Pattern Recognition 70 (2017) 152–162

Contents lists available at ScienceDirect

Pattern Recognition

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

Higher order spectral regression discriminant analysis (HOSRDA): A tensor feature reduction method for ERP detection

Mina Jamshidi Idaji*, Mohammad B. Shamsollahi, Sepideh Hajipour Sardouie

Biomedical Signal and Image Processing Lab (BiSIPL), Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Article history:
Received 27 October 2016
Revised 17 March 2017
Accepted 7 May 2017
Available online 8 May 2017

Keywords:
HOSRDA
Tensor decomposition
Tucker decomposition
P300 speller
BCI
SRDA
LDA
HODA

ABSTRACT

Tensors are valuable tools to represent Electroencephalogram (EEG) data. Tucker decomposition is the most used tensor decomposition in multidimensional discriminant analysis and tensor extension of Linear Discriminant Analysis (LDA), called Higher Order Discriminant Analysis (HODA), is a popular tensor discriminant method used for analyzing Event Related Potentials (ERP). In this paper, we introduce a new tensor-based feature reduction technique, named Higher Order Spectral Regression Discriminant Analysis (HOSRDA), for use in a classification framework for ERP detection. The proposed method (HOSRDA) is a tensor extension of Spectral Regression Discriminant Analysis (SRDA) and casts the eigenproblem of HODA to a regression problem. The formulation of HOSRDA can open a new framework for adding different regularization constraints in higher order feature reduction problem. Additionally, when the dimension and number of samples is very large, the regression problem can be solved via efficient iterative algorithms. We applied HOSRDA on data of a P300 speller from BCI competition III and reached average character detection accuracy of 96.5% for the two subjects. HOSRDA outperforms almost all of other reported methods on this dataset. Additionally, the results of our method are fairly comparable with those of other methods when 5 and 10 repetitions are used in the P300 speller paradigm.

1. Introduction

Tensors are natural representations of data containing information in higher order modes. In the recent years, tensor-based signal processing [1,2] and dimensionality reduction methods [3–14] have achieved tremendous popularity for analyzing multidimensional data. Working with such data in the "flat-world" of matrices may prevent us from making full use of the information provided in each mode and also the interactions among them. As a matter of fact, tensor decompositions are tools that exploit the multidimensional nature of tensor data to discover the hidden information and interactive relations among different modes. Electroencephalogram (EEG) data is one example, for which tensors are good representations: EEG naturally includes information in different modes of trials, time, frequency, channel, etc. [2] and various trends in decomposing EEG data with tensor decomposition tools have emerged in recent decade [15].

The spectrum of popularity of using tensor tools for EEG data extends to a wide range of applications, from epileptic EEG analysis [16–21], to Brain-Computer Interface [22], and to Event Related Potential (ERP) analysis. BCIs use ERP, Steady State Visual Evoked Potential (SSVEP), or Event Related Desynchronization/Synchronzation (ERD/ERS), such as motor imagery. In [23,24], tensor decomposition has been used for motor imagery EEG. Additionally, decomposing the data of SSVEP-based BCIs has been widely experimented. Various tensor extensions of Canonical Correlation Analysis (CCA) have been adapted to optimize the reference signals of the CCA for SSVEP frequency recognition [25–28]. Linked component analysis methods and their tensor extensions have been thoroughly reviewed in [29] and their application on biomedical data, such as SSVEP-based BCIs, has been illustrated.

Almost all the related studies of assessing ERP have exploited whether Canonical Polyadic (CP) decomposition, also called Parallel Factor Analysis (PARAFAC), or Tucker decomposition [15]. Although the very first attempts at applying CP decomposition to ERP data goes back to the late 1980s with the name of *topographic component analysis* [30–32], in recent years many researches have used CP and its variants as powerful tools for analyzing and feature reduction of ERP data [33–39].

Due to its flexible nature, Tucker decomposition has been widely used as a powerful tool for tensor-based discriminant analysis methods, and with application for ERP data. Higher Order Discriminant Analysis (HODA) [7], also called DATER [8], is a tensor extension of conventional Linear Discriminant Analysis (LDA) feature reduction and is one of the most popular tensor discriminant

* Corresponding author.
E-mail address: minajamshidi91@gmail.com (M. Jamshidi Idaji).

http://dx.doi.org/10.1016/j.patcog.2017.05.004
0031-3203/

analysis methods. Recently the application of HODA in ERP-based BCI has been shown. Onishi et al. have applied HODA on data of a P300 speller and reported a good performance [40]. Also, Spatial-Temporal Discriminant Analysis (STDA), which is a special form of HODA, is introduced to be applied on data of P300-based paradigm [41]. To the best of our knowledge, STDA is the state-of-the-art multiway discriminant analysis method used for feature reduction in ERP-based BCIs.

In this paper we introduce a new tensor-based feature reduction technique. Due to this main direction, we hesitate to discuss tensor decomposition algorithms [2,42–44]. Our proposed method is named Higher Order Spectral Regression Discriminant Analysis (HOSRDA), which is a tensor extension of Spectral Regression Discriminant Analysis (SRDA) [45]. SRDA is a variant of LDA that casts the eigenvalue problem of LDA to solving a set of linear equations. HOSRDA benefits from all the advantages of SRDA over LDA: HOSRDA solves the problem of HODA by solving a set of linear equations, thus it can provide the ability to impose different regularizations on the subspace basis factors. Furthermore, HOSRDA can benefit from low complexity algorithms for solving linear equations and therefore omit the need for computationally demanding eigenvalue decomposition. Additionally, in the cases that the scatter matrices of HODA are ill-conditioned (specially in the early iterations [7]), the eigenvalue decomposition does not have stable solution; HOSRDA does not suffer from this, since it uses regression as a building block that can be utilized in order to overcome the stability problem.

