Introducing the Use of Model-Based Evolutionary Algorithms for EEG-Based Motor Imagery Classification

Roberto Santana
Intelligent Systems Group
University of the Basque Country (UPV/EHU)
roberto.santana@ehu.es

Laurent Bonnet
INRIA-Rennes
Campus Universitaire de Beaulieu
F-35042 Rennes Cedex, France
laurent.bonnet@inria.fr

Jozef Legény
INRIA-Rennes
Campus Universitaire de Beaulieu
F-35042 Rennes Cedex, France
jozef.legeny@inria.fr

Anatole Lécuyer
INRIA-Rennes
Campus Universitaire de Beaulieu
F-35042 Rennes Cedex, France
anatole.lecuyer@inria.fr

ABSTRACT

Brain computer interfaces (BCIs) allow the direct human-computer interaction without the need of motor intervention. To properly and efficiently decode brain signals into computer commands the application of machine-learning techniques is required. Evolutionary algorithms have been increasingly applied in different steps of BCI implementations. In this paper we introduce the use of the covariance matrix adaptation evolution strategy (CMA-ES) for BCI systems based on motor imagery. The optimization algorithm is used to evolve linear classifiers able to outperform other traditional classifiers. We also analyze the role of modeling variables interactions for additional insight in the understanding of the BCI paradigms.

Categories and Subject Descriptors

H.1 [**Information Systems**]: Models and Principles—*User-Machine Systems*; I.2.8 [**Computing Methodologies**]: Artificial Intelligence—*Problem Solving, Control Methods, and Search*; I.5.4 [**Computing Methodologies**]: Pattern Recognition—*Signal processing*

General Terms

Heuristic methods

GECCO'12, July 7-11, 2012, Philadelphia, Pennsylvania, USA.

Keywords

brain computer interface, model-based evolutionary algorithms, CMA-ES, optimization, motor-imagery

1. INTRODUCTION

Brain computer interfaces (BCIs) [15, 28] are used to translate electrical signals into commands without the need for motor intervention. They are particularly useful for implementing assistive technologies providing communication and control to people with severe muscular or neural disabilities [11]. More recently BCIs have also found application in domains such as gaming [18], virtual reality environments [22], and space applications [23].

BCIs require a decoding component in which brain signals are translated into commands. Usually, classification algorithms are applied to predict the human intention from the analysis of the signals. The choice of the classification algorithm depends on many factors such as the BCI paradigm and type of recorded data (e.g. electroencephalography (EEG), magnetoencelography (MEG), etc.). Several classification algorithms have been used to analyze brain data in the context of BCI applications [16]. They include linear discriminant classifiers (LDA) [6], support vector machines (SVMs) [21], neural networks (NNs) [10], and other classification methods [16]. The use of machine learning in BCI techniques is not constrained to the use of classification methods because several other tasks such as channel selection, human and BCI adaptation, etc., require the implementation of efficient and adaptive procedures.

Evolutionary algorithms (EAs) have been increasingly applied within different BCI frameworks [4, 5, 9, 20, 25]. They have been mainly used for supporting different stages of the classification process (e.g. feature selection, classifier training and evaluation, etc.) and have been shown to improve the classification accuracy for a variety of BCI paradigms.

One of the limitations of classical evolutionary algorithms and other optimization methods is that they do not provide a model of the search space or fitness function being optimized. Usually in EAs, genetic operators can effectively move the search to an area where optimal solutions are but nothing is learned from the, potentially relevant, relationships between the problem variables. In some real-world optimization problems, the patterns shared by the optimal solutions and the relationships between the problem variables can be highly relevant for understanding the problem being solved. In BCI, the relationships between the different brain areas from which signals are collected or between the different frequencies in which information is contained can be useful for understanding the neural processes involved in the given BCI paradigm.

There is a class of evolutionary algorithms that models the relationships between the variables and exploit the modeled relationships to orient the search to more promising areas of the search space. Examples of these methods are estimation of distribution algorithms (EDAs) [14, 17, 19] and the covariance matrix adaptation evolution strategy (CMA-ES) [7]. Both classes of algorithms use probabilistic modeling of the solutions to implement a more efficient search.

In this paper we investigate the performance of CMA-ES as a classification component of a BCI based on the Graz protocol. In the Graz protocol, EEG signals are recorded from sensorimotor areas during mental imagination of specific movements. In particular, we address an off-line BCI scheme in which recorded data from left hand and right hand imagery movements was used and the classifier task was to predict one of these two conditions from the signals. The CMA-ES algorithm was adapted to work within OpenViBE [22], a highly modular and portable software platform for BCI implementation. Then, two different scenarios that implement the Graz protocol were evaluated and the performance of different classifiers were compared.

The purpose of this paper is threefold. First, we compare the ability of CMA-ES as classification method to other classifiers usually applied for BCI. Second, we evaluate the feasibility of incorporating model-based evolutionary algorithms as independent modules of modular software platforms such as OpenViBE. Finally, we analyze the models produced by CMA-ES in order to unveil the potentially informative interactions captured by the algorithm.

