transform, and therefore the same transformation can be expressed by the second term in Eq. (4). We will take advantage of this fact later in order to improve the efficiency of the filter learning process. A preliminary version of the preprocessing without this improvement can be found in [1].

$$\mathbf{FFT}(M * S) = \mathbf{FFT}(M) * S \tag{4}$$

Let us remember that our final goal is to learn automatically the filters (S and B), so that the classification of EEG signals is optimal. Before that process is explained in Section 5, a more detailed explanation of how to carry out the **FFT**/S/B three preprocessing steps is given.

Let c be the number of electrodes or channels and let f be the sampling frequency. An acquisition session is a session where data is acquired from a subject wearing an EEG cap. If the acquisition session lasts for s seconds, then the temporal series for the whole acquisition session contains $f * s$ data points (time instants) for each of the c channels. Therefore, a session can be represented as a $(f * s) \times c$ matrix, that will be named $\mathbf{M}$.

As explained in Section 2, during on-line use, the output of the BCI is continuous: at each time instant, a chunk of signal is preprocessed and sent to the classifier in order to generate an output. In this paper, the training instances are generated by a process that mimics on-line use: every training instance is based on the previous signal chunk (the chunk size is the same than during on-line use). In order to simulate the pass of time, a moving window of t time instants (i.e. the signal chunk size) moves through $\mathbf{M}$ by steps of δt time instants. This process creates a sequence of submatrices of dimension $t \times c$, $M_1, M_2, \ldots, M_n, \ldots$ extracted from $\mathbf{M}$. Each one of the submatrices represents a signal chunk and will be used to generate one training instance. Thus, M_1 is extracted from $\mathbf{M}$ from 1 to t, M_2 from $1 + \delta t$ to $t + \delta t$, M_3 from $1 + 2\delta t$ to $t + 2\delta t$, and so on. In general, submatrix M_n is computed as:

$$M_n = \mathbf{M}(1 + (n-1) * \delta t : t + (n-1) * \delta t, 1 : c) \tag{5}$$

for $n = 1, 2, \ldots$, where t is the window size (the signal chunk size) and δt represents the jump between consecutive windows. In fact, δt determines the amount of overlap between windows (the larger δt, the smaller the overlap). Notation $(1{:}t{,}1{:}c)$ means that the submatrix contains all time instants from 1 to t and all channels from 1 to c. Hence, for each session $\mathbf{M}$ a set of submatrices M_n of dimension $(t \times c)$ are obtained.

Let us remember that the acquisition session was divided into periods, each one devoted to a mental task. This means that, for instance, the first 4 s of signal could be related to left hand imagery, the 4 next seconds to right hand imagery, and so on. In most of the cases, each M_n will fit completely within a period (a single mental task), but in some cases it might belong to two different mental tasks simultaneously. Since the generation of training instances is off-line, we are free to select the most appropriate ones for training. Thus, in order to remove noise from the learning process, if M_n falls within a transition between mental tasks, it will be discarded.

So far, matrices M_n represent the signal chunks that will be used to generate each of the training instances. Now, each signal chunk M_n has to be preprocessed. Let us remember that the first step is transforming to the frequency domain by means of **FFT**. In fact, a Short Time Fast Fourier Transform is used, since **FFT** is applied to a finite length signal chunk M_n. This is easily done by applying

$$M'_n = \mathbf{FFT}(M_n) \tag{6}$$

Once the **FFT** is applied, rows of M'_n from 1 to $t/2$ represent the frequency band $[0 - f/2]$ Hz (where f is the sampling frequency and t is the window size), with a resolution of $\delta f = (f/t)$ Hz. The frequency bands contained in the matrix are $[0 - \delta f]$, $[\delta f - 2 * \delta f]$, etc. At this point, some of the frequency bands can already be discarded, because it is known that outside the [8–32] Hz frequency band, there is no physiological information of interest for BCI's. Therefore, only the rows from ceil $(8/\delta f) + 1$ to ceil $(32/\delta f) + 1$ will be selected, where ceil (x) rounds x to the nearest integers greater than or equal to x. In order to simplify the notation, we will assume that M'_n already contains only the rows related to the [8–32] Hz frequency band. Thus, the number of rows remaining in M'_n is $t' = \text{ceil}\ (8/\delta f) - \ \text{ceil}\ (32/\delta f) + 1$.

Next, the spatial filter S can be applied. The spatial filters used in this work are linear transformations represented by $c \times c'$ matrices, denoted by S. The result of applying S is just the matrix product shown in

$$M''_n = |M'_n * S| \tag{7}$$

The filtered matrix M''_n is a $t' \times c'$ matrix. If $c' = c$, then M''_n has the same number of columns (i.e. the number of channels) as the original matrix M_n. If $c' < c$, then the number of channels of M''_n is reduced. In some sense, the spatial filter S transforms c channels into c' channels. The operator || in Eq. (7) computes the modulus of each one of the components of matrix M''_n. **FFT** returns complex numbers, with phase and modulus. SMR phenomena can be easily detected by means of the amplitude, so the phase will be ignored in this paper, but we will try to use the information contained in the phase in future research [7].

At this point, a training instance I_n could be constructed from M''_n. The attributes of I_n are just each of the $t' * c'$ components of M''_n. However, in most cases there would be too many attributes, some of them irrelevant, and it is well known that classification problems with many irrelevant attributes usually lead to overfitting. In this work, frequency-band selection filters B are applied to select the most appropriate frequency-bands for every user. B is represented by a matrix with the same dimensions as M''_n but containing only binary values (0 or 1). If component $B(i,j) = 0$ the $M''_n(i,j)$ component will be removed from the set of attributes of instance I_n. Only those components of $M''_n(i,j)$ with $B(i,j) = 1$ will remain.