HOSRDA is accompanied by LDA classifier; the whole package of feature reduction and classification is noted by "HOSRDA+LDA" and it is exploited for classification of P300 speller data from BCI competition III-dataset II and compared to STDA, as the state-of-the-art tensor algorithm in this context. Also, we show that HOSRDA+LDA outperforms almost all the published methods on this dataset in terms of character detection accuracy.

The rest of the paper is organized as follows: In Section 2 we will go through the background needed for our work, which includes a brief review on LDA and SRDA. Section 3 gives an overview on tensor definitions and HODA. The mathematical formulation of the proposed method is presented in Section 4. The results and discussion come on Section 5. Eventually, the last section is the conclusion.

2. Background

2.1. Notations

In this paper higher order tensors are shown by calligraphic letters (e.g. $\mathcal{X}$ is a tensor), matrices are denoted by boldface capital letters (e.g. $\boldsymbol{X}$ is a matrix), boldface lower-case letters are used to denote vectors (e.g. $\boldsymbol{x}$ is a vector), and normal letters show scalars (e.g. x or X are scalar).

Ω is the set of all indices of training data samples (e.g. if we have K training data, $\Omega = \{1, \cdots, K\}$). We denote the set of indices of data in the cth class with Ω_c and its size by K_c.

2.2. Linear discriminant analysis (LDA)

LDA is one of the most well-known and common techniques for vector feature reduction. Let $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K\}$ be the set of K data points ($\boldsymbol{x}_i \in \mathbb{R}^m$) from C classes. LDA aims to find a linear transformation $\boldsymbol{A}$, which maps the data points to an l-dimensional space ($l < m$), where an adopted class separability criterion is optimized [46]. In this regard, the map of each point $\boldsymbol{x}_i$ is $\boldsymbol{y}_i = \boldsymbol{A}^T\boldsymbol{x}_i$. This linear mapping is obtained by maximizing the Fisher criterion as follows:

$$\boldsymbol{A} = \underset{\boldsymbol{A}}{argmax} \frac{tr\{\boldsymbol{A}^T\boldsymbol{S}_b\boldsymbol{A}\}}{tr\{\boldsymbol{A}^T\boldsymbol{S}_w\boldsymbol{A}\}} \tag{1}$$

In the above optimization problem, $\boldsymbol{S}_b$ and $\boldsymbol{S}_w$ are between-class and within-class scatter matrices respectively that are computed as follows:

$$\boldsymbol{S}_b = \sum_{c=1}^{C} K_c \left(\boldsymbol{\mu}^{(c)} - \boldsymbol{\mu}\right)\left(\boldsymbol{\mu}^{(c)} - \boldsymbol{\mu}\right)^T \tag{2}$$

$$\boldsymbol{S}_w = \sum_{c=1}^{C} \left(\sum_{k=1}^{K_c} \left(\boldsymbol{x}_k^{(c)} - \boldsymbol{\mu}^{(c)}\right)\left(\boldsymbol{x}_k^{(c)} - \boldsymbol{\mu}^{(c)}\right)^T \right) \tag{3}$$

where $\boldsymbol{\mu}^{(c)}$ and K_c are the mean and the size of the cth class respectively. $\boldsymbol{\mu}$ is the mean of all data regardless of their class.

By defining the total scatter matrix as $\boldsymbol{S}_t = \sum_{k=1}^{K} (\boldsymbol{x}_k - \boldsymbol{\mu})(\boldsymbol{x}_k - \boldsymbol{\mu})^T = \boldsymbol{S}_b + \boldsymbol{S}_w$, the optimization problem in (1) can be replaced by:

$$\boldsymbol{A} = \underset{\boldsymbol{A}}{argmax} \frac{tr\{\boldsymbol{A}^T\boldsymbol{S}_b\boldsymbol{A}\}}{tr\{\boldsymbol{A}^T\boldsymbol{S}_t\boldsymbol{A}\}} \tag{4}$$

whose solution is obtained by solving the GEVD problem $\boldsymbol{S}_b\boldsymbol{a} = \lambda\boldsymbol{S}_t\boldsymbol{a}$ and finding l leading eigenvectors as columns of $\boldsymbol{A}$. Since $\boldsymbol{S}_b$ is of rank $C-1$, $l \leq C-1$ should hold [46].

An issue in LDA is singularity of $\boldsymbol{S}_t$. Another concern for LDA is the computation of applying eigendecomposition in the case of high number of features. To overcome these concerns, SRDA has been introduced to solve the problem of LDA with a new formulation [45].

2.3. Spectral regression discriminant analysis (SRDA)

In [45], Cai et al. introduced SRDA, which casts the LDA technique into a regression problem. SRDA needs only to solve a (regularized) regression problem and omits the need for eigendecomposition in discriminant analysis. In the following, we briefly summarize SRDA.

Having the same definitions as in previous subsection, if for each $k = 1, \cdots, K_C$ and $c = 1, \cdots, C$ we put $\bar{\boldsymbol{x}}_k^{(c)} = \boldsymbol{x}_k - \boldsymbol{\mu}$ and $\bar{\boldsymbol{X}}^{(c)} = \left[\bar{\boldsymbol{x}}_1^{(c)}, \bar{\boldsymbol{x}}_2^{(c)}, \cdots, \bar{\boldsymbol{x}}_{K_c}^{(c)}\right]$, it can be shown that:

$$\boldsymbol{S}_t = \bar{\boldsymbol{X}}\bar{\boldsymbol{X}}^T, \quad \boldsymbol{S}_b = \bar{\boldsymbol{X}}\boldsymbol{W}\bar{\boldsymbol{X}}^T \tag{5}$$

where $\bar{\boldsymbol{X}} = \left[\bar{\boldsymbol{X}}^{(1)}, \cdots, \bar{\boldsymbol{X}}^{(C)}\right]$ and $\boldsymbol{W} = blockdiag\left(\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(C)}\right)$. $\boldsymbol{W}^{(c)}$ is a $K_c \times K_c$ matrix with all elements of $1/K_c$.