The paper is organized as follows: The next section discuses related work on the application of evolutionary algorithms to BCIs. Section 3 briefly describes the BCI paradigm used in our experiment and the main steps for data acquisition and signal processing. Section 4 presents a formulation of the optimization problem and introduces the optimization algorithms. Section 5 describes the experimental framework to evaluate our proposal and presents the numerical results. The main contributions of the paper are summarized in Section 6 where some lines for future research are also discussed.

2. RELATED WORK

Early work on the application of EAs to BCIs was presented in [4] where a genetic algorithm (GA) was simultaneously applied to feature selection and classifier optimization. The task goal was brain-activity based dictation of characters and the used BCI paradigm was P300. The fitness function was the average difference between the polynomial classifier output and the correct output. Authors achieved classification accuracies up to 87%. More recently, Poli et al. [20] describe the use of an EA for the design of an analogue BCI mouse. The BCI is based on the P300 paradigm and the EA is used for feature selection and for training an adaptive system using them. Although the system needs to be adapted and optimized by an EA before a user can use it, the authors claim that the evolution system makes it possible to control the pointer for a person having undergone no previous training.

GAs have been also applied to asynchronous BCI where brain signals are continuously monitored and the system is activated only when the user intends control. In [5], a hybrid GA (simple GA combined with a local search method) is used for customization of the system for a specific user. Significant improvements in the classification accuracy for the two subjects considered were reported. In [25] different variants of evolutionary classifiers were applied to the identification of the P300 component from EEG data. The optimizers were used to evolve sets of solutions that improve the accuracy of single classifiers. The introduced approaches were able to achieve a classification rate over 80% on test data.

Multi-objective algorithms have been also applied to BCI. In [9], two EAs were applied to solve the problem of channel selection in the classification of continuous EEG without trial structure. The use of multi-objective optimization contributed to improve the classification accuracy by selecting an informative set of channels. In [24], a multi-objective GA and a multi-objective EDA are used to simultaneously maximize classifier accuracy in multiple subjects in the classification of task-related mental activity from MEG data. The evolutionary algorithms were able to improve the classification accuracy compared to approaches whose classifiers use only one type of MEG information or for which the set of channels is fixed a priori.

3. MOTOR IMAGERY DATA

A BCI based on "motor imagery" makes use of changes in oscillatory EEG activity induced by various motor imageries performed by the user, e.g. related to the imagination of right hand, left hand or feet motion. These changes are detected in the signals and transformed into control signals for external devices. In this paper we addressed a classification problem that arises in the use of this paradigm.

The data used in our experiments was collected from a 25-year-old male subject using two different experimental conditions. The first experimental setting (motor-imagery) corresponded to imagery movement of the right vs. left hand. In the second setting (motor-movement), the subject was instructed to actually execute the movement of the right vs. left hand given a stimulus presentation. One session consisted of 40 left trials and 40 right trials (order randomly chosen) with 8 minutes per session.

Each trial consisted of the following steps[1]: At time $t = 0$ a cross was shown on screen. At time $t = 1s$ the subject received the instruction (right or left arrow). Feedback was on at time $t = 2.25s$ and disappeared at time $t = 6s$. The inter-trial duration was randomly selected between 1.5 and 3.5 seconds. Stimulus was presented on a desktop computer, $19'$ screen and EEG signal was recorded with a g.Tec g.USBAmp amplifier (512 Hz).

3.1 Signal analysis

[1]Images of the instruction and feedback are available at: http://openvibe.inria.fr/motor-imagery-bci/

The signal analysis relied on the common spatial pattern (CSP) [3] to compute spatial filters over different selected bands. The goal of CSP is to improve the discrimination of two types of signals. The spatial filters are constructed in a way that they maximize the variance for signals of the first condition while at the same time they minimize it for the second condition. The different stages of the signal analysis procedure follows:

- Ten active electrodes were located over the motor cortex of the hands: C3;C4;FC3;FC4;C5;C1;C2;C6;CP3;CP4.
- All channels were re-referenced to a single channel (which was located on the ear).
- A spatial filter (surface laplacian filter) was applied to the ten channels producing two outputs (one for left and one for the right).
- Each of the two output signals was filtered in one of six following frequency ranges: $\{[8, 12], [12, 16], [16, 20], [20, 24], [24, 28], [28, 32]\}$.
- Four seconds of signals, half a second after the instruction was shown to the user, were selected.
- The signal is then splitted in blocks of 1 second every 16th second.
- The logarithmic band power was computed and the matrices were converted into feature vectors.
- For each trial, the feature vector comprises 12 components, 6 corresponding to each output signal.

The classifier task is to infer the correct answer from the feature vector associated to each condition.

