ARTICLE IN PRESS

Neurocomputing ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Adaptive semi-supervised classification to reduce intersession non-stationarity in multiclass motor imagery-based brain–computer interfaces

Luis F. Nicolas-Alonso*, Rebeca Corralejo, Javier Gomez-Pilar, Daniel Álvarez, Roberto Hornero

Biomedical Engineering Group, E.T.S. Ingenieros de Telecomunicación, Universidad de Valladolid, Valladolid, Spain

ARTICLE INFO

Article history:
Received 3 September 2014
Received in revised form
10 January 2015
Accepted 3 February 2015
Communicated by Wei Wu

Keywords:
Adaptive classification
Brain–computer interfaces
Electroencephalography
Non-stationarity
Semi-supervised classification

ABSTRACT

The intersession non-stationarity in electroencephalogram (EEG) data is a major issue to robust operation of brain–computer interfaces (BCIs). The aim of this paper is to propose a semi-supervised classification algorithm whereby the model is gradually enhanced with unlabeled data collected online. Additionally, a processing stage is introduced before classification to adaptively reduce the small fluctuations between the features from training and evaluation sessions. The key element of the classification algorithm is an optimized version of kernel discriminant analysis called spectral regression kernel discriminant analysis (SRKDA) in order to meet the low computational cost requirement for online BCI applications. Four different approaches, SRKDA and sequential updating semi-supervised SRKDA (SUSS-SRKDA) with or without adaptive processing stage are considered to quantify the advantages of semi-supervised learning and adaptive stage. The session-to-session performance for each of them is evaluated on the multiclass problem (four motor imagery tasks: the imagination of movement of the left hand, right hand, both feet, and tongue) posed in the BCI Competition IV dataset 2a. The results agree with previous studies reporting semi-supervised learning enhances the adaptability of BCIs to non-stationary EEG data. Moreover, we show that reducing the inter-session non-stationarity before classification further boosts its performance. The classification method combining adaptive processing and semi-supervised learning is found to yield the highest session-to session transfer results presented so far for this multiclass dataset: accuracy (77%) and Cohen's kappa coefficient (0.70). Thus, the proposed methodology could be of great interest for real-life BCIs.

1. Introduction

The aim of electroencephalogram (EEG)-based brain–computer interface (BCI) research is to design systems that enable humans to interact with their surroundings, without the involvement of peripheral nerves and muscles, by using control signals generated from EEG activity [1]. BCIs create an alternative non-muscular communication channel that directly translates brain activity into sequences of control commands for external devices such as computers, speech synthesizers, assistive appliances, and neural prostheses amongst many others.

EEG-based BCIs utilize some control signals such as visual evoked potentials, P300 evoked potentials, slow cortical rhythms, or sensorimotor rhythms to know the users' intentions [1]. The detection and classification of sensorimotor rhythms has attracted growing attention in BCI research [2–4]. These control signals make it possible to design endogenous BCIs, as well as convey continuous commands. However, the applicability of motor imagery based-BCIs (MI-BCIs) in real environments is still limited by low transfer rates [5]. MI-BCIs are usually built to discriminate just two brain states: left and right hand movements. Hence, extending the number of recognizable motor brain states has been researched to speed up the communication. Ehrsson et al. [6] showed that motor activity from different body parts such as fingers, toes, and tongue are mapped into different locations in the primary motor cortex and, therefore, the spatial patterns generated can be discriminated. Boosting transfer rates by means of multiclass paradigms is an ongoing research that challenges the discrimination capability of current BCI systems. Attempts to increase the number of tasks naturally suffer from higher

* Correspondence to: E.T.S. Ingenieros de Telecomunicación, Universidad de Valladolid, Paseo de Belén, 15, 47011 Valladolid, Spain. Tel.: +34 983 423000x3708; fax: +34 983 423667.

E-mail addresses: lnicalo@ribera.tel.uva.es (L.F. Nicolas-Alonso), rebeca.corralejo@gib.tel.uva.es (R. Corralejo), javier.gomez@gib.tel.uva.es (J. Gomez-Pilar), dalvgon@ribera.tel.uva.es (D. Álvarez), robhor@tel.uva.es (R. Hornero).

http://dx.doi.org/10.1016/j.neucom.2015.02.005
0925-2312/

Please cite this article as: L.F. Nicolas-Alonso, et al., Adaptive semi-supervised classification to reduce intersession non-stationarity in multiclass motor imagery-based brain–computer interfaces, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.02.005

misclassification rate because there is a trade-off between the number of motor tasks and the accuracy [5]. In this context, to enhance the performance of multiclass MI-BCIs, several studies have been conducted either in the feature extraction [7–11] or classification [12,13] stages, to cite only a few examples. This paper focuses on improving the performance of multiclass MI-BCIs introducing a novel adaptive classification methodology. To the best of our knowledge, no adaptive classifier has been evaluated so far in multiclass settings.

BCI systems can be seen as pattern recognition systems that identify the user's intentions on the basis of a set of features that characterizes the brain activity. They are usually calibrated by supervised learning during training sessions. The user is asked to perform different tasks and brain signals together with their class labels are stored. The recorded data serve as the training dataset for the BCI. After the calibration session, the machine is assumed to be able to detect the patterns of brain signals recorded in the subsequent online sessions. However, EEG signals are naturally non-stationary and usually very noisy. Diverse behavioral and mental states continuously change the statistical properties of brain signals [14]. Furthermore, they are contaminated with non-stationary artifacts such as electromyogram (EMG) and electro-oculogram (EOG) [15]. Patterns observed during calibration sessions may be different from those recorded during online sessions. Hence, non-stationarity can result in degraded performance since the supervised machine learning algorithms implicitly assume stationary data.

Semi-supervised learning is becoming more important in an attempt to improve the adaptability and robustness of BCI systems, as well as reduce the labeled data needed to calibrate. Such approach uses both labeled and unlabeled data to train the classification model. The more often-used semi-supervised methods include: expectation-maximization, transductive support vector machines (TSVM), self-training, and co-training, amongst others (see Zhu [16] for a review). In BCI research, the expectation-maximization algorithm was used for the unsupervised readjustment of a Gaussian mixture model (GMM) [17] or a classifier based on linear discriminant analysis (LDA) [18], as well as adaptive extraction and classification of common spatial pattern (CSP) features [19]. TSVM was proposed to classify three different mental tasks [20]. Also, self-training algorithms based on support vector machines (SVM) were successfully applied in motor imagery [21] and P300-based BCIs [22].

Although semi-supervised approaches are capable of increasing the performance of BCIs even with small labeled training datasets and non-stationarity signals [23], it might not be the best approach in an adaptive framework since feature vectors from different sessions are used to build a single model. In semi-supervised approaches, the classification model is incrementally updated with the augmented dataset, which includes the initial labeled dataset and the new incoming points with the predicted labels. Despite the model being updated, the same probability density function for training and evaluation sessions is implicitly assumed. It goes against the finding that was explained above; feature vectors extracted from EEG data of different sessions follow different probability distributions. Some studies have already applied adaptive algorithms with no semi-supervised learning involvement to cope with this issue. Adaptive procedures such as bias adaptation [14,24], importance weighted cross validation [25,26], or data space adaptation based on the Kullback–Leibler divergence [27] were proposed to extend LDA to non-stationary environments. Likewise, dynamic Bayesian classifiers based on the Kalman filter [28–30] have been developed for on-line adaptive classification. All these methods sequentially update the model during the unlabeled dataset or testing sessions giving more importance to the most recent trials. This motivated us to study whether this kind of procedures that deal with the mismatching between training and evaluation data can improve the adaptability of semi-supervised-based BCI systems. We introduce an adaptive processing stage before classification that estimates and corrects the mismatching running exponentially weighted moving average (EWMA).

As may be seen from BCI literature, successful semi-supervised approaches make use of kernel methods such as SVMs [21,22]. However, they are not tailored for online semi-supervised learning applications, which require retraining the model with all the new and past data. Training conventional SVMs involves solving a quadratic programming problem, which is computationally intensive. The computational load can become unacceptably high due to the increasing amount of training data. Hence, finding new solutions is of great importance to design practical online semi-supervised BCIs. In this regard, Gu et al. [23] proposed a semi-supervised algorithm based on the least squares SVM (LS-SVM) instead of the common SVM to design a semi-supervised P300 BCI. The least square version of SVM replaces the quadratic programming problem by a set of linear equations to meet the requirement of low computational complexity. However, the algorithm relies on computing and sequentially updating the inverse of a matrix that can be inaccurate and unstable [31]. It motivated us to design an online semi-supervised classifier based on other equally efficient kernel-based classification method called spectral regression kernel discriminant analysis (SRKDA) [32]. SRKDA reformulates the popular kernel discriminant analysis (KDA) [33] or generalized discriminant analysis [34] to avoid the eigen-decomposition of the kernel matrix, which is very expensive when a large number of training samples exists. SRKDA only needs to solve a linear system of equations that can be efficiently performed using Cholesky decomposition. Furthermore, regarding online semi-supervised BCI applications, the algorithm can be incrementally formulated as new samples arrive saving a huge amount of computational cost.

The aim of this paper is to propose a classification algorithm whereby the model is gradually enhanced with the unlabeled data collected online. The novelty of the approach lies in two components. Firstly, we present a new sequential updating semi-supervised classification method based on SRKDA. After training the classifier with labeled data, a self-training algorithm is used to sequentially update the model using the arriving unlabeled data. The resultant algorithm is called sequential updating semi-supervised SRKDA (SUSS-SRKDA). Secondly, adaptive processing with EWMA is introduced before classification in order to reduce the existing non-stationarity between training and evaluation sessions. Then, four methods, SRKDA and SUSS-SRKDA with or without adaptive stage, are evaluated on the BCI Competition IV dataset 2a [35] to quantify the advantages of self-training-based classification and the adaptive stage. All these alternatives are compared to the winner of the competition and other methods tested on this dataset [8,36–41]. Likewise, we evaluate two adaptive classifiers in the BCI literature [18,24] on the features extracted in this study in order to further emphasize the benefits of our contribution. We refer the method proposed by Blumberg et al. [18] as expectation-maximization-based LDA (EM-LDA). Vidaurre et al. [24] introduced three types of adaptive procedures into the well-known LDA. Here, we just focus on the method providing the highest performance, which was called Pooled Mean (PMean) by the authors. Although the two methods were proposed to classify two motor imagery tasks, they can be straightforwardly extended to multiclass problems.