By replacing $\boldsymbol{S}_b$ and $\boldsymbol{S}_t$ from Eq. (5) in GEVD problem $\boldsymbol{S}_b\boldsymbol{a} = \lambda\boldsymbol{S}_t\boldsymbol{a}$ we have:

$$\bar{\boldsymbol{X}}\boldsymbol{W}\bar{\boldsymbol{X}}^T\boldsymbol{a} = \lambda\bar{\boldsymbol{X}}\bar{\boldsymbol{X}}^T\boldsymbol{a} \tag{6}$$

It is proved in [45] that if $\boldsymbol{y}$ is the eigenvector of $\boldsymbol{W}$ with eigenvalue λ (i.e. $\boldsymbol{W}\boldsymbol{y} = \lambda\boldsymbol{y}$) and also if $\bar{\boldsymbol{X}}^T\boldsymbol{a} = \boldsymbol{y}$ holds, then $\boldsymbol{a}$ is an eigenvector of problem in (6) with eigenvalue λ. Therefore, to solve the eigenproblem of LDA, we should find the eigenvectors of $\boldsymbol{W}$ and then solve a set of linear equations. It can be shown that finding the eigenvectors of $\boldsymbol{W}$ does not need eigendecomposition and they can be directly found from the Eq. (7):

$$\boldsymbol{y}_c = \left[\underbrace{0, \cdots, 0}_{\sum_{i=1}^{c-1} K_i}, \underbrace{1, \cdots, 1}_{K_c}, \underbrace{0, \cdots, 0}_{\sum_{i=c+1}^{C} K_i}\right]^T, \qquad c = 1, \cdots, C \tag{7}$$

where K_c is the size of class c.

3. Discriminant analysis for higher order data

3.1. Tensor operations and notations

Our notations in tensor algebra are very similar to those of [2] and [7]. Suppose that $\mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is an N-th order tensors. The mode-n matricization (unfolding) of $\mathcal{Y}$ is shown by $\boldsymbol{Y}_{(n)} \in \mathbb{R}^{I_n \times (\prod_{i=1, i\neq n}^{N} I_i)}$ and the mode-n product of $\mathcal{Y}$ and $\boldsymbol{A} \in \mathbb{R}^{J_n \times I_n}$ is denoted by $\mathcal{Z} = \mathcal{Y} \times_n \boldsymbol{A} \in \mathbb{R}^{I_1 \times \cdots \times J_n \times \cdots \times I_N}$ and $\boldsymbol{Z}_{(n)} = \boldsymbol{A}\boldsymbol{Y}_{(n)}$. Additionally, $\mathcal{Z} = \mathcal{Y} \times_1 \boldsymbol{A}^{(1)} \times_2 \cdots \times_N \boldsymbol{A}^{(N)}$ and $\mathcal{Z} = \mathcal{Y} \times_1 \boldsymbol{A}^{(1)} \times_2 \cdots \times_{n-1} \boldsymbol{A}^{(n-1)} \times_{n+1} \boldsymbol{A}^{(n+1)} \times_{n+2} \cdots \times_N \boldsymbol{A}^{(N)}$ are written in summarized form as $\mathcal{Z} = \mathcal{Y} \times \{\boldsymbol{A}\}$ and $\mathcal{Z} = \mathcal{Y} \times_{-n} \{\boldsymbol{A}\}$ respectively. In addition, it is defined $\boldsymbol{Z} = \langle \mathcal{Y}, \mathcal{Y} \rangle_{-n} = \boldsymbol{Y}_{(n)} \boldsymbol{Y}_{(n)}^T \in \mathbb{R}^{I_n \times I_n}$.

To the knowledge of the authors, the Tucker decomposition is the most exploited tensor decomposition method in discriminant analysis for higher order data. It decomposes a tensor into a core tensor multiplied by a matrix along each mode [44]. Tucker decomposition of rank-$(J_1, \cdots, J_N)$ (for $J_n \leq I_n$) is formulated as:

$$\mathcal{Y} \approx \mathcal{G} \times \{\boldsymbol{U}\} = \mathcal{G} \times_1 \boldsymbol{U}^{(1)} \times_2 \cdots \times_N \boldsymbol{U}^{(N)} \tag{8}$$

where $\boldsymbol{U}^{(n)} \in \mathbb{R}^{I_n \times J_n}$ are orthogonal factor matrices (or basis factors) of the decomposition. The core tensor $\mathcal{G} \in \mathbb{R}^{J_1 \times \cdots J_N}$ can be served as a compressed version of $\mathcal{Y}$. The core tensor can be obtained from tensor $\mathcal{Y}$ as below:

$$\mathcal{G} = \mathcal{Y} \times \{\boldsymbol{U}^T\} = \mathcal{Y} \times_1 \boldsymbol{U}^{(1)^T} \times_2 \cdots \times_N \boldsymbol{U}^{(N)^T} \tag{9}$$

3.2. Problem formulation: classification of tensor data

Suppose that we have the training dataset $\{\mathcal{X}^{(k)} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N},\ k = 1, \cdots, K\}$ along with their labels $\{c_k,\ k = 1, \cdots, K\}$ from C classes. We shall find a subspace where the separability of the classes is maximized, so that the labels of testing data can be predicted with minimum error.

The general model of high dimensional classification exploiting Tucker decomposition [7] aims to find the subspace spanned by $\{\boldsymbol{U}^{(n)} \in \mathbb{R}^{I_n \times J_n}, n = 1, \cdots, N\}$, in which the data samples $\mathcal{X}^{(k)}$ are represented as $\mathcal{G}^{(k)} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ in such a way that the data of different classes are better separated. As defined, a sample $\mathcal{X}^{(k)}$ and its projection $\mathcal{G}^{(k)}$ are related as follows:

$$\mathcal{X}^{(k)} = \mathcal{G}^{(k)} \times \{\boldsymbol{U}\} \tag{10}$$

It can be shown [7] that if we concatenate all data tensors along the mode-(N+1) in the tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times K}$ (i.e. $\mathcal{X} = cat\left(N+1, \mathcal{X}^{(1)}, \cdots, \mathcal{X}^{(K)}\right)$), we have:

$$\mathcal{X} = \mathcal{G} \times_{-(N+1)} \{\boldsymbol{U}\} = \mathcal{G} \times_1 \boldsymbol{U}^{(1)} \times_2 \cdots \times_N \boldsymbol{U}^{(N)} \tag{11}$$

where the projections of training data are concatenated in the mode-(N+1) of $\mathcal{G}$.