4. USING OPTIMIZATION FOR CLASSIFICATION

Let $\mathbf{X} = (X_1, \dots, X_n)$ be a set of n continuous variables. Each variable X_i represents a different problem feature. $\mathbf{x} = (x_1, \dots, x_n)$ is a possible assignment to the variables and $\mathbf{c} \in \{1, 2\}$ is the class assigned to a given set of features $\mathbf{x}$. The problem of predicting the class of $\mathbf{x}$ can be posed as the problem of finding a hyperplane H, as linear combination of the features, that allows to predict the class.

$$H : a_0 + \sum_{i=1}^{n} a_i x_i = 0 \qquad (1)$$

where a_i are the coefficients or parameters of the hyperplane.

The way to use the hyperplane for classification is straightforward: A vector $\mathbf{x}$ is classified as class 1 if $a_0 + \sum_{i=1}^{n} a_i x_i > 0$ otherwise it is classified as class 2.

An approach to compute the set of parameters a_i that maximizes the accuracy of the classification is to use linear discriminant analysis (LDA). In LDA, the solutions of the coefficients are found as:

$$a_0 = log(\frac{\pi_1}{\pi_2}) - \frac{1}{2}(\mu_1 + \mu_2)^T \sum^{-1}(\mu_1 - \mu_2) \qquad (2)$$

and

$$(a_1, \dots, a_n)^T = \sum^{-1}(\mu_1 - \mu_2) \qquad (3)$$

where $\pi_1 = \frac{N_1}{N}$, $\pi_2 = \frac{N_2}{N}$, $\mu_1 + \mu_2$ are the vectors of means and $\sum$ is the covariance matrix. LDA assumes that the independent variables are normally distributed.

In some cases the data distribution does not satisfy the normality assumptions of LDA. In these cases other types of models, like logistic regression, could be used to find a combination of features that separate the classes. Here we investigate a different alternative, in which the linear coefficients are found by direct optimization of the classification accuracy.

The optimization problem is then formalized as the search of the optimal parameters $(a_0, a_1, \dots, a_n)$ that maximize the accuracy function $f(a_0, a_1, \dots, a_{n+1})$ where $f(\mathbf{a})$ is computed as the percentage of cases correctly classified as predicted by the hyperplane $H(\mathbf{a})$ defined by $\mathbf{a}$.

Basically, the optimization algorithm should be able to find a combination of parameters that maximizes the classification accuracy.

4.1 Optimization algorithms

We have used two different optimization methods that differ in their complexity and the principles they employ to organize the search of the optimal solution.

The first algorithm corresponds to a simple random hill climbing method (RHC) previously used in [25] in the context of P300 speller classification [12]. The RHC algorithm's pseudocode is described in Algorithm 1. RHC works by modifying a single variable of the current solution within an interval specified by a parameter L. If the change to the variable improves the function value, then the new solution is taken as the best. The algorithm stops when a maximum number of trials, corresponding the number of evaluations, have been found. We use a RHC with restarts. In this version of the algorithm, *ntrials* calls to the RHC are done and the algorithms outputs the best solution out of the *ntrials* RHC runs.

Algorithm 1: **Random Hill Climbing**

```
Randomly generate an initial solution y.
do {
    Randomly select a variable Y_i
    Generate a random value α ∈ (y_i − L, y_i + L)
    Create a new solution ŷ such that ŷ_i = α and ŷ_j = y_j, ∀j ≠ i.
    If f(ŷ) ≥ f(y) then y = ŷ
} until Maximum number of changes is achieved
Return y.
```

CMA-ES [7, 8, 1] is a model-based EA conceived for non-linear non-convex optimization problems in continuous domains. The information about the optimization process is modeled using a multivariate normal distribution. New candidate solutions are sampled according to this distribution. Pairwise dependencies between the variables in this distribution are represented by a covariance matrix. The covariance matrix adaptation (CMA) is a method to update the covariance matrix of this distribution. This is particularly useful, if the fitness function is ill-conditioned. The modeling steps contribute to accelerate the convergence of the algorithm to promising areas of the search space.

There are two fundamental steps that serve to explain the rationale of the algorithm. First, the mean of the distribution is updated such that the likelihood of previously successful candidate solutions is maximized. Second, the covariance matrix of the distribution is updated (incrementally) such that the likelihood of previously successful search steps is increased. Both updates can be interpreted as a natural gradient descent. Algorithm 2 presents a pseudocode of the basic CMA-ES steps as described in [2]. We use an advanced implementation of CMA-ES[2] that incorporates the use of a restart mechanism to avoid convergence to local solutions [1].

Algorithm 2: **CMA-ES**

---

1 Generate an initial random solution.

2 The λ offspring at $g+1$ generation are sampled from a Gaussian distribution using covariance matrix and global step size computed at generation g.

$$x_k^{g+1} = z_k, z_k = N\left(\langle x\rangle_\mu^{(g)}, {\sigma^{(g)}}^2\mathbb{C}^{(g)}\right), k = 1,\ldots,\lambda \quad (4)$$

where $\langle x\rangle_\mu^{(g)} = \sum_{i=1}^{\mu} x_i^{(g)}$ with μ being the selected best individuals from the population.