Eq. (8) summarizes how instance I_n is generated from signal chunk M_n by means of S and B. $\otimes$ represents the component-wise selection of the components of M''_n by the corresponding elements of B and **flatten** is the flattening of the resulting matrix (concatenating all of its components). Fig. 2 displays graphically the whole preprocessing procedure, from the original time series M_n to the instances I_n used for training the classifier.

$$I_n = \mathbf{flatten}(|(\mathbf{FFT}(M_n) * S)| \otimes B) \tag{8}$$

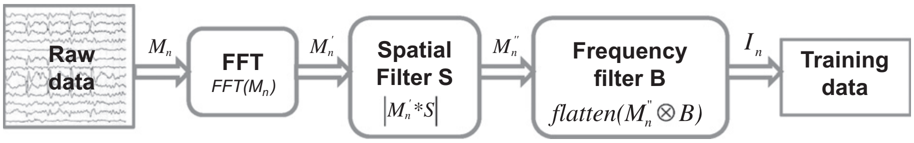


Fig. 2. Preprocessing procedure.

Training instances I_n could be used to build a classifier by means of any of the machine learning classification algorithms (such as neural networks, support vector machines, etc.). However, it would be necessary to define first the components of matrices B and S. This could be done by hand by means of trial and error. In this paper, we propose an automatic method for computing both B and S so that high classification rates can be obtained. This will be explained in the next Section.

5. Evolution of spatial and frequency-selection filters

In this Section, it will be explained how the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) described in Section 3 has been used to optimize simultaneously S and B for a classifier C. In the previous section, it was shown that the learning of a spatial filter is equivalent to constructing a $c \times c'$ matrix S and the learning of the frequency-selection filter is equivalent to constructing a $t' \times c'$ binary matrix B. In order to specify the problem to CMA-ES, both the representation of the candidate solutions or chromosome (spatial and frequency-selection filters) and the function to be optimized (the fitness function), must be described. Also, in this section a general description of the complete process is included.

5.1. The chromosome

The chromosome contains the elements to be optimized: the spatial filter S and the frequency-selection filter B, as it shown in Fig. 3. The spatial filter is encoded in the first part of the chromosome as a sequence of real numbers that is decoded as matrix S. The frequency-selection filter is contained in the second part of the chromosome and represents a sequence of boolean values. It can be decoded as a matrix where each value determines if a particular frequency band is selected or not for a particular channel.

Since CMA-ES evolves real numbers, the matrix B must be also encoded as real numbers. Hence a procedure to transform real number to binary ones must be applied. A straightforward mapping would be, for instance, negative values mapped to 0 and positive values to 1. However, this solution was discarded because for large values it would be very difficult for the mutation operator to move from positive to negative, or viceversa. The encoding that has been finally used first computes the integer part and then maps odd numbers to 1 and even numbers 0. The idea is to interleave 1s and 0s (or equivalently odd and even numbers), so that the mutation operator can flip easily between 1s and 0s.

5.2. The fitness function

In order to evaluate the quality of filters (B,S) encoded in a chromosome, a classifier C is built on the training data filtered by S and B, as summarized in Eq. (8) and explained in Section 4. However, an important remark is that Eq. (8) has to be applied to evaluate every CMA-ES individual in every generation. The number of fitness evaluations in evolutionary methods is usually high, hence any efficiency improvement is important. Eq. (8) shows that $\mathbf{FFT}(M_n)$ is independent of S and B. This means that $M_n' = \mathbf{FFT}(M_n)$ can be precomputed before CMA-ES starts, and constructing the training instances by means of

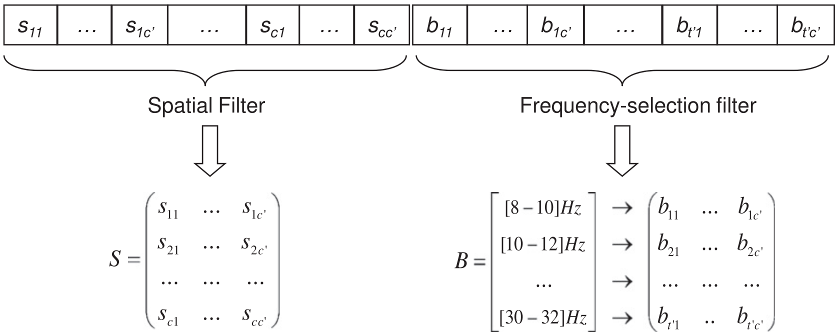

s_{11}
…
$s_{1c'}$
…
s_{c1}
…
$s_{cc'}$
b_{11}
…
$b_{1c'}$
…
$b_{t'1}$
…
$b_{t'c'}$
Spatial Filter
Frequency-selection filter
$S = \begin{pmatrix} s_{11} & \dots & s_{1c'} \\ s_{21} & \dots & s_{2c'} \\ \dots & \dots & \dots \\ s_{c1} & \dots & s_{cc'} \end{pmatrix}$
$B = \begin{bmatrix} [8-10]Hz \\ [10-12]Hz \\ \dots \\ [30-32]Hz \end{bmatrix} \begin{matrix} \rightarrow \\ \rightarrow \\ \rightarrow \\ \rightarrow \end{matrix} \begin{pmatrix} b_{11} & \dots & b_{1c'} \\ b_{21} & \dots & b_{2c'} \\ \dots & \dots & \dots \\ b_{t'1} & .. & b_{t'c'} \end{pmatrix}$

Fig. 3. Structure of the chromosome.