2. BCI Competition IV dataset 2a description

The algorithms proposed are evaluated on the BCI Competition IV dataset 2a provided by Graz University [35]. This dataset contains EEG

Please cite this article as: L.F. Nicolas-Alonso, et al., Adaptive semi-supervised classification to reduce intersession non-stationarity in multiclass motor imagery-based brain–computer interfaces, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.02.005

signals from 9 healthy subjects performing four different motor imagery tasks: movement of the left hand, right hand, feet, and tongue. Two sessions, one for training and the other for evaluation, were recorded on different days for each subject. Each session includes 288 trials of data (72 for each of the four possible tasks) recorded with 22 EEG channels and 3 monopolar EOG channels (with left mastoid serving as reference). The signals were sampled at 250 Hz and band-pass filtered between 0.5 Hz and 100 Hz. A 50 Hz notch filter was also applied to suppress power line noise.

All volunteers were sitting in an armchair, watching a flat screen monitor. At the beginning of each trial a cross was shown on the black screen and a short warning tone was given. At second 2, a cue in the form of an arrow pointing to the left, right, down, or up (corresponding to one of the four classes left hand, right hand, foot or tongue) was presented during 1.25 s. Depending on the direction of the arrow, the subjects were prompted to perform the corresponding motor imagery task until the cross disappeared from the screen at second 6. Refer to Tangermann et al. [35] for further details on the BCI Competition IV dataset 2a.

3. Proposed methods

The architecture of the proposed algorithm is illustrated in Fig. 1. It comprises five consecutive stages: multiple bandpass filtering using finite impulse response (FIR) filters, spatial filtering using the CSP algorithm, feature selection, adaptive processing, and classification of the selected CSP features. Configurable parameters are adjusted for each subject using the trials labeled with the respective motor imagery tasks from the training session. These parameters are then used to compute the motor imagery task for each trial over the evaluation session. Fig. 2 illustrates how a single-trial is processed as well as the time scheme of the paradigm. The algorithm computes feature vectors at any point in time using a sliding 2 second window of EEG data. The classification output is continuously computed with the feature vector of the corresponding window satisfying the causality criterion required by the competition [35]. Note that no classification output is generated during the first 2 s. Each stage of the algorithm is explained in more detail in following sections.

3.1. Band-pass filtering

The first stage employs a filter bank that decomposes the EEG into 9 frequency pass bands, namely, 4–8 Hz, 8–12 Hz,…, 36–40 Hz [10]. Nine FIR filters designed by means of Kaiser Window are used. FIR filters are particularly suitable for the design of filter banks because they have linear phase, which does not distort the phase of the filtered signal. The transition bandwidth is set at 1 Hz. Other configurations are as effective, but this transition bandwidth yields a reasonable order filter and discriminative capacity between frequency bands.

3.2. Spatial filtering

The second stage of feature extraction performs spatial filtering using CSP algorithm for each band-pass signal. CSP is a successful algorithm for the design of motor imagery-based BCIs [42]. It has been devised for the analysis of multichannel data belonging to 2-class problems. Consequently, although other options are feasible in multiclass problems, we adopt the one-versus-rest approach [7]. CSP filters are computed on the basis of the trials for each class versus the trials for all other classes.

CSP calculates the spatial filters by solving an eigenvalue decomposition problem that involves the mean spatial covariances for each of the two classes [42]. The spatial filtered signal Z is obtained from the EEG trial E as

$$Z = W^T E, \tag{1}$$

where W is a matrix containing the spatial filters computed by CSP. Each column of W represents a spatial filter. There are as many spatial filters as EEG channels. For each frequency band, CSP feature vectors are given by

$$\boldsymbol{x} = \log\left[\frac{\mathrm{diag}(\tilde{W}^T E E^T \tilde{W})}{\mathrm{trace}(\tilde{W}^T E E^T \tilde{W})}\right], \tag{2}$$

where $\tilde{W}$ represents a matrix having some spatial filters of W. Since all spatial filters of W are not relevant for subsequent classification, the first 2 and the last 2 columns of W are selected [10]. In accordance to the one-versus-rest approach, 16 features are obtained as a result of repeating the CSP algorithm for each class. Finally, the 16 features of the 9 frequency bands for a single-trial are concatenated to form a single feature vector of 144 features.

3.3. Feature selection

After spatial filtering, mutual information-based best individual feature (MIBIF) algorithm [10] is employed to select the most discriminative features. MIBIF involves the computation of the mutual information between each feature and class labels. Then, the features with higher mutual information are selected. In this work, the number of selected features is configured by 10-fold cross validation on the training session.

3.4. Classification

Firstly, the adaptive processing stage is presented. This stage performs an unsupervised adaptation whereby the extracted features are processed before classification in order to reduce the mismatching between training and evaluation data. Secondly, the

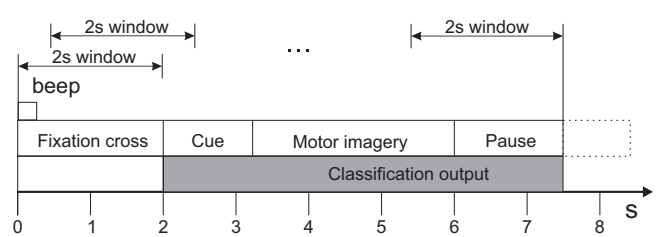


Fig. 2. Feature extraction from a single-trial EEG. The spatial filters and the most discriminant bands computed in the training phase are used to extract the features in the evaluation phase.

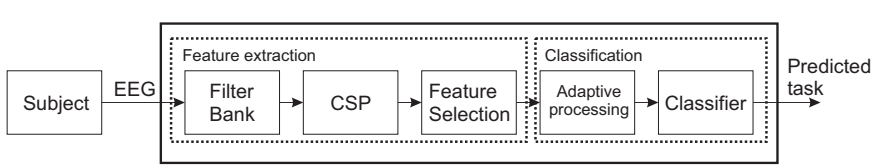


Fig. 1. Architecture of the algorithm. The parameters of CSP, feature selection, and classification stages are adjusted for each subject using training data labeled with the respective motor imagery tasks. These parameters computed from the training phase are then used to compute the single-trial motor imagery task during the evaluation phase.

Please cite this article as: L.F. Nicolas-Alonso, et al., Adaptive semi-supervised classification to reduce intersession non-stationarity in multiclass motor imagery-based brain–computer interfaces, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.02.005

classification algorithms, KDA and its efficient version, SRKDA, are briefly introduced. Finally, semi-supervised SRKDA (SS-SRKDA) and its sequential updating semi-supervised version (SUSS-SRKDA) are proposed. Semi-supervised algorithms use the unlabeled samples to augment the training dataset. In this work, we use a variant of semi-supervised learning known as self-training [16]. The classifier is firstly trained with the labeled data and then the unlabeled data are classified. After that, the unlabeled data together with their predicted labels are added to the training dataset and the classifier is retrained. These steps are repeated until the algorithm converges.

3.4.1. Adaptive processing

The adaptive processing stage centers every incoming data by subtracting the global mean. Firstly, the global mean is estimated from the whole training data. Across the evaluation session, upon the arrival of a new sample at the time t from the i-th evaluation trial, the global mean $\boldsymbol{\mu}_G(i,t)$ is updated by means of EWMA. EWMA is a powerful tool for mean estimation in noisy environments [43]. The sequential estimations are given by

$$\boldsymbol{\mu}_G(i,t) = (1-\eta)\cdot\boldsymbol{\mu}_G(i-1,t)+\eta\cdot\boldsymbol{x}(i,t), \tag{3}$$

where $\boldsymbol{x}(i,t)$ is the current input feature vector of the i-th evaluation trial at the time t and η is the update coefficient, which has to be configured by the user. The exponential rule estimates the global mean by an amount that is proportional to the most recent forecast error. Simple algebraic manipulation reveals that $\boldsymbol{\mu}_G(i,t)$ can be written as a weighted average of all past observations, in which weights for older samples decay exponentially. This is consistent with the idea that adaptive procedures should give more importance to the most recent terms in the time series and less importance to older data [24].

3.4.2. Kernel discriminant analysis

KDA [33] was proposed to extend LDA to the non-linear case. KDA is a classifier that, in a similar way to LDA, seeks directions that improve class separation. However, KDA considers the problem in the feature space $\mathfrak{F}$ induced by some non-linear mapping $\phi:\Re^{N_F}\to\mathfrak{F}$, where N_F is the number of features. The KDA basis is to map the feature vectors into a high dimensional space where complex classification problems are more likely to be linearly separable [44]. The objective function of KDA to find the optimal projective functions $\boldsymbol{v}_{opt}$ is as follows:

$$\boldsymbol{v}_{opt} = \operatorname{argmax}\frac{\boldsymbol{v}^T S_B^{\phi}\boldsymbol{v}}{\boldsymbol{v}^T S_W^{\phi}\boldsymbol{v}}, \tag{4}$$

where $\mathbf{v}\in\mathfrak{F}.S_B^{\phi}$ and S_W^{ϕ} are the between-class and within-class scatter matrices in $\mathfrak{F}$, i.e.

$$S_B^{\phi} = \sum_{k=1}^{C} M_k(\boldsymbol{\mu}_{\phi}^{(k)}-\boldsymbol{\mu}_{\phi})(\boldsymbol{\mu}_{\phi}^{(k)}-\boldsymbol{\mu}_{\phi})^T \tag{5}$$

and

$$S_w^{\phi} = \sum_{k=1}^{C}\left(\sum_{i=1}^{M_k}\left(\phi(\boldsymbol{x}_i^{(k)})-\boldsymbol{\mu}_{\phi}^{(k)}\right)\left(\phi(\boldsymbol{x}_i^{(k)})-\boldsymbol{\mu}_{\phi}^{(k)}\right)^T\right). \tag{6}$$

C is the number of classes, $\boldsymbol{\mu}_{\phi}^{(k)}$ and $\boldsymbol{\mu}_{\phi}$ are the centroids of the k-th class and the global mean, respectively, in the feature space, and M_k is the number of feature vectors in the k-th class.