3 The evolution path $\mathbb{P}_c^{(g+1)}$ is computed as follows:

$$\mathbb{P}_c^{(g+1)} = (1-c_c).\mathbb{P}_c^{(g)} + \sqrt{c_c(2-c_c)}.\frac{\sqrt{\mu}}{\sigma^{(g)}}G \quad (5)$$

$$\mathbb{C}^{(g+1)} = (1-c_{cov}).\mathbb{C}^{(g)} + c_{cov}.\left(\frac{1}{\mu}F + (1-\frac{1}{\mu})\frac{1}{\mu}\sum_{i=1}^{\mu}\frac{1}{{\sigma^{(g)}}^2}HI\right) \quad (6)$$

where $G = \left(\langle x\rangle_\mu^{(g+1)} - \langle x\rangle_\mu^{(g)}\right)$, $F = (\mathbb{P}_c^{g+1})(\mathbb{P}_c^{g+1})^T$, $H = (x_i^{(g+1)} - \langle x\rangle_\mu^{(g)})$, and $I = (x_i^{(g+1)} - \langle x\rangle_\mu^{(g)})^T$ The strategy parameter $c_{cov} \in [0,1]$ determines the rate of change of the covariance matrix $\mathbb{C}$.

4 Adaptation of global step size $\sigma^{(g+1)}$ is based on a conjugate evolution path $\mathbb{P}_c^{(g+1)}$.

$$\mathbb{P}_c^{(g+1)} = (1-c_\sigma).\mathbb{P}_\sigma^g + \sqrt{c_\sigma(2-c_c)}.\mathbb{B}^{(g)}(\mathbb{B}^{(g)})^{-1}\mathbb{B}^{(g)}\frac{\sqrt{\mu}}{\sigma^{(g)}}G \quad (7)$$

the matrices $\mathbb{B}^{(g)}$ and $\mathbb{D}^{(g)}$ are obtained through a principal component analysis:

$$\mathbb{C}^{(g)} = \mathbb{B}^{(g)}(\mathbb{D}^{(g)})^2(\mathbb{B}^{(g)})^T \quad (8)$$

where the columns of $\mathbb{B}^{(g)}$ are the normalized eigen vectors of $\mathbb{C}^{(g)}$, and $\mathbb{D}^{(g)}$ is the diagonal matrix of the square roots of the eigen values of $\mathbb{C}^{(g)}$. The global step size $\sigma^{(g+1)}$ is determined by

$$\sigma^{(g+1)} = \sigma^{(g)}exp\left(\frac{c_\sigma}{d}\left(\frac{||\mathbb{P}_c^{(g+1)}||}{E(||N(0,\mathbb{I}||))}\right) - 1\right) \quad (9)$$

5 Repeat Steps $2-4$ until a maximum number of function evaluations are reached.

4.2 Factors that influence the optimization approach to classification

There are a number of factors that influence the performance of the optimization algorithms as classifiers. These factors include issues related to the characteristics of the classification problem, such as the number of features and the number of cases, and issues related to the characteristics of the optimization algorithm such as the maximum number of allowed objective function evaluations.

The number of variables is an important factor in the scalability of optimization algorithms. Some optimizations methods may be efficient for a small number of variables and deteriorate their performance when more variables are added. The choice of the optimization-based classification method may thus consider which is the number of features involved in the problem. Similarly, the number of cases of the classification problem influences the time spent in the evaluation of the fitness function.

Choosing an initial solution closer to the basin of attraction of optimal solutions can accelerate the convergence of certain optimization algorithms. This facts opens the possibility of using the set of linear coefficients output by LDA and other methods as an initial point of the optimization algorithm. In Section 5 we investigate this possibility by evaluating the results of the CMA-ES-LDA method.

The type of optimization algorithm used to find the solution will also influence the quality of the final classifier. Since the classifiers will be applied to very diverse sets of data (depending on the characteristics of the experiments the classification features have been extracted from), the optimization algorithms should be robust in different fitness landscapes. Efficiency is another important factor because k-folding strategies, usually applied to assess the quality of the classifiers, determine that the search for the optimal solution will be repeated k times. This implies to solve k different optimization problems to obtain a k-folded based prediction of the classifier accuracy.

The maximum number of evaluations allowed to the optimization algorithm also affects the quality of the final classifier. On one hand, a very small number of evaluations may determine that the solutions are far from giving a high classification accuracy. On the other hand, allowing too many function evaluations may contribute to improve the quality of the solution on the training data but at the expense of obtaining an overfitted classifier.

5. EVALUATION

In this section, we compare the results of the optimization-based classifiers to the results achieved using classical and most-common used classifiers in the BCI community [16]. The classifiers we proposed are based on the direct optimization of the hyperplane parameters and the method that uses CMA-ES is able to build a model of the variables interactions to guarantee a more efficient search. We evaluate the behavior of RHC, CMA-ES and CMA-ES-LDA for the motor-movement and motor-imagery scenarios described in Section 3 using different parameters. We later analyze the covariance matrices produced by CMA-ES and identify the differences in the interactions between the variables for the two scenarios.