Eq. (9), where M'_n has been precomputed at the beginning of the run. This is in fact the reason why **FFT** is applied before S and not the other way around (let us remember that, from Eq. (4), both ways are equivalent).

$$I_n = \mathbf{flatten}\left(|M'_n * S| \otimes B\right) \tag{9}$$

C is constructed from the training set made of instances I_n. In this case, C has been chosen to be the Fisher Discriminant (FD), a well-known fast linear classifier [8]. In order to deal with multiclass problems, the one-versus-all approach has been applied to FD. An N_c-class classification problem is transformed into N_c binary (2-class) problems where the goal is to separate class i from the rest. Thus, a Fisher Discriminant FD_i is learned for each binary problem so that $FD_i(x) > 0$ if instance x belongs to class i and $FD_i(x) < 0$ if x belongs to the rest of classes. Every FD_i is a hyperplane represented by $FD_i(x) = w_i * x + b_i$. The one-versus-all approach works by classifying instance x according to

$$FD(x) = \mathbf{maxarg}_i FD_i(x) = \mathbf{maxarg}_i (w_i * x + b_i) \tag{10}$$

The quality of a classifier can be measured as the accuracy rate (the percentage of instances correctly classified, a value between 0% and 100%) or the classification error (1 minus the accuracy rate). Classification error could have been optimized directly [1]. However, it is a discrete quantity. In this paper, in order to provide a continuous and more precise feedback, it has been decided to optimize the mean squared error instead, defined in

$$\mathbf{MSE}_{FD} = \sum_n \sum_{i=1}^{i=N_c} (\sigma(w_i * I_n + b_i) - y_{i,I_n})^2 / N \tag{11}$$

where n is the number of instances, σ is a sigmoid function between 0 and 1, $\sigma(x) = 1/(1 + e^{(-x)})$, and y_{i,I_n} is the desired output for instance I_n and binary problem i ($y_{i,I_n} = 1$ if the actual class of I_n is i and 0 otherwise). $\mathbf{MSE}_{FD}$ is computed on the same training data used to construct FD (the set of instances I_n).

In order to select the most relevant frequency-bands, the number of components set to 1 in matrix B is included in the fitness function. This number has been normalized to $[0,1]$ and is denoted by $|B|$. A parameter λ is used to weight the importance of attribute minimization versus error minimization. Eq. (12) displays the fitness function to be minimized by CMA-ES.

$$\mathbf{fitness}(B, S) = \mathbf{MSE}_{FD} + \lambda |B| \tag{12}$$

where $\mathbf{MSE}_{FD}$ is the **MSE** of the FD classifier applied on the training data transformed by Eq. (9) and λ is the regularization parameter.

As it has been mentioned before, in this work Linear Discriminant Analysis (LDA a.k.a. the Fisher Discriminant or FD) has been chosen as classifier. Other supervised classification techniques could have been used to train the classifier C: support vector machines, neural networks, etc. Previous literature suggests to use linear methods, specially if data is scarce [24]. Additionally, as the fitness function is evaluated for every chromosome, its computation must be fast, and FD satisfies this requirement. Although linear-kernel Support Vector Machines (SVM) usually achieve higher accuracy, it is much slower. SVM can still be used at the end of the run, once the best (B,S) pair has been obtained. The rationale is that if a (B,S) pair works well for FD, similar or better results could be obtained with a better linear classifier (SVM). Of course, this will be tested experimentally.

5.3. The complete process

Obtaining the spatial and frequency-selection filters, involves the following process, that has been summarized graphically in Fig. 4.

1. Acquiring the raw EEG data (**M**) from a subject during an acquisition session.
2. Given the overlap parameter δt and given the size of the windows t, compute from **M** the submatrix M_n, for $n = 1, 2, 3, \ldots$, moving the temporal windows until **M** has been processed.

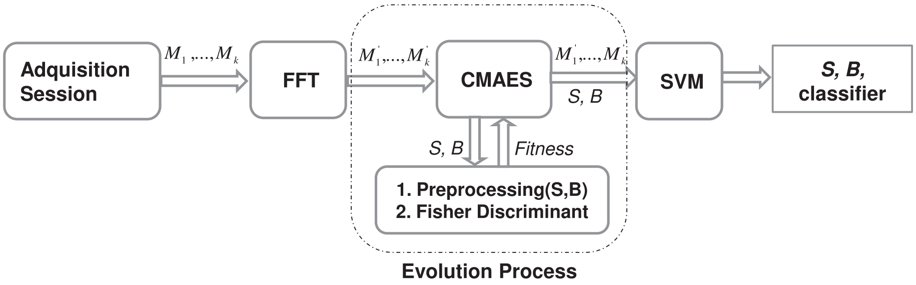


Fig. 4. Complete process for obtaining filters S, B, and the classifier.

3. Transform the M_n time series from the time domain to the frequency domain by means of $M'_n = \mathbf{FFT}(M_n)$. The M'_n matrices are stored.
4. Now, the optimization algorithm CMA-ES can be started. For every fitness evaluation of each chromosome in the population do:
   (a) Apply the spatial and the frequency-band selection filters S and B codified in the chromosome to the stored data M'_n according to Eq. (9).
   (b) The set of training instances I_n obtained are used to build a classifier C by means of the Fisher Discriminant algorithm.
   (c) The error of C on the training data is computed. That value and the number of frequencies selected ($|B|$) are used to calculate the fitness value of the chromosome using Eq. (12).
5. Once CMA-ES stops, the best S and B are returned and are used to filter the training data and construct a final classifier by means of a linear-kernel support vector machine.

6. Experimental validation

6.1. Data sets description

In this paper three datasets acquired in the IDIAP Research Institute will be used [21]. They have been previously tested in the 2005 BCI-III competition.[2] Each dataset contains data from a different subject during 4 non-feedback sessions. 32 electrodes were located on the subjects's scalp. There are 3 mental tasks, so this is a three-class classification problem:

- Imagination of repetitive self-paced left hand movements.
- Imagination of repetitive self-paced right hand movements.
- Generation of words beginning with the same random letter.