It can be proved that the above maximization problem can be solved efficiently using the *kernel trick* [33]. For a chosen mapping function ϕ, an inner product $\langle,\rangle$ can be defined on $\mathfrak{F}$, which makes for the so-called reproducing kernel Hilbert space (RKHS) $\langle\phi(\mathbf{x}),\phi(\mathbf{x})\rangle = K(\mathbf{x},\mathbf{y})$, where $K(\mathbf{x},\mathbf{y})$ is a positive semi-definite kernel function. Then, from the theory of reproducing kernels, we know that any solution $\boldsymbol{v}_{opt}\in\mathfrak{F}$ must lie in the span of all training samples in $\mathfrak{F}$. There exist coefficients α_i such that

$$\boldsymbol{v}_{opt} = \sum_{i=1}^{M}\alpha_i\phi(\boldsymbol{x}_i), \tag{7}$$

where M is the number of total training data points.

Let $\boldsymbol{\alpha}_{opt} = [\alpha_1,\alpha_2,\ldots,\alpha_M]$, then it can be proved the Eq. (4) is equivalent to

$$\boldsymbol{\alpha}_{opt} = \operatorname*{argmax}_{\boldsymbol{\alpha}}\frac{\boldsymbol{\alpha}^T KVK\boldsymbol{\alpha}}{\boldsymbol{\alpha}^T KK\boldsymbol{\alpha}}, \tag{8}$$

where, K is the kernel matrix $K_{ij} = K(\boldsymbol{x}_i,\boldsymbol{x}_j)$, and V is defined as

$$V = \begin{cases} 1/M_k, & \text{if } \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ both belong to the } k-\text{th class} \\ 0, & \text{otherwise} \end{cases}. \tag{9}$$

The above maximization problem corresponds to the following eigenvalue decomposition problem

$$KVK\boldsymbol{\alpha} = \lambda KK\boldsymbol{\alpha}. \tag{10}$$

Each eigenvector $\boldsymbol{\alpha}_{opt}$ gives the projection of a new test pattern $\hat{\boldsymbol{x}}$ onto $\boldsymbol{v}$ in the feature space. For a new data example $\hat{\boldsymbol{x}}$, we have

$$\Theta(\hat{\boldsymbol{x}},\boldsymbol{\alpha}_{opt}) = \langle v,\phi(\hat{\boldsymbol{x}})\rangle = \sum_{i=0}^{M}\alpha_i K(\boldsymbol{x}_i,\hat{\boldsymbol{x}}) \tag{11}$$

Finally, $\hat{\boldsymbol{x}}$ is classified on the basis of the Euclidean distance to the projected mean for each class

$$\hat{l} = \operatorname*{argmin}_k \|\langle v,\phi(\hat{\boldsymbol{x}})\rangle - \left\langle v,\boldsymbol{\mu}_{\phi}^{(k)}\right\rangle\|. \tag{12}$$

3.4.3. Spectral regression kernel discriminant analysis

SRKDA [32] is an improvement of KDA that finds optimal projection casting KDA into a regression framework. The great advantage of this approach is to facilitate efficient computation since there is no eigenvector computation involved to solve Eq. (8). In order to derive SRKDA, Cai et al. [32] proved the following theorem:

Let $\boldsymbol{y}$ be the eigenvector of eigen-problem

$$V\boldsymbol{y} = \lambda\boldsymbol{y}, \tag{13}$$

with eigenvalue λ. If $K\boldsymbol{\alpha} = \boldsymbol{y}$, then $\boldsymbol{\alpha}$ is an eigenvector of the eigen-problem in Eq. (10) with the same eigenvalue λ.

According to the above theorem, projective functions can be obtained through two steps: (1) solving the eigen-problem in Eq. (13) to get $\boldsymbol{y}$, and (2) finding $\boldsymbol{\alpha}_{opt}$ which satisfies $K\boldsymbol{\alpha} = \boldsymbol{y}$. The solution of the eigen-problem in the first step can be trivially found exploiting the special structure of V. Without loss of generality, we can assume that the training data points are ordered according to their labels. Then, V has a block-diagonal structure

$$V = \begin{pmatrix} V^{(1)} & 0 & \cdots & 0 \\ 0 & V^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & V^{(c)} \end{pmatrix}, \tag{14}$$

where $\{V^{(k)}\}_{k=1}^{C}$ is an $M_k\times M_k$ matrix with all the elements equal to $1/M_k$. It is straightforward to show that $V^{(k)}$ has only one eigenvector $\boldsymbol{e}^{(k)} = [1,1,\ldots,1]^T\in\Re^{M_k}$, which is associated with eigenvalue 1. Due to the block-diagonal structure of V, the eigenvalues and eigenvectors are the union of the eigenvalues and eigenvectors of its blocks (the latter padded appropriately with zeros). Therefore, there are C eigenvectors of V with the same eigenvalue 1. These eigenvectors are

$$\bar{\boldsymbol{y}}_k = [\underbrace{0,\ldots,0}_{\sum_{i=1}^{k-1}M_i},\underbrace{1,\ldots,1}_{M_k},\underbrace{0,\ldots,0}_{\sum_{i=k+1}^{C}M_i}]^T \quad k=1,\ldots,C \tag{15}$$

Since all eigenvalues of V are 1, we can just pick any other C orthogonal vectors in the space spanned by $\{\bar{\boldsymbol{y}}_k\}_{k=1}^{C}$, and define them to be our C eigenvectors. The vector of all ones is naturally in the spanned space. This vector is useless since the corresponding projective function will embed all the samples to the same point. Therefore, this vector is picked as the first eigenvector of V and the remaining eigenvectors are found by means of the Gram–Schmidt algorithm. The vector of all ones can then be removed leaving exactly $C-1$ eigenvectors of V, $\{\boldsymbol{y}_k\}_{k=1}^{C-1}$.

In the second step, an $\boldsymbol{\alpha}_{opt}$ is obtained for each eigenvector of V by solving the corresponding linear equation system $K\boldsymbol{\alpha}=\boldsymbol{y}$. The kernel matrix K is positive semi-definite. When K is singular, the system may have no solution or have infinite solutions. Then, a possible way is to adopt the regularization technique to obtain an approximate estimator: $(K+\delta I)\boldsymbol{\alpha}_{opt,k}=\boldsymbol{y}_k$, where I is the identity matrix and $\delta \geq 0$ is the regularization parameter. Once the matrix $K+\delta I$ is positive definite, the Cholesky decomposition can be used to efficiently compute the solution. Finally, after the two steps, the new patterns are classified projecting the feature vector with the $C-1$ projective functions $\{\boldsymbol{\alpha}_{opt,k}\}_{k=1}^{C-1}$ in the same way as KDA. For the purposes of clarity, the notation of the $C-1$ projective functions $\{\boldsymbol{\alpha}_{opt,k}\}_{k=1}^{C-1}$ and $C-1$ eigenvectors of V are hereinafter denoted as $\boldsymbol{\alpha}$ and $\boldsymbol{y}$.

3.4.4. Semi-supervised spectral regression kernel discriminant analysis

SS-SRKDA algorithm builds the classification model with an initial training dataset $D=\{(\boldsymbol{x}_i, l_i)\}_{i=1}^{M}$ and updates it with an unlabeled feature vector $\hat{\boldsymbol{x}}$ as described below.

- Step (1) Training SRKDA classifier using the training dataset D to obtain the initial eigenvectors $\boldsymbol{y}$ of V_M, Cholesky decomposition $R_M = cholesky\,(K_M+\delta I_M)$, and $\boldsymbol{\alpha}^{(0)} \in \Re^M$, where I_M is the $M \times M$ size identity matrix. The subscript M denotes the number of training samples and the notation $(.)^{(j)}$ denotes the j-th iteration.
- Step (2) Finding the predicted label $\hat{l}^{(0)}$ for $\hat{\boldsymbol{x}}$ by using SRKDA with $\boldsymbol{\alpha}^{(0)}$. Meanwhile, the augmented Cholesky decomposition $R_{M+1} = cholesky(K_{M+1}+\delta I_{M+1})$ can be obtained from the augmented dataset $D \cup \{\hat{\boldsymbol{x}}\}$ since it does not depend on labels. Note that R_{M+1} can be efficiently computed in the incremental manner. When the Cholesky decomposition R_M of the $M \times M$ submatrix $K_M+\delta I_M$ is known, the Cholesky decomposition R_{M+1} of the $(M+1)\times(M+1)$ submatrix $K_{M+1}+\delta I_{M+1}$ can be easily computed by *Sherman's march* algorithm [32,45].
- Step (3) For the j-th iteration, $j \geq 1$, training SRKDA with the augmented training dataset $\hat{D}^{(j)} = D \cup \{\hat{\boldsymbol{x}}, \hat{l}^{(j-1)}\}$ to obtain the eigenvectors $\hat{\boldsymbol{y}}^{(j-1)}$ of $\hat{V}^{(j-1)}$. Finally, $\boldsymbol{\alpha}^{(j)} \in \Re^{M+1}$ is found using $R_{M+1} \in \Re^{(M+1)\times(M+1)}$ and $\hat{\boldsymbol{y}}^{(j-1)}$.
- Step (4) With $\boldsymbol{\alpha}^{(j)}$, finding the predicted label $\hat{l}^{(j)}$ for $\hat{\boldsymbol{x}}$.
- Step (5) If $\hat{l}^{(j)} = \hat{l}^{(j-1)}$, then the algorithm converged and terminate. Otherwise, repeat the Steps 3–5.

The convergence of the SS-SRKDA algorithm is proved in the Appendix A.

3.4.5. Sequential updating semi-supervised SRKDA

With an initial training dataset $D=\{(\boldsymbol{x}_i, l_i)\}_{i=1}^{M}$, and the sequentially arriving unlabeled feature vector from i-th trial, $\{\hat{\boldsymbol{x}}_i\}_{i=1}^{N}$, SUSS-SRKDA successively updates the classification model as described below.

- Step (1) After receiving the unlabeled feature vector $\hat{\boldsymbol{x}}_1$ from the first testing trial, the predicted label $\hat{l}_1$ is found by using SS-SRKDA trained with the dataset D. Then, the training dataset is augmented, $\hat{D}_1 = D \cup \{\hat{\boldsymbol{x}}_1, \hat{l}_1\}$.
- Step (2) For each unlabeled feature vector $\hat{\boldsymbol{x}}_i$, the predicted label $\hat{l}_i$ is found by using SS-SRKDA trained with the dataset $\hat{D}_{i-1}$. Then, the training dataset $\hat{D}_{i-1}$ is augmented $\hat{D}_i = \hat{D}_{i-1} \cup \{\hat{\boldsymbol{x}}_i, \hat{l}_i\}$. This step is repeated across the whole test dataset $i=2,\ldots,N$.