Eq. (11) represents the *Tucker-N decomposition* of the (N+1)th order tensor $\mathcal{X}$ (it decomposes $\mathcal{X}$ to a core tensor and N factor matrices.).

In fact, if we obtain the subspace basis factors via optimizing a discriminant cost function, then we can say that each core tensor $\mathcal{G}^{(k)}$ contains the discriminant features of $\mathcal{X}^{(k)}$ in the subspace spanned by $\{\boldsymbol{U}^{(n)} \in \mathbb{R}^{I_n \times J_n}, n = 1, \cdots, N\}$.

When the basis factors are found, each test data can be projected onto this subspace by Eq. (9) and after vectorizing the obtained tensor features (the projections), they can be passed to a classifier trained by the projections of training data. In this study we have used LDA classifier.

3.3. Higher order discriminant analysis (HODA)

HODA [7] is a tensor extension of LDA, introduced in 2005 by Yan, et al., which is first called DATER [8]. HODA obtains the basis factors $\boldsymbol{U}^{(n)}$ of Eq. (11) via maximizing the Fisher ratio between the core tensors $\mathcal{G}^{(k)}$ defined as follows [7]:

$$\varphi = \underset{\boldsymbol{U}^{(1)}, \cdots, \boldsymbol{U}^{(N)}}{argmax} \frac{\sum_{c=1}^{C} K_c ||\bar{\mathcal{G}}^{(c)} - \bar{\bar{\mathcal{G}}}||_F^2}{\sum_{k=1}^{K} ||\mathcal{G}^{(k)} - \bar{\mathcal{G}}^{(c_k)}||_F^2} \quad s.t. \quad \boldsymbol{U}^{(n)T}\boldsymbol{U}^{(n)} = \boldsymbol{I}, \quad n = 1, \cdots, N \tag{12}$$

where $\bar{\bar{\mathcal{G}}} = \frac{1}{K}\sum_{k=1}^{K} \mathcal{G}^{(k)}$ is the mean of all the training feature tensors and $\bar{\mathcal{G}}^{(c)} = \frac{1}{K_c}\sum_{k \in \Omega_c} \mathcal{G}^{(k)}$ is the mean of the feature tensors of the training data in class c. Also $\bar{\mathcal{G}}^{(c_k)}$ is the mean of the feature tensors of the data in the same class with the kth training sample.

Since the optimization problem in (12) cannot be solved directly, alternating solution is used. In this approach the cost function is optimized for each $\boldsymbol{U}^{(n)}$, while it is assumed that all other basis factors $\boldsymbol{U}^{(1)}, \cdots, \boldsymbol{U}^{(n-1)}, \boldsymbol{U}^{(n+1)}, \cdots \boldsymbol{U}^{(N)}$ are fixed. In [7] it is shown that the learning rule for a fixed n can be found via solving the below optimization problem:

$$\boldsymbol{U}^{(n)} = \underset{\boldsymbol{U}^{(n)}}{argmax} \frac{tr[\boldsymbol{U}^{(n)^T}\boldsymbol{S}_b^{-n}\boldsymbol{U}^{(n)}]}{tr[\boldsymbol{U}^{(n)^T}\boldsymbol{S}_w^{-n}\boldsymbol{U}^{(n)}]}, \quad s.t. \quad \boldsymbol{U}^{(n)T}\boldsymbol{U}^{(n)} = \boldsymbol{I} \tag{13}$$

In (13) the between-class scatter matrix $\boldsymbol{S}_b^{-n}$ is defined as follows:

$$\boldsymbol{S}_b^{-n} = \sum_{c=1}^{C} \left\langle \check{\mathcal{Z}}^{(c)-n}, \check{\mathcal{Z}}^{(c)-n} \right\rangle_{-n} = \left\langle \check{\mathcal{Z}}^{-n}, \check{\mathcal{Z}}^{-n} \right\rangle_{-n} \tag{14}$$

where

$$\check{\mathcal{X}}^{(c)} = \sqrt{K_c}\left(\bar{\mathcal{X}}^{(c)} - \bar{\bar{\mathcal{X}}}\right), \quad \check{\mathcal{X}} = cat\left(N+1, \check{\mathcal{X}}^{(1)}, \cdots, \check{\mathcal{X}}^{(C)}\right) \tag{15}$$

$$\check{\mathcal{Z}}^{(c)-n} = \check{\mathcal{X}}^{(c)} \times_{-n} \{\boldsymbol{U}^T\}, \quad \check{\mathcal{Z}}^{-n} = \check{\mathcal{X}} \times_{-(n,N+1)} \{\boldsymbol{U}^T\} \tag{16}$$

and

$$\bar{\bar{\mathcal{X}}} = \frac{1}{K}\sum_{k=1}^{K} \mathcal{X}^{(k)} \tag{17}$$

$$\bar{\mathcal{X}}^{(c)} = \frac{1}{K_c}\sum_{k \in \Omega_c} \mathcal{X}^{(k)}, \quad c = 1, \cdots, C \tag{18}$$