All four sessions of a given subject were acquired on the same day, each lasting 4 min with 5–10 min breaks in between them. The subject performed a given task for about 15 s and then switched randomly to another task at the operator's request. EEG data is not splitted in trials since the subjects are continuously performing any of the mental tasks. Data was provided by the competition organizers in two ways: raw EEG signals with 32 electrodes, and data with precomputed features with 8 selected electrodes. In this paper, we use both versions of the datasets: our method to evolve filters is applied on the former while the latter is used for comparison purposes.

The dataset with precomputed features provided by the competition organizers had been obtained by the following procedure. First, the raw EEG potentials were spatially filtered by means of a manually tuned surface Laplacian. Then, 16 times per second the power spectral density (PSD) in the band [8–32] Hz was estimated over the last second of data with a frequency resolution of 2 Hz. Additionally, physiological knowledge was used to select 8 centro-parietal channels (C3, Cz, C4, CP1, CP2, P3, Pz, and P4) out of the 32 original electrodes. As a result, an EEG sample is a 96-dimensional vector (8 channels times 12 frequency components).

6.2. Experimental setting

As explained before, there are 4 sessions for each one of the three subjects. The three first sessions were available for training, while the last one was used for testing the classifiers designed by the participants of the competition. The following parameters have been set to the values suggested by the provider of the data [21]:

- Sampling frequency $f = 512$ Hz.
- 1 s of data is used to construct every training instance, therefore the size of temporal windows is $t = 512$.
- Training instances are sampled 16 times per second, therefore $\delta t = \frac{512}{16} = 32$.
- The number of electrodes is $c = 32$.

Similarly to the competition precomputed features datasets, frequency bands outside the [8–32] Hz range have been removed because they contain no relevant information for our work. Also, the frequency-band width considered by the frequency-selection filter B has been set to 2 Hz, as in the competition. This means, that the frequency-bands are [8–10] Hz, [10–12] Hz, ..., [28–30] Hz, [30–32] Hz. Therefore, B is made of 12 binary values for each channel. Our spatial filter S is a $(c \times c')$ matrix: it transforms c channels into c' channels. It is known that imagination of left (right) hand movements is related to a certain part of the right (left) hemisphere [7]. If we had only these two classes, it would be reasonable to set $c' = 2$. The general idea is to set c' to the number of classes. In our problem there is a third class (imagination of random words), therefore we have also tested $c' = 3$.

The value of the λ regularization parameter must also be fixed establishing a tradeoff between the two objectives in Eq. (12): the accuracy of the classifier and the number of frequency bands selected. Large values of λ imply a strong reduction of

[2] http://www.bbci.de/competition/iii/results/index.html#martigny.

Table 1
Median and interquartile ranges over 10 runs of filters (B,S) evolved by CMA-ES of the number of frequency-bands selected, and the testing accuracy (percent) by FD and SVM for session 4. Also shown, the average and standard deviation of the number of iterations.

	Subject 1	Subject 2	Subject 3
Frequency-bands selected	4.0 (0.0)	2.0 (1.0)	5.0 (0.75)
Test accuracy (FD)	75.13 (0.63)	71.34 (0.74)	58.77 (1.25)
Test accuracy (SVM)	78.14 (0.57)	71.33 (0.71)	59.07 (1.50)
Generations	304.7 ± 71.23	580.0 ± 122.4	492.1 ± 154.7

the number of frequency bands at the beginning of the run. This means that CMA-ES can improve the fitness by drastically reducing the number of attributes during the first iterations, but sometimes this is achieved by removing some relevant frequency-bands, and therefore removing the possibility of reducing the classification error later in the run. Hence, λ should be small.

The final values of c' and λ were decided by following the above general guidelines and running some preliminary experiments with $c' = 2$, $c' = 3$, and λ from 0.05 to 0.25. Sessions 1 and 2 were used for training and session 3 for testing, so that the final testing set (session 4) is not used at all for parameter tuning. According to these experiments $c' = 2$ and $c' = 3$ give similar results. Therefore $c' = 2$ was chosen in order to facilitate the search, because the length of the chromosome and the search space are smaller. Also, $\lambda = 0.1$ provided a good tradeoff between accuracy and reduction in the number of attributes.

In this work, the MATLAB CMA-ES version 3.23.beta has been used. It can be found in http://www.lri.fr/hansen/cmaes_inmatlab.html#matlab. With respect to CMA-ES parameters, the authors provide settings that are robust for many different problems and these default values have been left unchanged. The only explicit parameters of CMA-ES are the initial search point and the initial standard deviation. The initial standard deviation has been set to 1. The initial search point contains a frequency-band selection filter with all frequency-bands selected, and a spatial filter initialized with uniform random values between -1 and +1.

In order to avoid overfitting, a stopping criterion that uses a validation set has been imposed on CMA-ES for all datasets. The validation set is obtained by mixing and randomizing processed patterns obtained from sessions 1, 2, and 3. 80% of data is selected for building the classifier and to compute the fitness function. The remaining 20% of data is used as the validation set-based stopping criterion. The classification error on the validation set is measured at each iteration and evolution is stopped when the validation error becomes almost stable (more specifically, when it changes less than 0.5% for 30 iterations).

6.3. Experimental results

For each subject, ten experiments have been carried out varying the random seed in the CMA-ES algorithm. Table 1 displays the median/interquartile ranges of the number of frequency-bands selected by the best B filter, the FD classification rate for the test data (session 4) for each subject, and the test classification rate obtained by a linear SVM.[3] All samples were tested for normality by means of the Kolmogorov–Smirnov test, but the normality hypothesis had to be rejected at $\alpha = 5\%$. Therefore, results are reported in terms of median and interquartile range. The average/standard deviation of the number of generations is also reported in Table 1. In order to visualize the spread of the distributions, the three figures in Table 2 displays the FD and SVM boxplots for subjects 1, 2, and 3, respectively (for session 4).