3.4.6. Computational cost analysis

In this section, the computational complexities of the proposed SS-SRKDA and SUSS-SRKDA are derived in detail. For the sake of completeness, computational complexity of SRKDA is also included, although it has been already described by Cai et al. [32]. We use the same term *flam* [32], a compound operation consisting of one addition and one multiplication, to measure the operation counts.

SRKDA training involves two steps: generating the responses $\{\boldsymbol{y}_k\}_{k=1}^{C-1}$ and solving $(K+\delta I)\boldsymbol{\alpha}=\boldsymbol{y}$. Responses are computed in the first step by using Gram–Schmidt method, which requires $MC^2-1/3C^3$ flam [32]. The cost of the second step is mainly the cost of solving $C-1$ linear equations with Cholesky decomposition and the cost of computing the kernel matrix K, which require about $1/6M^3+M^2C$ and M^2N_F flam, respectively [32]. Thus, the computational cost of SRKDA is about $1/6M^3+M^2C+M^2N_F+MC^2-1/3C^3$ flam.

The cost of SS-SRKDA is derived as follows. When a new sample $\hat{\boldsymbol{x}}$ arrives, SS-SRKDA predicts the label $\hat{l}^{(0)}$. It involves two steps: the projection of the new test pattern $\hat{\boldsymbol{x}}$ onto $\boldsymbol{v}$ in the feature space (Eq. 11) and computing the Euclidean distance to the projected mean for each class (Eq. 12). The cost of the first step is the cost of computing $K(\boldsymbol{x}_i, \hat{\boldsymbol{x}})$ and the inner product between $\boldsymbol{\alpha}_{opt}$ and $K(\boldsymbol{x}_i, \hat{\boldsymbol{x}})$, which require MN_F and $M(C-1)$ flam, respectively. The cost of the second step is $(C-1)^2$ flam. Now, we consider the cost of the Cholesky decomposition of the augmented kernel matrix. We firstly need to calculate the additional part of kernel matrix. However, it was partially computed before to classify $\hat{\mathbf{x}}$. Therefore, we only need to compute $K(\hat{\boldsymbol{x}}, \hat{\boldsymbol{x}})$, which requires N_F flam. The computational cost of incremental Cholesky decomposition is $1/6(M+1)^3-1/6M^3$ flam [32]. Computing the responses $\hat{\boldsymbol{y}}^{(j-1)}$ needs $(M+1)C^2-1/3C^3$ flam, as explained in the SRKDA analysis. Besides, updating $\boldsymbol{\alpha}^{(j)}$ by solving the $C-1$ linear equations with the Cholesky decomposition, which can be done within $(M+1)^2C$ flam. Updating the predicted label $\hat{l}^{(j)}$ for $\hat{\boldsymbol{x}}$ requires $(M+1)(C-1)+(C-1)^2$ flam, as explained above. This process is repeated until the algorithm converges. To sum up, when $C \ll M$, the computation cost of incremental SS-SRKDA measured by flam is: $(1/2+C)M^2+O(MN_F)$.

Finally, the cost of SUSS-SRKDA is the cost of SS-SRKDA for classifying the i-th test trial $\hat{\mathbf{x}}_i$, namely, $(1/2+C)(M+i)^2+O(MN_F)$. It increases as new test trials are integrated into the augmented training dataset.

4. Results

In this section, the effectiveness of the proposed methods is evaluated on the publicly available BCI Competition IV dataset 2a. Four different approaches, SRKDA and SUSS-SRKDA with or without adaptive processing stage are considered. After the model selection procedure, the corresponding model is optimized on the training session and, afterwards, applied to the evaluation session. We obtain test results for each method in terms of accuracy and

kappa [46]. All the alternatives are compared with other methods in the literature.

4.1. Design and optimization on the training session

The training session is used to find the optimal configuration of the number of features selected N_F, the regularization parameter δ of SRKDA, and the update coefficient η of the adaptive processing stage. The kernel employed is the linear kernel $k(\boldsymbol{x},\boldsymbol{y})=\boldsymbol{x}^T\boldsymbol{y}$. The optimal number of features N_F and the regularization parameter δ are jointly determined running 10-fold cross validation on the training session. The optimization of such parameters is carried out independently of η. That is, the adaptive processing stage is removed from the signal processing chain (the update coefficient η is set to 0) for the cross validation. A wide range of values is defined in order to analyze their effect on generalization ability: N_F is varied from 1 to 144 units whereas δ takes values between 1 and 120. It is important to note that maximum number of features is 144 and the classification performance clearly decays when the regularization parameter exceeds 120. Fig. 3 illustrates the 10-fold cross validation classification results according to the number of features and the regularization parameter. The performance becomes higher as N_F increases. However, setting N_F higher than about 97 resulted in lower performance due to features with little relevance are included in the model. Likewise, we observe performance increases with δ. However, there is no substantial improvement beyond a given value of δ, which approximately corresponds to $\delta=85$. Therefore, the N_F and δ are fixed to 97 and 85, respectively. Finally, identical procedure is applied to configure the optimal number of features for LDA. EM-LDA and PMean, which are used as baseline, are based on LDA. We can see the optimal value is around $N_F=35$ (Fig. 4).

Having tuned the number of selected features and the regularization parameter, the update coefficient is fixed. It is an important factor affecting the performance because there is a trade-off between being highly sensitive to the changes in the global mean and being robust to noise. With a large η, the estimated mean follows the changes too truly presenting peaks, whereas, with a small η, peaks are suppressed but the variations in the real mean are followed too slowly by the estimation. The optimal value of the update coefficient depends on the natural sequence of trials. Therefore, we have to perform chronological validation on the training session instead of cross validation [47]. The training session is chronologically split into two subsets containing the 60% and 40% of the training trials. The first subset of data is used to train a classifier which is then applied to the second subset of data to evaluate the performance of the classifier. We use this procedure to imitate the online learning scenario over the evaluation session. For the proposed adaptive algorithms, different configurations with η ranging from 0 to 0.2 in steps of 0.01 are evaluated. The value yielding the highest kappa value is selected as the optimal η value. Fig. 5 illustrates the kappa values for SRKDA, SUSS-SRKDA, and PMean methods as the update coefficient varies. We can see that the classification performance becomes higher in accordance with the update coefficient. However, the improvement of kappa value slows down after about 0.10 for SRKDA and SUSS-SRKDA, and after about 0.05 for PMean. The same procedure is followed to configure the number of recent trials taken into account in the expectation maximization algorithm of EM-LDA [18]. After chronological validation, the optimal number of recent

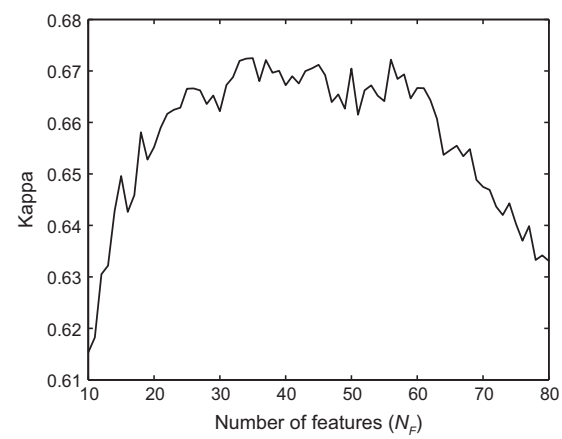


Fig. 4. Influence of the number of features N_F on the performance of LDA. PMean and EM-LDA are based on LDA.

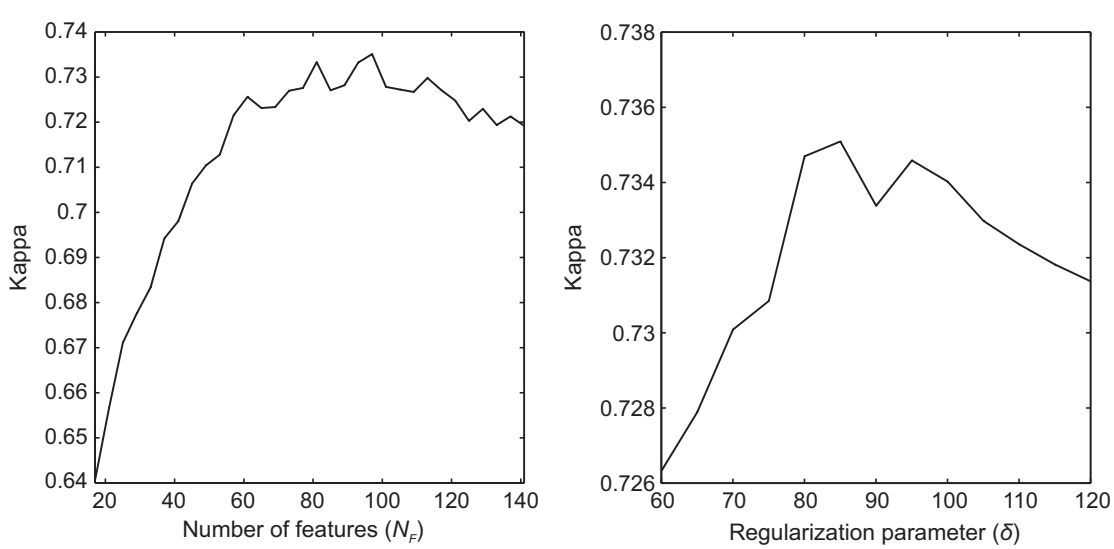


Fig. 3. Influence of the number of features N_F and regularization parameter δ on the performance of SRKDA. On the left, the plot shows the dependence between the mean kappa value and N_F when δ takes the optimum value, $\delta=85$. On the right, the plot shows the dependence between the mean kappa value and δ when N_F takes the optimum value, $N_F=97$. Although both parameters are jointly determined running 10-fold cross validation, for the sake of clarity, we show the result for each parameter in two separate plots.