The within-class scatter matrix $\boldsymbol{S}_w^{-n}$ in (13) is defined as follows:

$$\boldsymbol{S}_w^{-n} = \sum_{k=1}^{K} \left\langle \tilde{\mathcal{Z}}^{(k)-n}, \tilde{\mathcal{Z}}^{(k)-n} \right\rangle_{-n} = \left\langle \tilde{\mathcal{Z}}^{-n}, \tilde{\mathcal{Z}}^{-n} \right\rangle_{-n} \tag{19}$$

where

$$\tilde{\mathcal{X}}^{(k)} = \mathcal{X}^{(k)} - \bar{\mathcal{X}}^{(c_k)}, \quad \tilde{\mathcal{X}} = cat\left(N+1, \tilde{\mathcal{X}}^{(1)}, \cdots, \tilde{\mathcal{X}}^{(K)}\right) \tag{20}$$

$$\tilde{\mathcal{Z}}^{(k)-n} = \tilde{\mathcal{X}}^{(k)} \times_{-n} \{\boldsymbol{U}^T\}, \quad \tilde{\mathcal{Z}}^{-n} = \tilde{\mathcal{X}} \times_{-(n,N+1)} \{\boldsymbol{U}^T\} \tag{21}$$

In above equations, c_k is the class of $\mathcal{X}^{(k)}$ and $\bar{\mathcal{X}}^{(c_k)}$ is the mean of the data in class c_k.

The problem of Eq. (13) can be solved via a GEVD problem, i.e. columns of $\boldsymbol{U}^{(n)}$ can be found as the J_n leading left eigenvectors of problem $\boldsymbol{S}_b^{-n}\boldsymbol{u} = \lambda \boldsymbol{S}_w^{-n}\boldsymbol{u}$. Since $\boldsymbol{S}_w$ can be very ill-conditioned in the early updates [7], some sort of regularization is needed, e.g. substituting $\boldsymbol{S}_w$ by $\boldsymbol{S}_w + \alpha \boldsymbol{I}$, where $\alpha \geq 0$ and $\boldsymbol{I}$ is the identity matrix.

In [41], Zhang et al. introduced Spatial-Temporal Discriminant Analysis (STDA), which is composed of a feature reduction stage followed by an LDA classifier. The formulation of feature reduction stage of STDA is the same as HODA for N=2 and $J_1 = J_2$. They used the below stop criterion for the iterative algorithm:

$$Error = ||\boldsymbol{U}^{(n)}(i) - \boldsymbol{U}^{(n)}(i-1)|| < \varepsilon, \quad n = 1, 2 \tag{22}$$

where $\boldsymbol{U}^{(n)}(i)$ is the estimated factor matrix of mode-n in the iteration number i.

3.3.1. Computational complexity of HODA

For assessing the computational cost of HODA algorithm, we count the number of *flops* that the algorithm requires. Each flop consists of one addition, subtraction, multiplication, or division [47]. Since HODA is an iterative algorithm, we first compute the number of flops for each iteration. The HODA algorithm needs to compute the tensors in Eqs. (14)–(21) and solve a GEVD problem. Assume $I = max(I_1, I_2)$. Eqs. (15), (17), (18), (20) need $O(KI^2)$ flops, while (16) and (21) require $O(LKI^2)$ flops, in which L is the number of iterations needed for the algorithm to converge. Computing the scatter matrices of Eqs. (14) and (19) also require $O(LKI^2)$ flops. Additionally, solving the GEVD problem requires $O(LI^3)$. Therefore, HODA requires $O(LKI^2 + LI^3)$ flops.

4. Higher order spectral regression discriminant analysis (HOSRDA)

To investigate HOSRDA formulation, we need to reformulate the scatter matrices of HODA. In this regard, we rewrite the Eq. (14) and find a new formulation for between-class scatter matrix of tensor data as follows:

$$\boldsymbol{S}_b^{-n} = \sum_{c=1}^{C} \left\langle \check{\mathcal{Z}}^{(c)}_{-n}, \check{\mathcal{Z}}^{(c)}_{-n} \right\rangle_{-n} = \sum_{c=1}^{C} \frac{1}{K_c} \left\langle \sum_{i\in\Omega_c} \mathcal{H}_i^{(c)-n}, \sum_{i\in\Omega_c} \mathcal{H}_i^{(c)-n} \right\rangle_{-n} \tag{23}$$

where for $i \in \Omega_c$, $\mathcal{H}_i^{(c)-n} = \left(\mathcal{X}^{(i)} - \bar{\bar{\mathcal{X}}}\right) \times_{-n} \{\boldsymbol{U}^T\}$. Therefore we have:

$$\boldsymbol{S}_b^{-n} = \sum_{c=1}^{C} \frac{1}{K_c} \left(\sum_{i\in\Omega_c} \boldsymbol{H}^{(c)-n}_{i_{(n)}} \right) \left(\sum_{j\in\Omega_c} \boldsymbol{H}^{(c)-n}_{j_{(n)}} \right) = \sum_{c=1}^{C} \boldsymbol{H}^{(c)-n}_{(n)} \boldsymbol{W}^{(c)} \boldsymbol{H}^{(c)^T-n}_{(n)} \tag{24}$$

where $\mathcal{H}^{(c)-n} = cat\left(N+1, \mathcal{H}_1^{(c)-n}, \cdots, \mathcal{H}_{K_c}^{(c)-n}\right)$ and $\boldsymbol{W}^{(c)}$ is a $KP_n \times KP_n$ block matrix whose all blocks are $P_n \times P_n$ identity matrices with $P_n = \prod_{\substack{m=1 \\ m\neq n}}^{N} J_m$ as follows:

$$\boldsymbol{W}^{(c)} = \frac{1}{K_c} \begin{bmatrix} \boldsymbol{I}_{P_n} & \boldsymbol{I}_{P_n} & \cdots & \boldsymbol{I}_{P_n} \\ \boldsymbol{I}_{P_n} & \boldsymbol{I}_{P_n} & \cdots & \boldsymbol{I}_{P_n} \\ \cdots & \cdots & \ddots & \vdots \\ \boldsymbol{I}_{P_n} & \boldsymbol{I}_{P_n} & \cdots & \boldsymbol{I}_{P_n} \end{bmatrix} \tag{25}$$