The SVM implementation used solves multi-class problems by means of pairwise classification. SVM are generally better than FD because they return the maximal margin hyperplane which shows very good generalization capabilities. SVM has not been used for fitness computation during the evolutionary process because of its high computational cost, being FD much faster. But linear SVM can be used once the best (B,S) pair has been found by CMA-ES, with the aim of improving results further.

According to a non-parametric Wilcoxon signed-rank test, differences between FD and SVM in Table 1 are statistically significant only for subject 1. For the rest of subjects, FD and SVM perform similarly. It is also noticeable that accuracy test results are highly consistent across runs: the interquartile ranges are smaller than 1% for subjects 1 and 2 and 2% for subject 3.

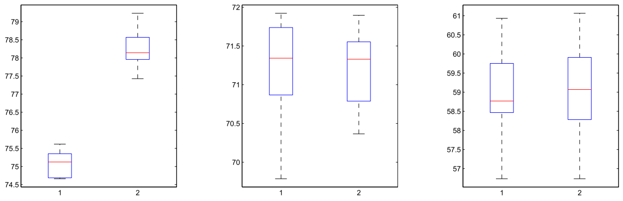
Table 1 also shows that different subjects require different number of frequency-bands: subject 2 requires the least median number of frequency bands (2.0) and Subject 3 needs the highest number (5.0). As we analyze in the next paragraph, it seems that the evolutionary algorithm is able to find the relevant frequency bands for each subject.

Table 3 displays the set of attributes selected for every channel in each of the 10 runs. In order to simplify the notation, numbers will be used to refer to the following bands: 1: [8–10] Hz, 2: [10–12] Hz, 3: [12–14] Hz, 4: [14–16] Hz, 6: [18–20] Hz, 7: [20–22] Hz. First, it can be seen that the frequency bands selected are consistent with known physiological information: SMR-based BCIs use information around the 11 Hz frequency band (and also around 20 Hz and 40 Hz). Frequency bands 1, 2, and 3 have been very often selected by our system for all three subjects. But it has to be remarked that the specific frequency bands also depend on the subject. For instance, band 2 is important for subjects 1 and 2, but not so much for subject 3, where bands 3 and 4 predominate. The method is able to select the most relevant frequency bands for every subject.

Second, it can be seen that there is a strong consistency in the frequency bands selected within subjects across runs. For subject 1, the frequency bands selected are 2 for the first channel and 1, 2, and 7 for the second channel. This happens in 8

[3] Weka's SMO implementation has been used [12].

Table 2
Boxplots for FD (1) and SVM (2) for subjects 1, 2, and 3 (for session 4).



Subject 1 Subject 2 Subject 3

Table 3
Frequency bands Selected. The numbers refer to the following bands: 1: [8–10] Hz, 2: [10–12] Hz, 3: [12–14] Hz, 4: [14–16] Hz, 6: [18–20] Hz, 7: [20–22] Hz.

	Subject 1		Subject 2		Subject 3	
	Channel 1	Channel 2	Channel 1	Channel 2	Channel 1	Channel 2
Run 1	2	2, 3, 7	2	2, 7	1, 3, 4	3, 4
Run 2	2	2, 3, 7	2	2	1, 3	3, 4, 6
Run 3	2	2, 3, 7	2	2	1, 3	3, 4
Run 4	1, 2, 7	3	2	2	3, 4	1, 3, 4
Run 5	2	2, 3, 7	2	2	1, 3, 4	3, 4
Run 6	2	2, 3, 7	2, 7	2	1, 3, 4	3, 4, 6
Run 7	1, 2, 7	3	2	1, 2	1, 3, 4, 6	2, 3, 4
Run 8	2	2, 3, 7	2	1, 2	3, 4	1, 3, 4
Run 9	2	2, 3, 7	1, 2	1, 2	3, 4, 6	2, 3, 4
Run 10	2	2, 3, 7	2	2	3, 4	2, 3, 4

out of the 10 experiments. For subject 2, the selected bands are 2 for channels 1 and 2 in six runs. In the rest of runs, those frequency bands are also selected together with other bands such as 1 and 7 for the first and second channels. Finally, for subject 3 a higher number of bands is selected, but band 3 appears in all runs for the first and second channels and band 4 appears in all runs for the second channel and in 8 runs for the first one.

Next, the evolution of the error corresponding to a single run for all subjects is shown. Figs. 5–7 display a graphical evolution of the classification error on the training, validation, and the test sets for subject 1, 2 and, 3. Let us remember that training and validation sets are obtained from sessions 1, 2, and 3. The test set is given by session 4. It can be seen that evolution stops when the training error becomes stable, and that corresponds also when the test error becomes stable. Every iteration in Figs. 5–7 take approximately 3 min (in a computer with a 2.5 GHz CPU and 4 Gb RAM). The computational cost is high, but the evolution of the filters takes place off-line, and once the filters are available, they can be used in real time without any additional computational effort. In any case, evolutionary computation techniques can be easily paralelized and computation effort reduced drastically.

6.4. Comparative analysis

The purpose of this Section is to compare our results with experiments carried out by researchers that participated in the 2005 BCI-III Competition on the same datasets [4].