5.1 Experimental apparatus and parameters of the algorithms

There are many software tools for off-line and online analysis of EEG and biomedical signals that can be used for designing and implementing BCIs [26]. For the acquisition, signal processing and visual application study we used OpenViBE [22], a software platform which enables researchers to design, test, and use BCIs. OpenViBE is portable, independent of the hardware or software, and is based on free and

---

[2]http://www.lri.fr/$\tilde$hansen/cmaes$\
_$inmatlab.html#C

open source software. Modularity is a particularly important OpenViBE's feature since it helps to easily integrate and test different methods for signal analysis and classification. Particularly relevant for the application presented in this paper is the classification module.

In principle, optimization could be conceived as an independent OpenViBE module that could be attached to different processes involved in the BCI scenarios (e.g. channel selection, feature selection, etc.). However, such a general module should be able to handle with a number of constraints since the objective functions and problem representations change between problem domains. Therefore, we focused on the implementation of optimization strategies for the classification module.

In OpenViBE's classification module the feature vector is fed into a (user selected) classifier that assigns a class to each feature vector, this class being an identifier of the brain signal that has been recognized. In general, the classifier is trained before-hand using a set of feature vectors from each class. It is during this training phase that our optimization approach will be applied.

For comparison, we used the LDA and SVM implementations included in the classification module of OpenViBE. The C language CMA-ES implementation[3] was adapted to the OpenViBE platform. The parameters used for the SVM implementation were those proposed by default in OpenViBE.

The range of values for generating the initial solutions was $[L_i - L, L_i + L]$ were $L \in \{50, 100\}$. For RHC and CMA-ES, $L_i = 0 \ \forall i \in \{0, \ldots, n\}$. For CMA-ES-LDA, L_i was the ith component of the solution given by the application of LDA. Different numbers of restarts $Ts \in \{3, 5, 10\}$ were allowed. Restarts were used as a way to scape from local optima. Following the setting recommended [1], after each restart of CMA-ES the population size is increased by a factor 2. The same increased is applied to restarts of RHC and CMA-ES-LDA. We did not tune the parameters of any of the classification methods used in our comparisons.

For all algorithms and parameters configuration, five trials were run and the maximum, mean, and minimum accuracies were computed. Notice that each trial corresponds to a 5-fold-cross validation of the found accuracy. Since to compute the classifier in each fold the optimization algorithm is applied, there are 25 optimization runs for each parameter configuration of the algorithms.

5.2 Classification results

The first objective of our analysis is to compare the performance of the classifiers for the data obtained from the subject for the motor-movement scenario. Results are shown in Table 1. In the table, L and Ts are respectively the parameters for the number of restarts and the range of the variables. Max, $Mean$, Min, and *std* respectively refer to the maximum, mean, minimum and standard deviations of the accuracies as computed from the five different repetitions.

It can be seen in Table 1 that the accuracies are in general over 90% for all the classifiers except SVM. The highest accuracies are obtained by CMA-ES when the initial range of parameters is $L = 50$ (these values appear in bold in the table). In addition, CMA-ES and CMA-ES-LDA seem to achieve better parameters than all the other classifiers for every possible configuration of parameters. The improvements over the results achieved by LDA are represented in Figure 1. Two aspects can be highlighted from the analysis of this figure. The first is that the RHC algorithm is not able to reach the accuracy results of LDA. A simple optimization procedure that does not take into account the regularities of the search space is not expected to produce consistent high accuracy results for this problem, at least for the brain data obtained from this single subject. The other observation is that CMA-ES-LDA does not improve the results of CMA-ES. Therefore, at least for this example, an initialization in the neighborhood of the solution obtained by LDA does not seem to improve the accuracy results.

We then analyze the behavior of the algorithms for the motor-imagery scenario. The same parameters are used for all the algorithms. Results are shown in Table 2. A first observation is that the accuracy results are much lower that for the motor-movement scenario. This phenomenon could be due to a stronger discriminative brain signal associated to the actual movements than to imagined movements. It can be seen in the table that CMA-ES outperforms all the other algorithms. For this scenario, SVM and RHC are able

Alg	L	Ts	Max	$Mean$	Min	*std*
LDA			93.16	92.92	92.75	0.17
SVM			89.95	89.81	89.69	0.10
CMA-ES	50	3	**94.08**	**93.80**	**93.41**	0.27
		5	93.67	93.22	92.81	0.36
	100	3	93.88	93.09	92.40	0.56
		5	93.92	93.49	93.01	0.33
RHC	50	3	92.95	91.70	89.23	1.43
		5	91.63	91.14	90.46	0.58
		10	92.65	91.94	91.17	0.64
	100	3	92.35	90.86	87.19	2.17
		5	92.70	91.98	91.07	0.66
		10	93.16	91.24	89.80	1.56
CMA-ES-LDA	50	3	93.83	93.38	92.91	0.38
		5	93.42	93.12	92.91	0.20

Table 1: Results of the different classifiers for the motor-movement scenario.