Please cite this article as: L.F. Nicolas-Alonso, et al., Adaptive semi-supervised classification to reduce intersession non-stationarity in multiclass motor imagery-based brain–computer interfaces, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.02.005

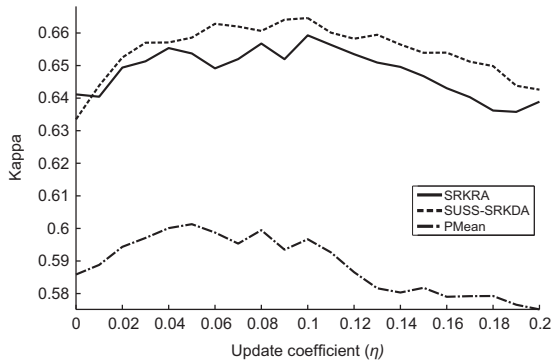


Fig. 5. Performance variation of SRKDA, SUSS-SRKDA, and PMean methods over update coefficient of the exponential rule used in the adaptive processing stage. In order to estimate the optimal update coefficient, the training session is chronologically split into two subsets containing the 60% and 40% of the training trials. The first subset of data are used to train a classifier which is then applied to the second subset of data to evaluate the performance of the classifier.

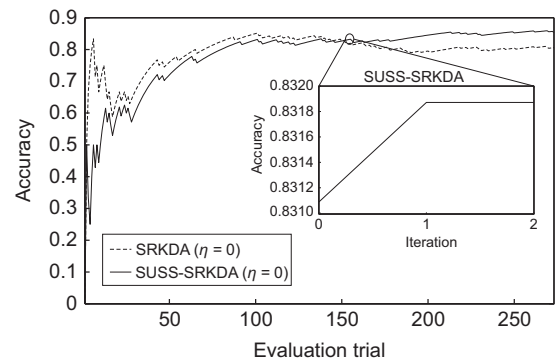


Fig. 6. Accuracy evolution of SRKDA ($\eta=0$) and SUSS-SRKDA ($\eta=0$) with different evaluation trials (Data from Subject A3). The accuracy is sequentially computed based on the actual label of the i-th evaluation trial and the actual labels of the trials that have been so far classified. In addition, figure shows an example of accuracy evolution of SUSS-SRKDA with different iterations while the algorithm is classifying the 155th evaluation trial and updating the model.

Table 1
Evaluation results obtained by applying SRKDA and SUSS-SRKDA with or without adaptive processing stage to the evaluation session from the BCI Competition IV dataset 2a.

Subjects	No adaptation ($\eta=0$)				Adaptation ($\eta>0$)			
	SRKDA		SUSS-SRKDA		SRKDA		SUSS-SRKDA	
	Acc. (%)	Kappa	Acc. (%)	Kappa	Acc. (%)	Kappa	Acc. (%)	Kappa
A1	85	0.81	86	0.81	86	0.81	**87**	**0.83**
A2	61	0.48	60	0.47	62	0.49	**64**	**0.51**
A3	81	0.74	85	0.80	90	0.87	**91**	**0.88**
A4	72	0.63	71	0.61	74	0.65	**76**	**0.68**
A5	56	0.42	54	0.38	65	0.53	**67**	**0.56**
A6	51	0.34	48	0.31	**53**	**0.38**	51	0.35
A7	89	0.86	**94**	**0.92**	90	0.87	92	0.90
A8	86	0.81	**88**	**0.84**	87	0.83	**88**	**0.84**
A9	68	0.57	72	0.62	80	0.73	**81**	**0.75**
Average	72	0.63	73	0.64	76	0.68	**77**	**0.70**

trials is 42. These values are selected as the optimum for the rest of the experiments on the unseen evaluation session.

4.2. Performance assessment on the evaluation session

Table 1 summarizes the performances of the four proposed methods on the evaluation session. According to the evaluation rules of the BCI Competition, each column in Table 1 contains the maximum accuracy and kappa achieved throughout the time course of the paradigm for each method. The highest classification performance for each subject is in boldface. The results evidence the superiority of the methods using self-training. Moreover, the mean performance is further increased by the adaptive stage. The method with adaptive processing and semi-supervised learning yields the highest mean accuracy (77%) and Cohen's kappa coefficient (0.70) in the multiclass problem. It is important to note that, although self-training alone (without the adaptive stage) achieves higher performance than SRKDA on average, the effect is just the opposite in the cases of the subjects with low performance, A2, A4, A5, A6, where A9 is the exception. Conversely, the degradation is avoided when the adaptive stage is used. SUSS-SRKDA outperforms SRKDA for all subjects with the exception of the subject A6.

Fig. 6 shows an example of accuracy evolution of SRKDA ($\eta=0$) and SUSS-SRKDA ($\eta=0$) with different evaluation trials. Although SRKDA outperforms SUSS-SRKDA at the beginning of the evaluation session, its performance is reduced at the end, which can be

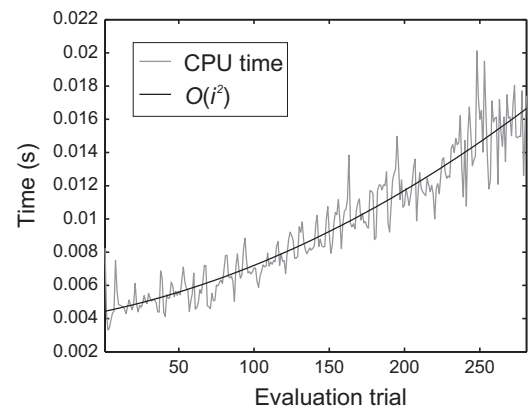


Fig. 7. CPU-time for each trial in test session. Data from Subject A1.

explained by changes in brain signal properties. On the contrary, SUSS-SRKDA reverses the trend of gradual deterioration because of updating the model with the evaluation trials. An example of the adaptation process is also depicted. We see that, upon the arrival of a new test trial, self-learning algorithm updates the model and converges in few iterations. The maximum number of iterations observed is 6. Although we show the adaptation process for one trial, similar results are found for the remainder. Worth mentioning is also the computation time over the evaluation session. As discussed earlier, this time depends on the number of evaluation trials already processed. Fig. 7 shows the CPU-time needed to classify the i-th evaluation trial using an Intel Core i7-2600 @ 3.40 GHz processor and 16 GB RAM. Solid black line corresponds to the polynomial growth $O(i^2)$.

Table 2 shows the comparison of our methods against the winner of the BCI Competition dataset 2a [10], other seven recent methods tested on this dataset [8,36–41] as well as the EM-LDA [18] and PMean [24]. Likewise, LDA results were reported as a baseline. Cohen's Kappa coefficient has been considered because just this evaluation criterion was used in the BCI Competition IV dataset 2a. The highest performance for each subject is highlighted in boldface. Although there is high variability in classification performance over subjects, overall our methods with adaptive stage clearly outperform all previously published methods. Table 3 presents the results of Wilcoxon statistical test [48] to evaluate the statistical significance of the difference between the performance of SUSS-SRKDA ($\eta>0$) and the other methods reported in Table 2.

Finally, the presence of outliers could affect the sequential estimation over the evaluation session. The BCI Competition IV

Table 2
Kappa values of the proposed and competing methods on the BCI Competition IV Dataset 2a.

Method	Subjects									AVG
	A1	A2	A3	A4	A5	A6	A7	A8	A9	
Ang et al. [10]	0.68	0.42	0.75	0.48	0.40	0.27	0.77	0.75	0.61	0.57
Gouy-Pailler et al. [8]	0.66	0.42	0.77	0.51	0.50	0.21	0.30	0.69	0.46	0.50
Wang [36]	0.67	0.49	0.77	0.59	0.52	0.31	0.48	0.75	0.65	0.58
Barachant et al. [37]	0.74	0.38	0.72	0.50	0.26	0.34	0.69	0.71	**0.76**	0.57
Wang et al. [38]	0.56	0.41	0.43	0.41	**0.68**	**0.48**	0.80	0.72	0.63	0.57
Kam et al. [39]	0.74	0.35	0.76	0.53	0.38	0.31	0.84	0.74	0.74	0.60
Asensio-Cubero et al. [40]	0.75	0.50	0.74	0.40	0.19	0.41	0.78	0.72	0.78	0.59
Asensio-Cubero et al. [41]	0.76	0.32	0.76	0.47	0.31	0.34	0.59	0.76	0.74	0.56
LDA	0.76	0.41	0.83	0.56	0.35	0.26	0.79	0.73	0.53	0.58
Blumberg et al. [18] (EM-LDA)	0.59	0.41	0.82	0.57	0.38	0.29	0.79	0.80	0.72	0.60
Vidaurre et al. [24] (PMean)	0.76	0.38	0.87	0.60	0.46	0.34	0.77	0.76	0.74	0.63
SRKDA($\eta=0$)	0.81	0.48	0.74	0.63	0.42	0.34	0.86	0.81	0.57	0.63
SUSS-SRKDA($\eta=0$)	0.81	0.47	0.80	0.61	0.38	0.31	**0.92**	**0.84**	0.62	0.64
SRKDA($\eta>0$)	0.81	0.49	0.87	0.65	0.53	0.38	0.87	0.83	0.73	0.68
SUSS-SRKDA($\eta>0$)	**0.83**	**0.51**	**0.88**	**0.68**	0.56	0.35	0.90	**0.84**	0.75	**0.70**

Table 3
Wilcoxon test results (p-values) evaluating the statistical significance of the difference between the performance of SUSS-SRKDA ($\eta>0$) and the other methods reported in Table 2.