First, the classification rates obtained by the evolved filters will be compared with the results obtained with the datasets with precomputed features supplied by the organizers of the competition. Let us remember that the latter dataset was obtained by applying a carefully hand-adjusted surface Laplacian spatial filter and selecting the physiologically appropriate channels, as explained in Section 6.1. The features obtained by this manually tuned filter were of high quality, as demonstrated during the competition: the best results were achieved by using these precomputed features rather than the raw signal dataset [4].[4]

Table 4 compares classification rates for test sets (session 4 for each subject) when datasets are filtered by the evolved filters (from the raw dataset) and when the precomputed features dataset is used. In both cases, SVM was applied to the filtered datasets (training was carried out with session 1, 2, and 3 and testing with session 4, for each subject). Given that the number of frequency-bands selected by the evolved filters is very small, we have also applied a WEKA attribute selection algorithm to the precomputed features dataset [12]. Different algorithms were tested and showed similar results. The

[4] See http://www.bbci.de/competition/iii/results/index.html#martigny. The first 8 best results used the precomputed features (column marked with PSD = Y). The best result obtained directly from the raw data comes only at 9th place (PSD = N).

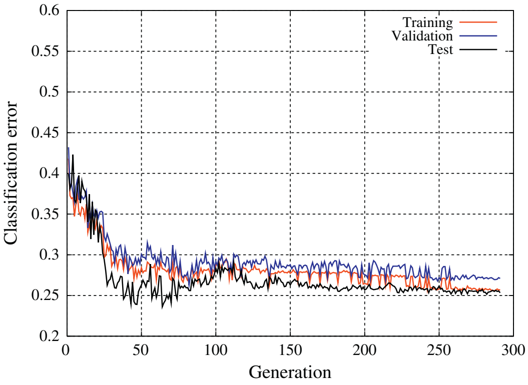


Fig. 5. Evolution of training, validation, and test classification error for subject 1.

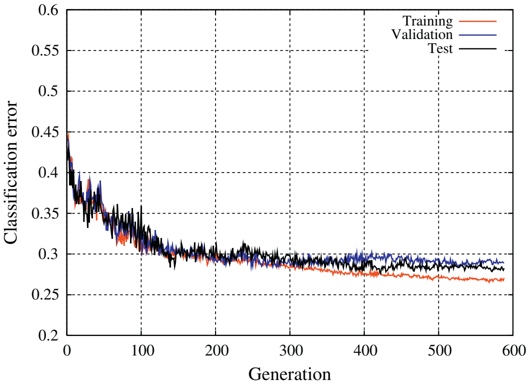


Fig. 6. Evolution of training, validation, and test classification error for subject 2.

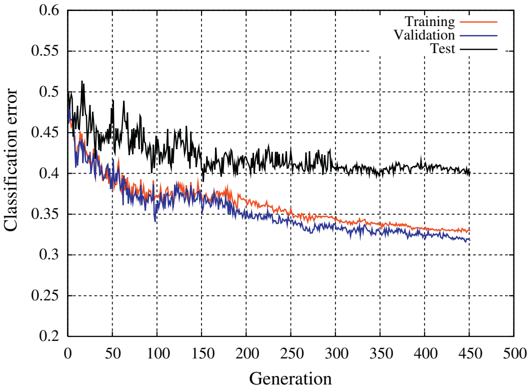


Fig. 7. Evolution of training, validation, and test classification error for subject 3.

selected algorithm (Best First Search + Correlation-based Feature Selection or CFS [13]) searches in the space of subsets of attributes. The algorithm prefers subsets of attributes that correlate well with the class but are not correlated among themselves (the algorithm penalizes redundant attributes). In summary, a best-first search is performed on the space of subsets of attributes so that a small set of non-redundant, highly correlated to class subset is computed [13].

We have also included in the table the success classification rate for session 4 obtained when only the spatial filter is evolved. That is, the system evolves only the spatial filter and all frequency bands [8–10] Hz, [10–12] Hz, . . ., [28–30] Hz, [30–32] Hz are used to build up the classifier. In this case, only the spatial filter is coded in the chromosome and the fitness function does not include the number of bands term, as it is shown in following equation

$$\mathbf{fitness}(S) = \mathbf{MSE}_{FD} \tag{13}$$

Table 4
Comparison between the median classification success rate for session 4 using precomputed features data (with and without attribute selection) and raw data filtered by the evolved filters (with and without frequency-band selection).

	Subject 1	Subject 2	Subject 3
SVM with precomputed features (all frequency-bands)	72.71	60.71	50.14
SVM with evolved spatial filter (all frequency-bands)	75.87	68.97	60.02
SVM with precomputed features + attribute selection.	75.25	65.29	48.42
Number of frequency-bands selected	7	2	3
SVM with evolved spatial + frequency-selection filters.	78.14	71.33	59.07
Number of frequency-bands selected	4	2	5

Table 5
Competition success rate classification (percent, session 4).

	Subject 1	Subject 2	Subject 3
SVM with evolved filters (raw data)	80.0 (1.17)	74.14 (1.09)	60.49 (1.94)
BCI III competition winner (precomputed features)	79.60	70.31	56.02
Best BCI III competition result on raw data	74.31	62.32	51.99

The main result is that the evolved filters (rows 2 and 4) obtain a higher success rate than the precomputed dataset (row 1). This is true both when only the spatial filter is evolved (row 2) and when the spatial and frequency-selection filters are evolved (row 4). Comparing the results when all frequencies bands are used (rows 1 and 2), the success classification rate is better when the evolutionary system is used (row 2). The improvement is very large ($\geqslant$8%) for subjects 2 and 3. According to a Wilcoxon test, all differences are statistically significant at $\alpha = 0.05$.

Also, it can be seen that results tend to improve when a selection of attributes is carried out, at least, for subjects 1 and 2 (see rows 3 and 4 for the precomputed dataset and the evolved-filters dataset, respectively). This improvement is higher when the evolutionary system finds the most relevant frequency bands (row 4) than when attribute selection is used on the precomputed dataset (row 3). This is true for all subjects. For subject 1, the improvement is around 3%. For subjects 2 and 3 the improvement is higher than 5%. All differences are also significant in this case.