Now if we define $\mathcal{H}^{-n} = cat\left(N+1, \mathcal{H}^{(1)}_{-n}, \cdots, \mathcal{H}^{(C)}_{-n}\right)$ and $\boldsymbol{W} = blockdiag\left(\boldsymbol{W}^{(1)}, \ldots, \boldsymbol{W}^{(C)}\right)$, the between-class scatter matrix can be formulated as:

$$\boldsymbol{S}_b^{-n} = \boldsymbol{H}^{-n}_{(n)} \boldsymbol{W} \boldsymbol{H}^{-n^T}_{(n)} \tag{26}$$

In the same way, it can be shown that the within-class scatter matrix can be written as:

$$\boldsymbol{S}_w^{-n} = \boldsymbol{H}^{-n}_{(n)} \boldsymbol{L} \boldsymbol{H}^{-n^T}_{(n)} \tag{27}$$

where $\boldsymbol{L} = \boldsymbol{I}_{KP_n} - \boldsymbol{W}$ ($\boldsymbol{I}_{KP_n}$ is a $KP_n \times KP_n$ identity matrix). Thus, we can define

$$\boldsymbol{S}_t^{-n} = \boldsymbol{H}^{-n}_{(n)} \boldsymbol{H}^{-n^T}_{(n)} = \boldsymbol{S}_w^{-n} + \boldsymbol{S}_b^{-n} \tag{28}$$

Now, instead of solving the optimization problem of (13), we can solve the problem below:

$$\boldsymbol{U}^{(n)} = \underset{U^{(n)}}{argmax} \frac{tr[\boldsymbol{U}^{(n)^T} \boldsymbol{S}_b^{-n} \boldsymbol{U}^{(n)}]}{tr[\boldsymbol{U}^{(n)^T} \boldsymbol{S}_t^{-n} \boldsymbol{U}^{(n)}]}, \quad s.t. \quad \boldsymbol{U}^{(n)^T} \boldsymbol{U}^{(n)} = \boldsymbol{I} \tag{29}$$

The solution of above optimization problem is a matrix whose columns are J_n leading eigenvectors of the below GEVD problem:

$$\boldsymbol{S}_b^{-n} \boldsymbol{u}^{(n)} = \mu \boldsymbol{S}_t^{-n} \boldsymbol{u}^{(n)} \tag{30}$$

With the same approach as discussed in Section 2.3, we can claim that if $\boldsymbol{y}$ is an eigenvector of $\boldsymbol{W}$ with eigenvalue λ and also we have $\boldsymbol{H}^{-n^T}_{(n)} \boldsymbol{u}^{(n)} = \boldsymbol{y}$, then $\boldsymbol{u}^{(n)}$ is a solution of the GEVD problem (30) with $\mu = \lambda$. Thus to find the columns of factor matrix $\boldsymbol{U}^{(n)}$, rather than solving a GEVD problem, we can find the J_n leading eigenvectors of $\boldsymbol{W}$ and put them in the columns of matrix $\boldsymbol{Y}$ and then solve the linear system of equations $\boldsymbol{H}^{-n^T}_{(n)} \boldsymbol{U}^{(n)} = \boldsymbol{Y}$ for $\boldsymbol{U}^{(n)}$.

We show that the eigenvectors of $\boldsymbol{W}$ can be obtained analytically without eigendecomposition. Since $\boldsymbol{W}$ is a block-diagonal matrix, its eigenvalues and eigenvectors can be obtained from the eigenvectors and eigenvalues of its blocks. Therefore, we first seek for the eigenvectors/eigenvalues of $\boldsymbol{W}^{(c)}$ for an arbitrary c.

Suppose that $\boldsymbol{y}^{(c)}$ is an eigenvector of $\boldsymbol{W}^{(c)}$ corresponding to eigenvalue λ (i.e. $\boldsymbol{W}^{(c)} \boldsymbol{y}^{(c)} = \lambda \boldsymbol{y}^{(c)}$). If we break $\boldsymbol{y}^{(c)}$ to blocks of vectors of length P_n as $\boldsymbol{y}^{(c)} = \left[\boldsymbol{y}_1^{(c)^T}, \cdots, \boldsymbol{y}_{K_c}^{(c)^T} \right]^T$, then we have:

$$\begin{aligned} \boldsymbol{W}^{(c)} \boldsymbol{y}^{(c)} &= \frac{1}{K_c} \begin{bmatrix} \boldsymbol{I}_{P_n} & \boldsymbol{I}_{P_n} & \cdots & \boldsymbol{I}_{P_n} \\ \boldsymbol{I}_{P_n} & \boldsymbol{I}_{P_n} & \cdots & \boldsymbol{I}_{P_n} \\ \cdots & \cdots & \ddots & \vdots \\ \boldsymbol{I}_{P_n} & \boldsymbol{I}_{P_n} & \cdots & \boldsymbol{I}_{P_n} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}_1^{(c)} \\ \boldsymbol{y}_2^{(c)} \\ \vdots \\ \boldsymbol{y}_{K_c}^{(c)} \end{bmatrix} \\ &= \frac{1}{K_c} \begin{bmatrix} \sum_{k=1}^{K_c} \boldsymbol{y}_k^{(c)} \\ \sum_{k=1}^{K_c} \boldsymbol{y}_k^{(c)} \\ \vdots \\ \sum_{k=1}^{K_c} \boldsymbol{y}_k^{(c)} \end{bmatrix} = \lambda \begin{bmatrix} \boldsymbol{y}_1^{(c)} \\ \boldsymbol{y}_2^{(c)} \\ \vdots \\ \boldsymbol{y}_{K_c}^{(c)} \end{bmatrix} \end{aligned} \tag{31}$$

Thus, for each $p = 1, \cdots, K_c$ we have $\boldsymbol{y}_p^{(c)} = \frac{1}{\lambda K_c} \sum_{k=1}^{K_c} \boldsymbol{y}_k^{(c)}$. If we define $\boldsymbol{a} \triangleq \boldsymbol{y}_p^{(c)}, \forall p$, it can be easily concluded that $\boldsymbol{a} \triangleq \boldsymbol{y}_p^{(c)} = \frac{1}{\lambda K_c} K_c \boldsymbol{a} = \frac{\boldsymbol{a}}{\lambda}$ and therefore $\lambda = 1$.