Subject	p-Value
SUSS-SRKDA ($\eta>0$) versus Ang et al. [10]	0.0020
SUSS-SRKDA ($\eta>0$) versus Gouy-Pailler et al. [8]	0.0020
SUSS-SRKDA ($\eta>0$) versus Wang [36]	0.0020
SUSS-SRKDA ($\eta>0$) versus Barachant et al. [37]	0.0059
SUSS-SRKDA ($\eta>0$) versus Wang et al. [38]	0.0762
SUSS-SRKDA ($\eta>0$) versus Kam et al. [39]	0.0020
SUSS-SRKDA ($\eta>0$) versus Asensio-Cubero et al. [40]	0.0020
SUSS-SRKDA ($\eta>0$) versus Asensio-Cubero et al. [41]	0.0176
SUSS-SRKDA ($\eta>0$) versus LDA	0.0020
SUSS-SRKDA ($\eta>0$) versus Blumberg et al. [18] (EM-LDA)	0.0020
SUSS-SRKDA ($\eta>0$) versus Vidaurre et al. [24] (PMean)	0.0020
SUSS-SRKDA ($\eta>0$) versus SRKDA ($\eta=0$)	0.0020
SUSS-SRKDA ($\eta>0$) versus SUSS-SRKDA ($\eta=0$)	0.0117
SUSS-SRKDA ($\eta>0$) versus SRKDA ($\eta>0$)	0.0391

dataset 2a includes several trials marked as artifacts by an expert. Following the competition criterion, we had discarded these trials in training and evaluation sessions. Then, we repeated the experiments including both valid and invalid trials to show the resilience of our methods to outliers. The results show that the performances of the proposed methods are not negatively affected by outliers. SRKDA ($\eta=0$), SUSS-SRKDA ($\eta=0$), SRKDA ($\eta>0$), and SUSS-SRKDA ($\eta>0$) produce mean kappa values of 0.62, 0.64, 0.68, and 0.70, respectively. They are equal or very similar to the ones obtained rejecting invalid trials: 0.63, 0.64, 0.68, and 0.70 (Table 1). Wilcoxon's test reveals no significant differences (p-values=0.8828, 0.9834, 0.8798, and 0.7785).

5. Discussion and conclusions

In this paper, we study the combination of adaptive processing and semi-supervised learning to discriminate four imaginary motor tasks. Four classification algorithms, SRKDA and SUSS-SRKDA with or without adaptive processing stage are presented. A filter bank and the CSP algorithm are employed in the feature extraction stage. The proposed approaches are assessed on EEG signals of 9 subjects provided by the BCI Competition IV dataset 2a. A comparative study amongst all our approaches and other nine methods using the same dataset is carried out. Our findings suggest that the method joining semi-supervised learning and adaptive processing can significantly increase the classification performance in multiclass settings. SUSS-SRKDA with adaptive processing yields the highest average classification performance: accuracy 77% and Cohen's kappa coefficient 0.70.

Classification performance analysis showed that online semi-supervised learning by itself obtains higher accuracy and kappa on average across the 9 subjects. This result agrees with other previous studies, where the classification accuracy is improved with the introduction of unlabeled data into model training [21–23]. The increase of performance can be associated with the enhancement of adaptability to non-stationary EEG signals since self-training uses new incoming data to update the classifier. However, our results also show that further greater performance is reached when adaptive processing before classification is introduced. The main concern of self-training is that feature vectors from different sessions are considered to build and update a single model. As was reported in previous studies [25,49], feature vectors extracted from EEG data of different sessions follow different probability distributions. The non-stationarity often leads to lower classification performances when it is assumed that a single model is valid across different sessions. This is a crucial point given that self-training suffers from the mistake-reinforcing danger implying that some classification mistakes can reinforce itself [50]. Therefore, self-training algorithms should include some sort of procedure that give more importance to the most recent feature vectors or minimize the mismatch between sessions before classification. Vidaurre et al. [24] already alluded to this idea by proposing three adaptive versions of the well-known LDA method, where the bias term and covariance matrices were estimated recursively by an unsupervised adaptive procedure that involved a forgetting factor. The methodology proposed in our paper consists of a semi-supervised method in addition to reducing session-to-session non-stationarity before classification. Our results support that minimizing the inter-session difference boosts the effectiveness of self-training. The adaptive stage removes the non-stationarity in terms of fluctuations in the global mean. EWMA estimates the changes in the global mean across the trials giving more importance to the more recent ones. The evolution of the global mean over the time is independent of the tasks and can be addressed without the need to know the labels.

Fig. 8 illustrates the impact of the adaptive stage in the density distribution of the projected features after applying the Eq. (11) of the SRKDA method. Since features vary across the time, we select the 2 second window where the highest kappa value of the subject 9 is produced. The mean and variances of the feature distributions for four classes are represented with darker-grey ellipsoids (training session) and lighter-grey ellipsoids (evaluation session). On the left,

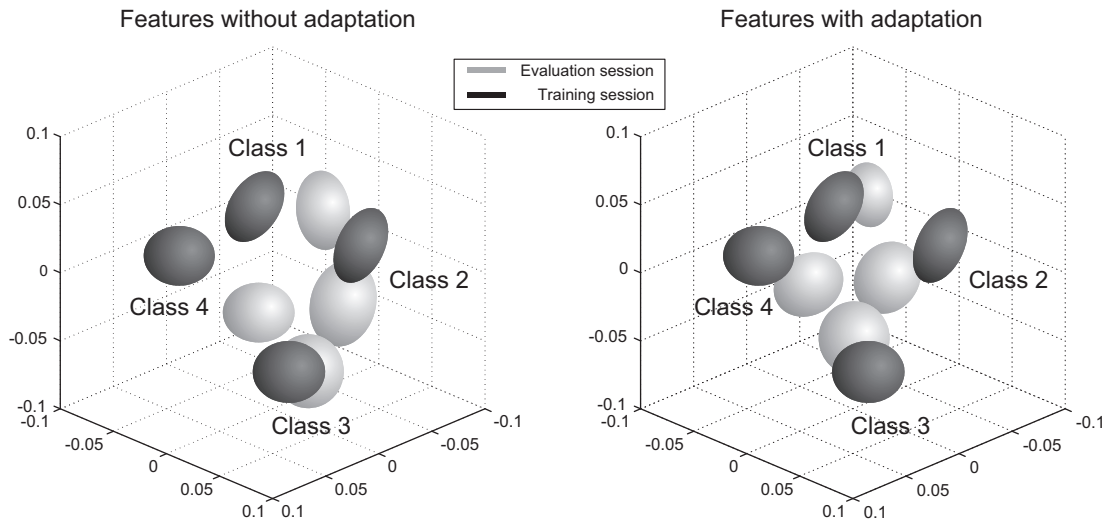


Fig. 8. Representation of the projected feature distributions by SRKDA with or without adaptation. Data from subject 9 from the BCI Competition IV, dataset 2a. The mean and variances of the feature distributions are represented for four classes with darker-gray ellipsoids (training session) and lighter-gray ellipsoids (evaluation session). On the left, we can clearly see an inter-session mean shift and rotation on the distributions. On the right, the same distributions are shown, but the shifts of class means are decreased.

we can clearly see inter-session mean shifts and rotations on the distributions. On the right, the same distributions are shown, but the mean shifts are reduced. Note that new patterns are classified by SRKDA on the basis of the Euclidean distance to the projected mean for each class according to Eq. (12), which is computed during the training session. Hence, reducing the inter-session mean shifts indeed achieves to decrease the misclassification rate.

Within the comparative analysis shown in the Table 2, SUSS-SRKDA with or without adaptive stage outperforms EM-LDA and PMean. The key point is that, whereas EM-LDA and PMean are based on LDA, SUSS-SRKDA updates a more powerful classifier that takes advantage of a regularized linear kernel approach. LDA assumes the covariance matrices of both classes to be equal and relies on the estimation of this common covariance matrix, which might be highly biased [33]. The estimation results in a high variability when the number of samples is small compared to the dimensionality. It is recommended to use, at least, five to ten times as many training samples per class as the dimensionality of feature vectors [51]. Indeed, due to this circumstance, it can be observed the optimal number of features found by cross-validation is lower for LDA than SRKDA (35 for LDA versus 97 for SRKDA). At this point, regularization [52] could be a simple and effective improvement of the EM-LDA and PMean methods. On the other hand, other more complex model involving non-linear kernels could be tried to improve the performance. However, some studies report non-linear methods perform only slightly better in motor EEG classification [53,54]. Furthermore, Liao et al. [20] evaluated a TSVM with either linear or Gaussian kernel on a set of EEG recordings of three subjects performing three mental tasks finding the non-linear method did not provide superior performance.

Another benefit of our sequential updating method with SRKDA is the lower computational requirement with respect to KDA. KDA cannot efficiently incorporate a new data sample as it becomes available [32]. Retraining the model with the augmented dataset by means of the standard KDA is computationally expensive because it involves eigen-decomposition of the kernel matrix, which can be unacceptable when a large amount of training samples exists. Instead, SRKDA only needs to solve a linear system of equations that can be efficiently solved by Cholesky decomposition. Comparing with other semi-supervised kernel classifiers in MI-BCI, semi-supervised SVMs have been also shown to improve the performance of the brain state classification using unlabeled data [20,21] but it suffers from high computational complexity, which might make its online applicability difficult.

Some limitations of this research have to be considered. Firstly, although semi-supervised learning and adaptation processing are able to increase classification performance, there may be even more room for improvement. The adaptive stage only deals with the non-stationarity in terms of changes in the global mean. Note that the classification performance is also affected by fluctuations in the position of each class-centroid as well as rotations in the class distributions. In this research, these issues are compensated to some extent by the self-training algorithm, which promptly incorporates new incoming data to update the classification model. However, as above discussed, forgetting the most out-of-date information could yield higher performance. On the other hand, our feature extraction method based on spatial filtering with CSP does not consider inter-session fluctuations. For instance, Li and Guan [19] showed that performance can be improved updating jointly spatial filters and classifier with semi-supervised learning. Secondly, although the complexity of sequential updating has been considerably reduced thanks to the use of SRKDA instead of KDA, it still grows with more and more data samples involved in updating step. Thirdly, and finally, further analysis about the performance gain with semi-supervised learning is required. It is of interest to understand why semi-supervised learning improves performance in some subjects whereas reduces it for others. Future work should find new methods that embed forgetting factors into semi-supervised learning or procedures that minimize the inter-session non-stationarity in terms of the position of the class-centroids as well as the rotations before classification. It is of particular importance in multiclass settings. Likewise, spatial filter updating along with SUSS-SRKDA should be tested as a mean of further increasing the performance. For the sake of the applicability in real environments, approaches whose complexity does not monotonically increase as more trials are processed should be considered. In this regard, approaches that limit the complexity by restarting the sequential updating procedure at a regular basis could be contemplated. Lastly, theoretical studies based on probably approximately correct (PAC) learning theory [55,56] could provide a framework for capturing which subjects unlabeled data can help.