Comparing the results when both filters are evolved (row 4) and when only the spatial filter is evolved (row 2), it is observed that the selection of some frequency bands (instead of using all the 24 bands) can help to improve the performance of classifiers for subjects 1 and 2. Results for subject 3 are slightly worse (only 1.08% less accurate). Observed differences are statistically significant. Additionally, the frequency band selection filters provide information about which are the most relevant bands for every subject, as we have seen in the previous subsection.

We will now show that our results are also competitive with those obtained in the BCI-III Competition by other researchers. One of the requirement of the competition was to get a response every 0.5 s. Given that input vectors were computed 16 times per second (i.e. 8 times every half a second), it is necessary to use 8 consecutive samples to compute a response every 0.5 s. In order to do this, our system calculates the predictions for each one of the 8 consecutive samples and returns the majority class. This process tends to improve results because mistakes in some of the predictions are removed by taking the majority class.

Table 5 displays these results. The first row shows the evolved filters results in terms of median and interquartile range (when both the spatial and frequency selection filters are evolved). The second row corresponds to the winner of the competition, who used the precomputed dataset provided by the organizers. The classification algorithm has been described in [11]: it used a multi-class generalization of the Fisher Discriminant for feature reduction and a statistical distance-based algorithm for classification. Most importantly, [11] included a transition detector between mental states. This helped the authors to increase classification rates considerably because by identifying when the mental task has changed from thought A to thought B, it is possible to transform a n-class classification problem into a $(n-1)$-class problem. This is because after the thought transition, thought A can be removed from the set of possible classes, therefore reducing uncertainty about the correct class.

The third row shows the results obtained by Shiliang Sun in the competition [32]. These results are relevant because they used the raw EEG data (similarly to our evolving filters approach), instead of the precomputed data. They applied a multi-class Common Spatial Patterns filter for preprocessing and a SVM for classification.

According to Table 5, the evolved filters are competitive with the winner of the competition (who used high quality precomputed data). This is remarkable, given that it also used a mental task transition detector in addition to preprocessing and classification algorithms. Also, evolved filters perform better than the best competition approach that worked on raw data. Differences are statistically significant for subjects 2 and 3 but not for subject 1.

7. Conclusions

Given the acknowledged importance of preprocessing in brain–computer interfaces, this work has presented an evolutionary approach based on CMA-ES to obtain automatically spatial and frequency selection filters. The aim is to maximize classification accuracy and to minimize the number of frequency-bands used for classification. The system evolves simultaneously both kinds of filters which are adapted for each particular subject.

Validation of this automatic evolution of filters has been done using three datasets, corresponding to three different subjects, acquired in the IDIAP Research Institute that had been previously tested in the 2005 BCI-III competition. From the experiments carried out, we can draw the following conclusions.

First, the frequency bands selected by the evolutionary system are consistent with physiological knowledge, but analysis has also shown that the system takes advantage of the variability among subjects by selecting the appropriate frequency bands for each subject. The evolutionary system is able to obtain high classification accuracies with very few frequency bands. Also, in spite of evolutionary algorithms being stochastic, the set of bands selected for a particular subject is very consistent across runs.

Second, evolved filters are more accurate than those manually tuned. When both spatial and frequency selection filters are evolved, the results are better than those obtained with the high quality manually filtered datasets, even when a selection of attributes is made on the latter. Also, although evolutionary algorithms are stochastic, high accuracies are consistently obtained across runs.

Finally, we have also compared our results with the winners of the BCI-III competition and our results are more accurate than the best competition performer that used the manually filtered data and the best performer that used the raw EEG data.

Acknowledgment

This work has been funded by the Spanish Ministry of Science under Contract TIN2008-06491-C04-03 (MSTAR project) and TIN2011-28336 (MOVES project).

References

[1] R. Aler, I.M. Galván, and J.M. Valls, Evolving spatial and frequency selection filters for brain–computer interfaces, in: IEEE Congress on Evolutionary Computation, 2010, pp. 1–7.

[2] Ali Bashashati, Mehrdad Fatourechi, Rabab K. Ward, Gary E. Birch, A survey of signal processing algorithms in braincomputer interfaces based on electrical brain signals, Journal of Neural Engineering 4 (2007) 32–57.

[3] C.M. Bishop, Pattern Recognition and Machine Learning, The Curse of Dimensionality, Springer, 2006. Chapter: Introduction.

[4] B. Blankertz, G. Dornhege, K-R. Müller, G. Schalk, D. Krusienski, J.R. Wolpaw, A. Schlogl, B. Graimann, G. Pfurtscheller, S. Chiappa, J. del R. Millán, M. Schröder, T. Hinterberger, T.N. Lal, Guido Widman, and Niels Birbaumer, Results of the BCI competition III, in: BCI Meeting 2005, Rensselaerville, New York, 2005. http://www.bbci.de/competition/iii/results/bci_competition_iii_results_list.pdf.

[5] E.A. Curran, M.J. Stokes, Learning to control brain activity: a review of the production and control of EEG components for driving braincomputer interface (BCI) systems, Brain Cognition 51 (2003).

[6] G. Dornhege, B. Blankertz, M. Krauledat, F. Losch, G. Curio, K.R. Muller, Combined optimization of spatial and temporal filters for improving braincomputer interfacing, IEEE Transactions on Biomedical Engineering 53 (11) (2006) 2274–2281.

[7] G. Dornhege, M. Krauledat, K-R. Muller, B. Blankertz, Toward Brain–Computer Interfacing, MIT Press, 2007. pp. 207–234. Chapter: General Signal Processing and Machine Learning Tools for BCI Analysis.

[8] R.A. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 7 (7) (1936) 179–188.

[9] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, 1990.