For an arbitrary c, it is clear that the rank of $\boldsymbol{W}^{(c)}$ is P_n. Thus, $\boldsymbol{W}^{(c)}$ has P_n eigenvalues of value *one* with eigenvectors having the form of $\boldsymbol{y}^{(c)} = \underbrace{\left[\boldsymbol{a}_{P_n}^T, \cdots, \boldsymbol{a}_{P_n}^T \right]}_{K_c}^T$, where $\boldsymbol{a}_{P_n}$ is an arbitrary vector of length P_n.

To find the eigenvectors of $\boldsymbol{W}$, it should be noted that the eigenvalues of a block-diagonal matrix is the union of eigenvalues of its blocks and the eigenvectors can be constructed from the eigenvectors of the blocks. Since one is the eigenvalue of all the blocks of $\boldsymbol{W}$, for any set $\{\boldsymbol{y}^{(1)}, \cdots, \boldsymbol{y}^{(C)}\}$ of eigenvectors of blocks of $\boldsymbol{W}$, an eigenvector of $\boldsymbol{W}$ can be defined with the form $\boldsymbol{y} \triangleq \left[\boldsymbol{y}^{(1)^T}, \cdots, \boldsymbol{y}^{(C)^T} \right]^T$, corresponding to eigenvalue one. Consequently, we have:

$$\begin{aligned} \boldsymbol{W}\boldsymbol{y} &= \begin{bmatrix} \boldsymbol{W}^{(1)} & 0 & \cdots & 0 \\ 0 & \boldsymbol{W}^{(2)} & \cdots & 0 \\ \cdots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{W}^{(C)} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}^{(1)} \\ \boldsymbol{y}^{(2)} \\ \vdots \\ \boldsymbol{y}^{(C)} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{W}^{(1)} \boldsymbol{y}^{(1)} \\ \boldsymbol{W}^{(2)} \boldsymbol{y}^{(2)} \\ \vdots \\ \boldsymbol{W}^{(C)} \boldsymbol{y}^{(C)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{y}^{(1)} \\ \boldsymbol{y}^{(2)} \\ \vdots \\ \boldsymbol{y}^{(C)} \end{bmatrix} = \boldsymbol{y} \end{aligned} \tag{32}$$

Now that we have shown $\boldsymbol{W}$ has CP_n eigenvectors (corresponding to eigenvalue one), which can be constructed using random vectors, the factor matrices of Tucker decomposition can be obtained through solving a linear system of equations (i.e. $\boldsymbol{H}^{-n^T}_{(n)} \boldsymbol{U}^{(n)} = \boldsymbol{Y}$ for $\boldsymbol{U}^{(n)}$). Fig. 1 depicts a block diagram of classification a framework using HOSRDA. Algorithm 1 is the pseudocode of proposed HOSRDA. In this pseudocode function *Gram-Schmidt* applies a gram-schmidt orthogonalization process to the columns of its input ma-