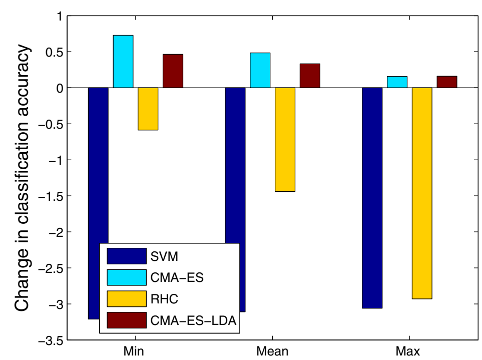

Figure 1: Improvement of the classification algorithms over LDA for the motor-movement scenario.

[3]Available from the author's website `http://www.lri.fr/~hansen/cmaes_inmatlab.html`

Alg	*L*	*Ts*	*Best*	*Mean*	*Worst*	*std*
LDA			71.99	71.68	71.30	0.25
SVM			72.86	72.53	71.79	0.43
CMA-ES	50	3	75.56	74.90	74.49	0.40
		5	75.56	**75.11**	**74.74**	0.33
	100	3	**75.97**	74.61	73.88	0.90
		5	75.82	74.73	73.97	0.73
RHC	50	3	74.28	72.15	70.61	1.35
		5	72.40	71.41	70.36	0.80
		10	72.70	71.37	70.24	0.95
	100	3	73.11	71.44	68.42	1.91
		5	73.57	71.38	69.95	1.37
		10	73.01	71.82	69.54	1.34
CMA-ES-LDA	50	3	75.30	74.77	73.52	0.72
		5	75.91	74.99	74.34	0.68

Table 2: Results of the different classifiers for the motor-imagery scenario.

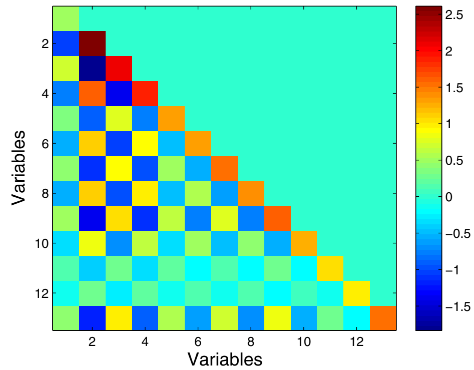

Figure 3: Covariance matrix computed during the application of CMA-ES to the motor-movement scenario

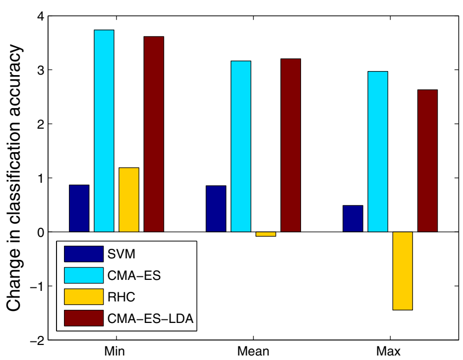

Figure 2: Improvement of the classification algorithms over LDA for the motor-imagery scenario

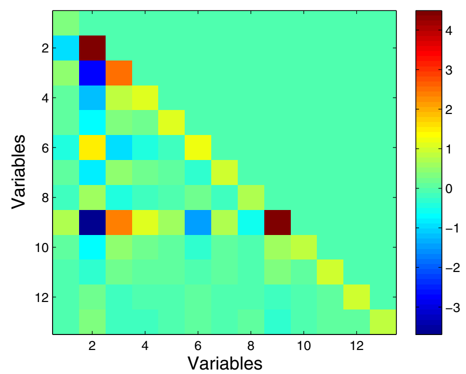

Figure 4: Covariance matrix computed during the application of CMA-ES to the motor-imagery scenario

to improve the results of LDA. It is particularly noticeable that RHC achieves maximum accuracies higher than SVM. Nevertheless, RHC is not a consistent good classifier. It has the highest standard deviation and the lowest mean accuracy among all the algorithms. This can be further appreciated in Figure 2 where it is the only algorithm whose performance is worse than LDA. It is worth to notice that the average improvements given by CMA-ES and CMA-ES-LDA are around the 3% of accuracy.

5.3 Analysis of the models

The next goal is to investigate the information captured in the covariance matrices learned by CMA-ES in the solution of the optimization problems. The objective behind the analysis of the matrices is threefold. First, we would like to identify differences between the variances computed for the different variables that could hint to their different roles in the optimization process. Second, we would like to detect possible interactions between the variables expressed in a strong negative or positive covariance. Finally, we would like to know if the differences between the motor-imagery and motor-movement scenarios are somewhat reflected in differences between the models captured for the two problems. We recall that 12 of the variables of the problem (corresponding to coefficients $(a_1, a_2, \dots, a_{12})$ of the hyperplane correspond to the features extracted from the CSP computed for different frequency bands and conditions. Therefore, the covariances between the variables could be an indicator of the relationships between the output of the CSPs.

To analyze the covariance matrix, we run CMA-ES allowing 10 possible restarts for each of the 5 folds corresponding to each scenario. We saved the last covariance matrix learned by the algorithm. Therefore, we obtained 50 covariance matrices learned from different CMA-ES initializations and folds. These matrices should be useful to detect whether any possible pattern in the covariances was consistent between datasets and runs of the algorithm. In addition, we run a CMA-ES, also with 10 restarts using the whole data set (i.e. all folds together) and similarly saved the last covariance matrix of each restart.