Please cite this article as: L.F. Nicolas-Alonso, et al., Adaptive semi-supervised classification to reduce intersession non-stationarity in multiclass motor imagery-based brain–computer interfaces, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.02.005

In summary, this work presents a new methodology that involves adaptive processing and a semi-supervised method such as self-training. Either combined or separate use of these two elements has been evaluated on a multiclass environment. This study provides evidences that adaptive processing before classification can highlight the advantages of self-training. Although self-training enhances by itself the adaptability to non-stationary EEG data, reducing the intersession non-stationarity increases the performance of multiclass motor imagery-based BCIs. In addition, this work introduces a new sequential updating semi-supervised algorithm. It has the advantage of being developed in a recursive manner, which hugely reduces the computational effort. Finally, we would like to note that the proposed classification methods are of great interest for real-life BCI systems because they mean that model trained during the first session of training can be used with acceptable classification accuracy during the following sessions. Furthermore, they are by no means limited to multiclass MI-BCIs, but also they are applicable to other kinds of single-trial EEG classification problems.

Acknowledgments

This research was supported in part by the Project Cero 2011 on Ageing from Fundación General CSIC, Obra Social La Caixa and CSIC (Spain) and a Grant by the Ministerio de Economía y Competitividad (Spain) and FEDER under project TEC2011-22987. L.F. Nicolas-Alonso was in receipt of a PIF-UVa Grant from Universidad de Valladolid.

Appendix A

In this appendix we give a proof of the convergence of SS-SRKDA. For the SS-SRKDA, the associated cost function at the j-th iteration is defined as [32]

$$\Psi^{(j)}(\boldsymbol{\alpha}) = \sum_{i=1}^{M} (\Theta(\boldsymbol{x}_i, \boldsymbol{\alpha}) - \hat{y}_i^{(j-1)})^2 + (\Theta(\boldsymbol{x}_i, \boldsymbol{\alpha}) - \hat{y}_{M+1}^{(j-1)})^2 + \delta \|\Theta\|_K^2, \tag{A.1}$$

where $\hat{y}_i^{(j)}$ is the i-th element of $\hat{\boldsymbol{y}}^{(j)}$, $\Theta(\boldsymbol{x}, \boldsymbol{\alpha})$ is the projective function in the feature space defined in Eq. (9), δ is the regularization term, and $\|\|_K$ is the corresponding norm in a RKHS defined by the positive definite kernel $K(\mathbf{x}, \mathbf{y})$ [57]. SS-SRKDA converges if the associated cost for each iteration $\Psi^{(j)}(\boldsymbol{\alpha}^{(j)})$ also converges.

According to Steps 1 and 2 of SS-SRKDA algorithm, $\boldsymbol{\alpha}^{(0)}$ is computed with dataset $D = \{(\boldsymbol{x}_i, l_i)\}_{i=1}^{M}$, and the predicted label $\hat{l}^{(0)}$ of $\hat{\boldsymbol{x}}$ obtained.

According to the Step 3, for iteration $j-1 \quad (j \geq 2)$, $\boldsymbol{\alpha}^{(j-1)}$ is computed with the augmented dataset $\hat{D}^{(j-1)} = D \cup \{\hat{\boldsymbol{x}}, \hat{l}^{(j-2)}\}$. The cost of this iteration is given by

$$\Psi^{(j-1)}(\boldsymbol{\alpha}^{(j-1)}) = \sum_{i=1}^{M} (\Theta(\boldsymbol{x}_i, \boldsymbol{\alpha}^{(j-1)}) - \hat{y}_i^{(j-2)})^2 + (\Theta(\hat{\boldsymbol{x}}, \boldsymbol{\alpha}^{(j-1)}) - \hat{y}_{M+1}^{(j-2)})^2 + \delta \|\Theta\|_K^2. \tag{A.2}$$

With $\boldsymbol{\alpha}^{(j-1)}$, the label $\hat{l}^{(j-1)}$ of $\hat{\boldsymbol{x}}$ and the corresponding $\hat{\boldsymbol{y}}^{(j-1)}$ are found. Replacing $\hat{\boldsymbol{y}}^{(j-2)}$ with $\hat{\boldsymbol{y}}^{(j-1)}$, we obtain the updated cost

$$\hat{\Psi}^{(j-1)}(\boldsymbol{\alpha}^{(j-1)}) = \sum_{i=1}^{M} (\Theta(\boldsymbol{x}_i, \boldsymbol{\alpha}^{(j-1)}) - \hat{y}_i^{(j-1)})^2 + (\Theta(\hat{\boldsymbol{x}}, \boldsymbol{\alpha}^{(j-1)}) - \hat{y}_{M+1}^{(j-1)})^2 + \delta \|\Theta\|_K^2. \tag{A.3}$$

It is straightforward that $\hat{\Psi}^{(j-1)}(\boldsymbol{\alpha}^{(j-1)}) \leq \Psi^{(j-1)}(\boldsymbol{\alpha}^{(j-1)})$. If the equality holds then the predicted label of the augmented dataset was not actually updated and the algorithm has converged. Otherwise, the model is retrained in the next iteration with the augmented training dataset $\hat{D}^{(j)} = D \cup \{\hat{\boldsymbol{x}}, \hat{l}^{(j-1)}\}$ to obtain $\boldsymbol{\alpha}^{(j)}$. The new cost is given by

$$\Psi^{(j)}(\boldsymbol{\alpha}^{(j)}) = \sum_{i=1}^{M} (\Theta(\boldsymbol{x}_i, \boldsymbol{\alpha}^{(j)}) - \hat{y}_i^{(j-1)})^2 + (\Theta(\hat{\boldsymbol{x}}, \boldsymbol{\alpha}^{(j)}) - \hat{y}_{M+1}^{(j-1)})^2 + \delta \|\Theta\|_K^2, \tag{A.4}$$

where $\Psi^{(j)}(\boldsymbol{\alpha}) = \hat{\Psi}^{(j-1)}(\boldsymbol{\alpha})$ for any $\boldsymbol{\alpha}$. It is known that $\Psi^{(j)}(\boldsymbol{\alpha}^{(j)}) \leq \Psi^{(j)}(\boldsymbol{\alpha})$ for any $\boldsymbol{\alpha}$, since $\Psi^{(j)}(\boldsymbol{\alpha})$ has a minimum at $\boldsymbol{\alpha}^{(j)}$. Hence, after each iteration, the cost is non-negative and becomes lower, $\Psi^{(j)}(\boldsymbol{\alpha}^{(j)}) \leq \Psi^{(j)}(\boldsymbol{\alpha}^{(j-1)}) = \hat{\Psi}^{(j-1)}(\boldsymbol{\alpha}^{(j-1)}) \leq \Psi^{(j-1)}(\boldsymbol{\alpha}^{(j-1)})$. Therefore, the algorithm will arrive at the infinitum where the classifier and the labels remain unchanged, i.e., SRKDA converges.

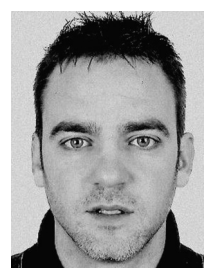
References

[1] L.F. Nicolas-Alonso, J. Gomez-Gil, Brain computer interfaces, a review, Sensors 12 (2012) 1211–1279.
[2] J.R. Wolpaw, D.J. McFarland, T.M. Vaughan, Brain–computer interface research at the Wadsworth Center, IEEE Trans. Rehabil. Eng. 8 (2000) 222–226.
[3] G. Pfurtscheller, C. Neuper, G. Muller, B. Obermaier, G. Krausz, A. Schlogl, R. Scherer, B. Graimann, C. Keinrath, D. Skliris, Graz-BCI: state of the art and clinical applications, IEEE Trans. Neural Syst. Rehabil. Eng. 11 (2003) 1–4.
[4] B. Blankertz, F. Losch, M. Krauledat, G. Dornhege, G. Curio, K.-R. Muller, The Berlin Brain–Computer Interface: accurate performance from first-session in BCI-naive subjects, IEEE Trans. Biomed. Eng. 55 (2008) 2452–2462.
[5] G. Dornhege, B. Blankertz, G. Curio, K.-R. Muller, Boosting bit rates in noninvasive EEG single-trial classifications by feature combination and multi-class paradigms, IEEE Trans. Biomed. Eng. 51 (2004) 993–1002.
[6] H.H. Ehrsson, S. Geyer, E. Naito, Imagery of voluntary movement of fingers, toes, and tongue activates corresponding body-part-specific motor representations, J. Neurophysiol. 90 (2003) 3304–3316.
[7] G. Townsend, B. Graimann, G. Pfurtscheller, A comparison of common spatial patterns with complex band power features in a four-class BCI experiment, IEEE Trans. Biomed. Eng. 53 (2006) 642–651.
[8] C. Gouy-Pailler, M. Congedo, C. Brunner, C. Jutten, G. Pfurtscheller, Nonstationary brain source separation for multiclass motor imagery, IEEE Trans. Biomed. Eng. 57 (2010) 469–478.
[9] H.I. Suk, S.W. Lee, Subject and class specific frequency bands selection for multiclass motor imagery classification, Int. J. Imag. Syst. Technol. 21 (2011) 123–130.
[10] K.K. Ang, Z.Y. Chin, C. Wang, C. Guan, H. Zhang, Filter Bank Common Spatial Pattern Algorithm on BCI Competition IV Datasets 2a and 2b, Front. Neurosci. 6 (2012) 39.
[11] N. Robinson, C. Guan, A. Vinod, K.K. Ang, K.P. Tee, Multi-class EEG classification of voluntary hand movement directions, J. Neural Eng. 10 (2013) 056018.
[12] I. Guler, E.D. Ubeyli, Multiclass support vector machines for EEG-signals classification, IEEE Trans. Inf. Technol. Biomed. 11 (2007) 117–126.
[13] P. Xu, P. Yang, X. Lei, D. Yao, An enhanced probabilistic LDA for multi-class brain computer interface, PloS One 6 (2011) e14634.
[14] P. Shenoy, M. Krauledat, B. Blankertz, R.P.N. Rao, K.R. Müller, Towards adaptive classification for BCI, J. Neural Eng. 3 (2006) R13.
[15] M. Fatourechi, A. Bashashati, R.K. Ward, G.E. Birch, EMG and EOG artifacts in brain computer interface systems: a survey, Clin. Neurophysiol. 118 (2007) 480–494.
[16] X. Zhu, Semi-supervised learning literature survey, Computer Science, University of Wisconsin-Madison, 2 (2006) 3.
[17] G. Liu, G. Huang, J. Meng, D. Zhang, X. Zhu, Improved GMM with parameter initialization for unsupervised adaptation of brain–computer interface, Int. J. Numer. Meth. Bio-Med. Eng. 26 (2010) 681–691.
[18] J. Blumberg, J. Rickert, S. Waldert, A. Schulze-Bonhage, A. Aertsen, C. Mehring, Adaptive classification for brain computer interfaces, in: Proceedings of IEEE Engineering in Medicine and Biology Society, IEEE, Lyon, France, 2007, pp. 2536–2539.
[19] Y. Li, C. Guan, An extended EM algorithm for joint feature extraction and classification in brain–computer interfaces, Neural Comput. 18 (2006) 2730–2761.
[20] X. Liao, D. Yao, C. Li, Transductive SVM for reducing the training effort in BCI, J. Neural Eng. 4 (2007) 246.
[21] J. Qin, Y. Li, W. Sun, A semisupervised support vector machines algorithm for BCI systems, Comput. Intell. Neurosci. 2007 (2007) 94397.
[22] Y. Li, C. Guan, H. Li, Z. Chin, A self-training semi-supervised SVM algorithm and its application in an EEG-based brain computer interface speller system, Pattern Recogn. Lett. 29 (2008) 1285–1294.
[23] Z. Gu, Z. Yu, Z. Shen, Y. Li, An online semi-supervised brain–computer interface, IEEE Trans. Biomed. Eng. 60 (2013) 2614–2623.
[24] C. Vidaurre, M. Kawanabe, P. von Bünau, B. Blankertz, K.R. Müller, Toward unsupervised adaptation of LDA for brain–computer interfaces, IEEE Trans. Biomed. Eng. 58 (2011) 587–597.