[10] C. Neuper, G. Pfurtscheller, N. Birbaumer, Motor Cortex in Voluntary Movements, CRC Press, 2005. pp. 367–401 (Chapter 14).

[11] Ferrán Galán, Francesc Oliva, Joan Guardia, Using mental tasks transitions detection to improve spontaneous mental activity classification, Medical and Biological Engineering and Computing 45 (6) (2007). 1741-0444.

[12] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten, The WEKA data mining software: an update, SIGKDD Explorations 11 (1) (2009) 10–18.

[13] Mark A. Hall and Lloyd A. Smith, Feature selection for machine learning. comparing a correlation-based filter approach to the wrapper, in: Twelfth International Florida Artificial Intelligence Research Society Conference (FLAIRS 99), 1998, pp. 235–239.

[14] N. Hansen and A. Ostermeier, Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation, in: Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, Citeseer, 1996, pp. 312–317.

[15] N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies, Evolutionary Computation 9 (2) (2001) 159–195.

[16] Nikolaus Hansen, The CMA Evolution Strategy: A Tutorial, Technische Universitat Berlin, TU Berlin, 2009.

[17] Bashar Awwad Shiekh Hasan, John Q. Gan, Temporal modeling of eeg during self-paced hand movement and its application in onset detection, Journal of Neural Engineering 8 (6) (2011) 1–8.

[18] A. Kubler, K-R. Müller, Toward Brain–Computer Interfacing, MIT Press, 2007. pp. 1–26. Chapter: An Introduction to Brain–Computer Interfacing.

[19] S. Lemm, B. Blankertz, G. Curio, K.R. Muller, Spatio-spectral filters for improved classification of single trial EEG, IEEE Transactions on Biomedical Engineering 52 (9) (2005) 1541–1548.

[20] Fabien Lotte, Marco Congedo, Anatole Lécuyer, Fabrice Lamarche, Bruno Arnaldi, A review of classification algorithms for EEG-based brain–computer interfaces, Journal of Neural Engineering 4 (2007).

[21] J. del R. Millán, On the need for on-line learning in brain–computer interfaces, in: Proceedings of the International Joint Conference on Neural Networks, Budapest, Hungary, July 2004. IDIAP-RR 03-30.

[22] J. Mourino, J. del R. Millan, F. Renkens, W. Gerstner, Noninvasive brain-actuated control of a mobile robot by human EEG, IEEE Transactions on Biomedical Engineering 51 (2004).

[23] B.S. Moon, H.C. Lee, Y.H. Lee, J.C. Park, I.S. Oh, J.W. Lee, Fuzzy systems to process ecg and eeg signals for quantification of the mental workload, Information Sciences 142 (14) (2002) 23–35.

[24] K.R. Muller, C.W. Anderson, G.E. Birch, Linear and nonlinear methods for brain–computer interfaces, IEEE Transactions on Neural Systems and Rehabilitation Engineering 11 (2) (2003) 165–169.

[25] K.-R. Muller, C.W. Anderson, G.E. Birch, Linear and nonlinear methods for brain–computer interfaces, IEEE Transactions on Neural Systems and Rehabilitation Engineering 11 (2) (2003) 165–169.

[26] Klaus-Robert Muller, Michael Tangermann, Guido Dornhege, Matthias Krauledat, Gabriel Curio, Benjamin Blankertz, Machine learning for real-time single-trial eeg-analysis: from brain–computer interfacing to mental state monitoring, Journal of Neuroscience Methods 167 (1) (2008) 82–90.

[27] A. Ostermeier, A. Gawelczyk, N. Hansen, A derandomized approach to self-adaptation of evolution strategies, Evolutionary Computation 4 (2) (1994) 369–380.

[28] G. Pfurtscheller, F.H.L. da Silva, Event-related synchronization of mu rhythm in the EEG over the cortical hand area in man, NeuroScience Letters 174 (1994).

[29] G. Saggio, P. Cavallo, A. Ferretti, F. Garzoli, L.R. Quitadamo, M.G. Marciani, F. Giannini, and L. Bianchi, Comparison of two different classifiers for mental tasks-based brain–computer interface: Mlp neural networks vs. fuzzy logic, in: World of Wireless, Mobile and Multimedia Networks Workshops, 2009. WoWMoM 2009. IEEE International Symposium on a, pages 1 –5, june 2009.

[30] Hans-Georg Stark, Wavelets and Signal Processing: An Application-Based Introduction, Springer, 2005.

[31] Heung-Il Suk and Seong-Whan Lee, Two-layer hidden markov models for multi-class motor imagery classification. In Brain Decoding: Pattern Recognition Challenges in Neuroimaging (WBD), 2010 First Workshop on, pages 5 –8, aug. 2010.

[32] Shiliang Sun, Changshui Zhang, and Jie Pan, Algorithm submitted to the BCI competition III, in: BCI Meeting 2005, Rensselaerville, New York, 2005.

[33] R. Tomioka, G. Dornhege, G. Nolte, B. Blankertz, K. Aihara, and K.R. Muller, Spectrally weighted common spatial pattern algorithm for single trial EEG classification. Technical Report 40, Department of Mathematical Engineering, University of Tokyo, Japan, 2006.

[34] G. Yang, Y. Lin, A driver fatigue recognition model based on information fusion and dynamic bayesian network, Information Sciences 180 (10) (2010) 1942–1954.

[35] Mingjun Zhong, Fabien Lotte, Mark Girolami, Anatole Lcuyer, Classifying eeg for brain–computer interfaces using gaussian processes, Pattern Recognition Letters 29 (3) (2008) 354–359.

[36] Shang-Ming Zhou, John Q. Gan, Francisco Sepulveda, Classifying mental tasks based on features of higher-order statistics from eeg signals in braincomputer interface, Information Sciences 178 (6) (2008) 1629–1640.