Figures 3 and 4 respectively show one of the covariance matrices learned for each scenario using the complete dataset. The colorbar represents the covariances strengh. For the sake of clarity we only display the lower triangular matri-

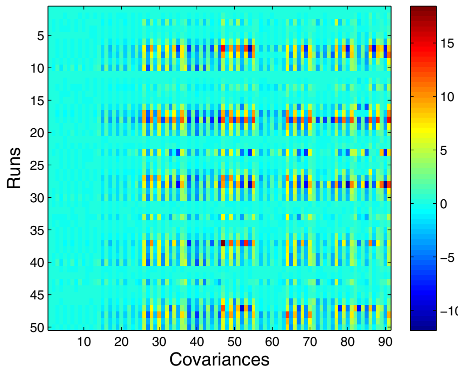

Figure 5: Covariance values for different runs of CMA-ES in its application to the motor-movement scenario

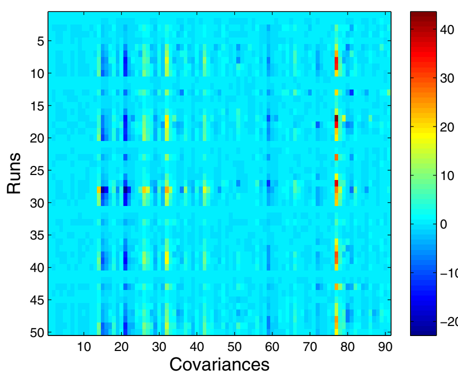

Figure 6: Covariance values for different runs of CMA-ES in its application to the motor-imagery scenario

ces. There are clear differences between the covariance matrices corresponding to the two scenarios. For the motor-movement scenario there is a checkerboard pattern of interaction between variables in which adjacent variables in the representation tend to have a negative covariance while variables at a distance two exhibit a positive covariance. On the contrary, for the motor-imagery scenario covariances can be very strong but are mainly limited to a strong positive variance for variables 2 and 9 and strong negative covariances between variables pairs $(2, 3)$ and $(2, 9)$. A high variance for one of the variables may indicate that different values of this variable are possible in good classifiers. The negative covariance may indicate how changes in one of the variables are somewhat "compensated" by changes in the correlated variables.

To investigate whether the patterns shown in Figures 3 and 4 are consistent, we inspected the covariance matrices learned in the 50 CMA-ES runs. The $\frac{13 \cdot 14}{2} = 91$ possible covariance values obtained at the end of the 50 runs are shown in Figures 5 and 6. The checkerboard pattern can be appreciated in Figure 5 in the form of differently colored stripes. The effect of the restarts can also be seen in the figures. In the first runs of the algorithm, when the population size is small, the algorithm is hardly able to detect any covariance between the variables, as the population is increased the same patterns of interactions appear. These patterns are repeated 5 times, one for each fold. Figure 6 shows that the distinctive patterns detected for the motor-imagery scenario are also repeated for the different runs and across the folds.

6. CONCLUSION AND FUTURE WORK

In this paper we have shown that model-based EAs like CMA-ES can be effective in the solution of a BCI classification problem based on motor imagery. Our approach has a number of merits: 1) The final classifier can be better adjusted to the characteristics of the data (no assumptions about the normality of the data are required); 2) The optimization algorithm can extract and exploit regularities from the search space allowing the adaptation to different dataset distributions; 3) The model built during the optimization process can reveal information about the problem domain. We have also evaluated the feasibility of OpenViBE as a tool for fast addition and validation of alternative machine learning algorithms for BCI development.

The results presented in this paper are preliminary. More experiments should be conducted to assess the statistical significance of the results and validation of the algorithm on other subjects is also required. Many extensions are possible to increase the accuracy of the classifiers. One possibility is to evolve sets of classifiers instead of single classifiers as done in [25]. When stochastic optimization methods are used, usually diverse optimal solutions can be obtained and combined to create more robust classifiers. The optimization approach also allows the fitness function to be modified. This can be done by assigning different weights to the different cases that are being classified allowing to deal with class imbalance and domain adaptation problems. In addition to CMA-ES other model-based EAs like estimation of distribution algorithms that learn Bayesian [14], Gaussian [13], and Markov networks [27] could be tried. Finally, more work is required to investigate the potential implications and use of the patterns of interactions captured by the models learned by the optimization algorithms.

7. ACKNOWLEDGMENTS

This work has been partially supported by the Saiotek and Research Groups 2007-2012 (IT-242-07) programs (Basque Government), TIN2010-14931 and Consolider Ingenio 2010 - CSD 2007 - 00018 projects (Spanish Ministry of Science and Innovation) and COMBIOMED network in computational biomedicine (Carlos III Health Institute). French National Research Agency through ANR OpenViBE2 project and Brittany Region through BRAINVOX project.