[25] M. Sugiyama, M. Krauledat, K.-R. Müller, Covariate shift adaptation by importance weighted cross validation, J. Mach. Learn. Res. 8 (2007) 985–1005.
[26] Y. Li, H. Kambara, Y. Koike, M. Sugiyama, Application of covariate shift adaptation techniques in brain–computer interfaces, IEEE Trans. Biomed. Eng. 57 (2010) 1318–1324.
[27] M. Arvaneh, C. Guan, K.K. Ang, C. Quek, EEG data space adaptation to reduce intersession nonstationarity in brain–computer interface, Neural Comput. 25 (2013) 2146–2171.
[28] P. Sykacek, S.J. Roberts, M. Stokes, Adaptive BCI based on variational Bayesian Kalman filtering: an empirical evaluation, IEEE Trans. Biomed. Eng. 51 (2004) 719–727.
[29] J.W. Yoon, S.J. Roberts, M. Dyson, J.Q. Gan, Adaptive classification for brain computer interface systems using sequential Monte Carlo sampling, Neural Netw. 22 (2009) 1286–1294.
[30] J.W. Yoon, S.J. Roberts, M. Dyson, J.Q. Gan, Bayesian inference for an adaptive Ordered Probit model: an application to Brain Computer Interfacing, Neural Netw. 24 (2011) 726–734.
[31] N.J. Higham, Accuracy and Stability of Numerical Algorithms, Second Edition, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, USA, 2002.
[32] D. Cai, X. He, J. Han, Speed up kernel discriminant analysis, VLDB J. 20 (2011) 21–33.
[33] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, K. Mullers, Fisher discriminant analysis with kernels, in: Proceedings of IEEE Signal Processing Society Workshop, IEEE, Madison, WI, 1999, pp. 41–48.
[34] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Comput. 12 (2000) 2385–2404.
[35] M. Tangermann, K.-R. Müller, A. Aertsen, N. Birbaumer, C. Braun, C. Brunner, R. Leeb, C. Mehring, K.J. Miller, G. Mueller-Putz, G. Nolte, G. Pfurtscheller, H. Preissl, G. Schalk, A. Schlögl, C. Vidaurre, S. Waldert, B. Blankertz, Review of the BCI Competition IV, Front. Neurosci. 6 (2012) 55.
[36] H. Wang, Multiclass filters by a weighted pairwise criterion for EEG single-trial classification, IEEE Trans. Biomed. Eng. 58 (2011) 1412–1420.
[37] A. Barachant, S. Bonnet, M. Congedo, C. Jutten, Multiclass brain–computer interface classification by Riemannian Geometry, IEEE Trans. Biomed. Eng. 59 (2012) 920–928.
[38] D. Wang, D. Miao, G. Blohm, Multi-class motor imagery EEG decoding for brain–computer interfaces, Front. Neurosci. 6 (2012) 00151.
[39] T.-E. Kam, H.-I. Suk, S.-W. Lee, Non-homogeneous spatial filter optimization for ElectroEncephaloGram (EEG)-based motor imagery classification, Neurocomputing 108 (2013) 58–68.
[40] J. Asensio-Cubero, J. Gan, R. Palaniappan, Multiresolution analysis over simple graphs for brain computer interfaces, J. Neural Eng. 10 (2013) 046014.
[41] J. Asensio-Cubero, J.Q. Gan, R. Palaniappan, Extracting optimal tempo-spatial features using local discriminant bases and common spatial patterns for brain computer interfacing, Biomed. Signal. Proces. 8 (2013) 772–778.
[42] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, K.R. Muller, Optimizing spatial filters for robust EEG single-trial analysis, IEEE Signal Process. Mag. 25 (2008) 41–56.
[43] J.M. Lucas, M.S. Saccucci, Exponentially weighted moving average control schemes: properties and enhancements, Technometrics 32 (1990) 1–12.
[44] T.M. Cover, Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, IEEE Trans. Electron. EC-14 (1965) 326–334.
[45] G.W. Stewart, Matrix Algorithms Volume 1: Basic Decompositions, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, USA, 1998.
[46] G. Dornhege, J.d.R. Millán, T. Hinterberger, D. McFarland, K.-R. Müller, Toward Brain–computer Interfacing, MIT press Cambridge, Boston, MA, USA, 2007.
[47] G. Dornhege, B. Blankertz, M. Krauledat, F. Losch, G. Curio, K.-R. Muller, Combined optimization of spatial and temporal filters for improving brain–computer interfacing, IEEE Trans. Biomed. Eng. 53 (2006) 2274–2281.
[48] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[49] L. Yan, H. Kambara, Y. Koike, M. Sugiyama, Application of covariate shift adaptation techniques in brain–computer interfaces, IEEE Trans. Biomed. Eng. 57 (2010) 1318–1324.
[50] X. Zhu, J. Lafferty, R. Rosenfeld, Semi-supervised Learning with Graphs, Carnegie Mellon University, Language Technologies Institute, School of Computer Science, Pittsburgh, PA, USA, 2005.
[51] S.J. Raudys, A.K. Jain, Small sample size effects in statistical pattern recognition: recommendations for practitioners, IEEE Trans. Pattern Anal. Mach. Intell. 13 (1991) 252–264.
[52] J.H. Friedman, Regularized discriminant analysis, J. Am. Stat. Assoc. 84 (1989) 165–175.
[53] D. Garrett, D.A. Peterson, C.W. Anderson, M.H. Thaut, Comparison of linear, nonlinear, and feature selection methods for EEG signal classification, IEEE Trans. Neural Syst. Rehabil. Eng. 11 (2003) 141–144.
[54] K.-R. Muller, C.W. Anderson, G.E. Birch, Linear and nonlinear methods for brain–computer interfaces, IEEE Trans. Neural Syst. Rehabil. Eng. 11 (2003) 165–169.
[55] S. Dasgupta, M.L. Littman, D. McAllester, PAC generalization bounds for co-training, Adv. Neural Inf. Process. Syst. 1 (2002) 375–382.
[56] M.-F. Balcan, A. Blum, A PAC-style Model for Learning From Labeled and Unlabeled Data, Learning Theory, Springer, Germany (2005) 111–126.
[57] V. Vapnik, Statistical Learning Theory, Wiley, University of Michigan, MI, USA, 1998.

Luis F. Nicolas-Alonso was born in Valladolid, Spain in 1988. He received the M.S. degree in telecommunication engineering, in 2011, from the University of Valladolid (Spain), where he is currently working toward the Ph.D. degree in the Department of Signal Theory and Communications and is a scholarship holder. He is a member of the Biomedical Engineering Group. His current research interests include signal processing and pattern recognition techniques applied to brain–computer interfaces.

Rebeca Corralejo was born in Burgos, Spain, in 1983. She received the M.S. degree in telecommunication engineering, in 2008, from the University of Valladolid (Spain), where she is currently working toward the Ph. D. degree in the Department of Signal Theory and Communications and is a scholarship holder. She is a member of the Biomedical Engineering Group. Her current research interests include biomedical signal processing applied to brain–computer interfaces and development of assistive applications.



Javier Gomez-Pilar was born in Alicante, Spain in 1983. He received the M.S. degree in telecommunication engineering, in 2012, from the University of Valladolid, Valladolid, Spain, where he is currently working toward the Ph.D. degree in the Department of Signal Theory and Communications. He is a Researcher at the Biomedical Engineering Group. His current research interests include signal processing, complex network theory, and rehabilitation techniques using brain–computer interfaces.

Daniel Alvarez was born in Bembibre, Spain, in 1978. He received the M.S. degree in telecommunication engineering and the Ph.D. degree from the University of Valladolid (Spain) in 2005 and 2011, respectively. Since 2005, he is a member of the Biomedical Engineering Group of the University of Valladolid. He is currently a researcher in the Department of Signal Theory and Communications of the University of Valladolid. His research interests include multivariate analysis and pattern recognition of biomedical signals. His work focuses on the development of novel methodologies aimed at assisting in the diagnosis of sleep disorders, as well as on the design of new assistive tools based on brain–computer interfaces for disabled/dependent people.

Roberto Hornero was born in Plasencia, Spain, in 1972. He received the M.S. degree in telecommunication engineering and the Ph.D. degree from the University of Valladolid (Spain), in 1995 and 1998, respectively. He is currently Professor in the Department of Signal Theory and Communications at the University of Valladolid. His main research interest is spectral and nonlinear analysis of biomedical signals to help physicians in the clinical diagnosis. He founded the Biomedical Engineering Group in 2004. The research interests of this group are connected with the field of nonlinear dynamics, chaotic theory, and wavelet transform with applications in biomedical signal and image processing.

Please cite this article as: L.F. Nicolas-Alonso, et al., Adaptive semi-supervised classification to reduce intersession non-stationarity in multiclass motor imagery-based brain–computer interfaces, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2015.02.005