8. REFERENCES

[1] A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *Proceedings of The IEEE Congress on Evolutionary Computation (CEC-2005)*, volume 2, pages 1769–1776. IEEE, 2005.

[2] S. Baskar, P. Suganthan, N. Ngo, A. Alphones, and R. Zheng. Design of triangular FBG filter for sensor applications using covariance matrix adapted

evolution algorithm. *Optics communications*, 260(2):716–722, 2006.

[3] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, and K. Muller. Optimizing spatial filters for robust EEG single-trial analysis. *Signal Processing Magazine, IEEE*, 25(1):41–56, 2008.

[4] L. Citi, R. Poli, C. Cinel, and F. Sepulveda. Feature selection and classification in brain computer interfaces by a genetic algorithm. In *Late-breaking papers of the Genetic and Evolutionary Computation Conference (GECCO-2004)*, volume 400, 2004.

[5] M. Fatourechi, A. Bashashati, R. Ward, and G. Birch. A hybrid genetic algorithm approach for improving the performance of the lf-asd brain computer interface. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on*, volume 5, pages v–345. IEEE, 2005.

[6] D. Garrett, D. Peterson, C. Anderson, and M. Thaut. Comparison of linear, nonlinear, and feature selection methods for EEG signal classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 11(2):141–144, 2003.

[7] 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*, pages 312–317, 1996.

[8] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

[9] B. Hasan, J. Gan, and Q. Zhang. Multi-objective evolutionary methods for channel selection in brain-computer interfaces: some preliminary experimental results. In *IEEE Congress on Evolutionary Computation (CEC-2010)*, pages 1–6. IEEE, 2010.

[10] E. Haselsteiner and G. Pfurtscheller. Using time-dependent neural networks for EEG classification. *IEEE Transactions on Rehabilitation Engineering*, 8(4):457–463, 2000.

[11] U. Hoffmann, J. Vesin, T. Ebrahimi, and K. Diserens. An efficient p300-based brain-computer interface for disabled subjects. *Journal of Neuroscience Methods*, 167(1):115–125, 2008.

[12] D. Krusienski, E. Sellers, F. Cabestaing, S. Bayoudh, D. McFarland, T. Vaughan, and J. Wolpaw. A comparison of classification techniques for the P300 speller. *Journal of neural engineering*, 3:299, 2006.

[13] P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. Peña. Combinatorial optimization by learning and simulation of Bayesian networks. In *Proceedings of the Sixteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-2000)*, pages 343–352, San Francisco, CA, 2000. Morgan Kaufmann Publishers.

[14] P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Boston/Dordrecht/London, 2002.

[15] M. Lebedev and M. Nicolelis. Brain-machine interfaces: Past, present and future. *TRENDS in Neurosciences*, 29(9):536–546, 2006.

[16] F. Lotte, M. Congedo, A. Lecuyer, F. Lamarche, and B. Arnaldi. A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, 4:R1–R13, 2007.

[17] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN IV*, volume 1141 of *Lectures Notes in Computer Science*, pages 178–187, Berlin, 1996. Springer.

[18] A. Nijholt. BCI for games: A state of the art survey. pages 225–228. Springer, 2009.

[19] M. Pelikan, D. E. Goldberg, and F. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20, 2002.

[20] R. Poli, L. Citi, F. Sepulveda, and C. Cinel. Analogue evolutionary brain computer interfaces [application notes]. *Computational Intelligence Magazine, IEEE*, 4(4):27–31, 2009.

[21] A. Rakotomamonjy and V. Guigue. BCI competition III: Dataset II-ensemble of SVMs for BCI P300 speller. *Biomedical Engineering, IEEE Transactions on*, 55(3):1147–1154, 2008.

[22] Y. Renard, F. Lotte, G. Gibert, M. Congedo, E. Maby, V. Delannoy, O. Bertrand, and A. Lécuyer. OpenViBE: An open-source software platform to design, test, and use brain-computer interfaces in real and virtual environments. *Presence: teleoperators and virtual environments*, 19(1):35–53, 2010.

[23] L. Rossini, D. Izzo, and L. Summerer. Brain-machine interfaces for space applications. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages 520–523, 2009.

[24] R. Santana, C. Bielza, and P. Larrañaga. Regularized logistic regression and multi-objective variable selection for classifying MEG data. 2012. Submmitted for publication.

[25] R. Santana, S. Muelas, A. Latorre, and J. M. Peña. A direct optimization approach to the P300 speller. In *Proceedings of the 2011 Genetic and Evolutionary Computation Conference GECCO-2011*, pages 1747–1754, Dublin, Ireland, 2011.

[26] A. Schlögl, C. Brunner, R. Scherer, and A. Glatz. *Towards Brain-Computer Interfacing*, chapter BioSig - an open source software library for BCI research, pages 347–358. MIT press, 2007.

[27] S. Shakya and R. Santana, editors. *Markov Networks in Evolutionary Computation*. Springer, 2012. In Press.

[28] J. Wolpaw, N. Birbaumer, D. McFarland, G. Pfurtscheller, and T. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6):767–791, 2002.