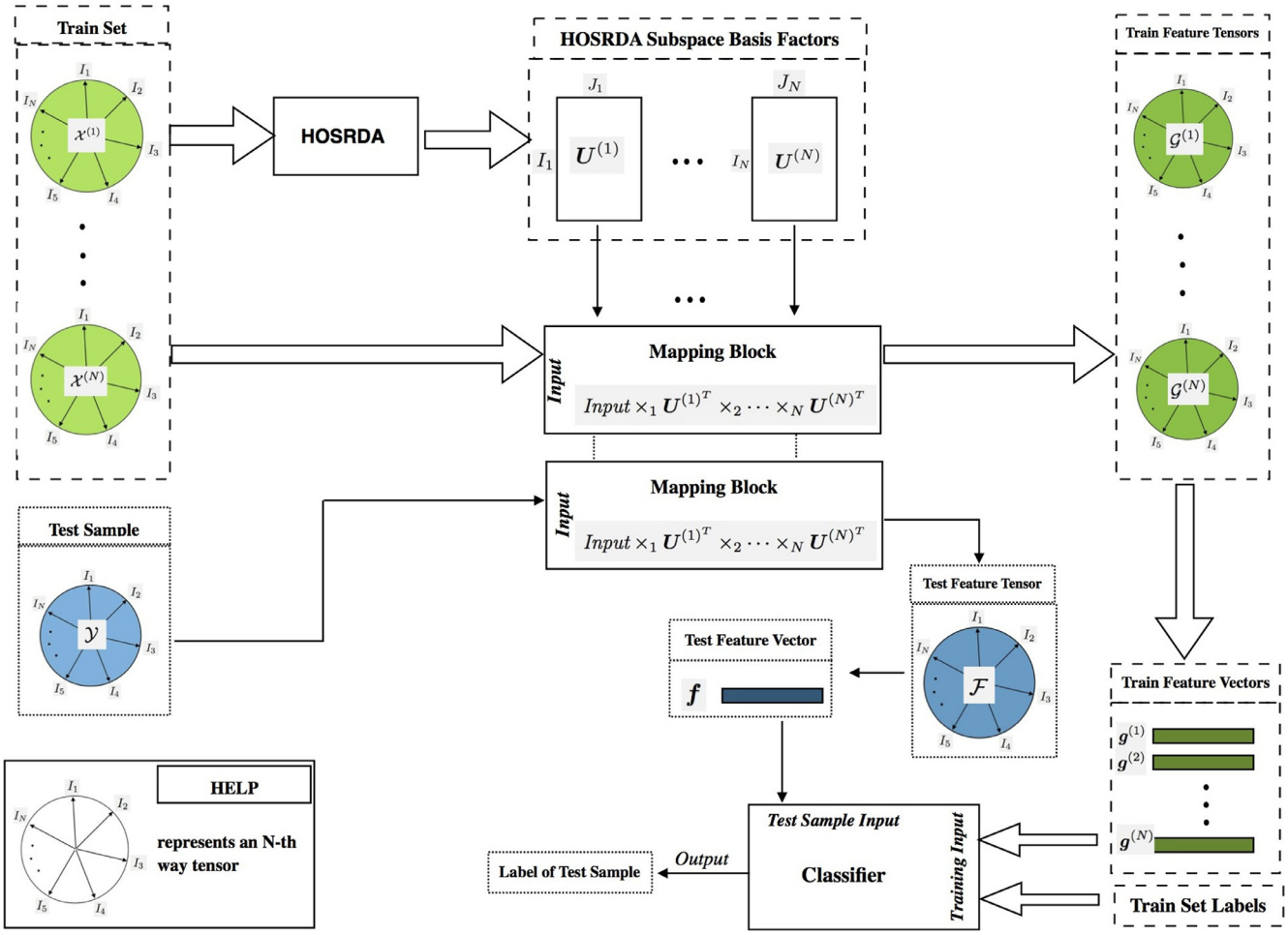

Fig. 1. The block diagram of classification framework with HOSRDA as the feature reduction stage.

trix. The two functions *rand* and *repmat* work the same as the MATLAB© functions with the same names. $rand(P_n, J_n)$ produces a $P_n \times J_n$ matrix of uniformly distributed random numbers between 0 and 1 and $repmat(\boldsymbol{A}, 1, m) = [\underbrace{\boldsymbol{A}, \cdots, \boldsymbol{A}}_{m}]$. Also the following tensors in the pseudocode are defined as:

$$\hat{\mathcal{X}} = cat\left(N+1, \hat{\mathcal{X}}^{(1)}, \cdots, \hat{\mathcal{X}}^{(C)}\right)$$
$$\hat{\mathcal{X}}^{(c)} = cat\left(N+1, \hat{\mathcal{X}}^{(c)}_{(1)}, \cdots, \hat{\mathcal{X}}^{(c)}_{(K_c)}\right) \tag{33}$$
$$\hat{\mathcal{X}}^{(c)}_{(k)} = \mathcal{X}^{(k)} - \bar{\bar{\mathcal{X}}}, \quad k \in \Omega_c$$

Note that since $\boldsymbol{W}$ is from rank CP_n, HOSRDA forces the upper band of CP_n to J_n, i.e. $J_n \leq CP_n$. Also, it's worth discussing the initialization of factor matrices. In most of algorithms based on Tucker decomposition, the initialization of factor matrices is done via HOSVD algorithm [44]. In HOSVD, nth factor matrix $\boldsymbol{U}^{(n)}$ is computed as the J_n leading left singular vectors of $\boldsymbol{X}_{(n)}$ (where $\boldsymbol{X}_{(n)}$ is the mode-n unfolding of tensor $\mathcal{X}$).

4.1. Computational complexity of HOSRDA

With the same assumptions as in Section 3.3.1, we compute the number of flops per iteration for HOSRDA. The computational load of HOSRDA is related to lines 2, 3, 6, 12, and 13 of Algorithm 1. Lines 2 and 3 require $O(3KI_1I_2)$ flops. For the remaining computations, referring to the for loop of line 5 and assuming $m = 3 - n$, lines 6, 12, and 13 require $2(I_m - 1)I_nKJ_m$, $2J_mKI_n(I_n + 1)$, and $2I_nJ_n^2$ flops. Note that for solving the linear regression problem in line 12, QR solver is used. Thus, the cost of HOSRDA is simplified as $O(LKI^2 + L(J_1^2 + J_2^2)I)$, where $I = max(I_1, I_2)$ and L is the number of iterations of the loop in line 4 of Algorithm 1.

5. Results and discussion

In this study, Tensor Toolbox [48,49], by Bader and Kolda, is used for tensor operations in MATLAB codes.

5.1. Dataset

Our dataset is data of a P300 speller from BCI Competition III-Dataset II, provided by Wadsworth center, Albany, NY, USA.[1] The experiment is recorded according to the paradigm of Donchin et al. in [50], originally by [51].

In this paradigm a 6 × 6 table of alphanumeric characters is presented to the subject. While the rows and columns of the table are successively and randomly intensified at a rate of 5.7 Hz, the subject is supposed to focus on a specific character, which is prescribed by the investigator. Among all 36 intensifications, one particular row and one particular column contain the target character, causing the recorded responses to these two intensifications be different from the columns/rows that do not contain the desired

[1] http://www.bbci.de/competition/iii/.

Algorithm 1: Higher Order Spectral Regression Discriminant Analysis (HOSRDA).

Input : K training data $\{\mathcal{X}_k \in \mathbb{R}^{I_1 \times \dots \times I_N}, k = 1, \cdots, K\}$ along with their labels $\{c_k, \quad k = 1, \cdots, K\}$, J_n, n=1,... ,N
Output: $\boldsymbol{U}^{(n)}$: N orthogonal basis factors $I_n \times J_n$, n=1,... ,N.

Initialize $\boldsymbol{U}^{(n)}, \quad n = 1, \cdots, N$.
$\bar{\mathcal{X}} = \frac{1}{K}\sum_{i=1}^{K} \mathcal{X}_i$.
Calculate $\hat{\mathcal{X}}$ according to equations (33).
repeat
  for $n = 1$ **to** N **do**
    $\mathcal{H}^{-n} = \hat{\mathcal{X}} \times_{-(n,N+1)} \{\boldsymbol{U}^T\}$.
    $P_n = \prod_{\substack{m=1 \\ m \neq n}}^{N} J_m$.
    for $c = 1$ **to** C **do**
      $\boldsymbol{Y}^{(c)} = repmat(rand(P_n, J_n), K_c, 1)$.
    end
    $\boldsymbol{Y} = \left[\boldsymbol{Y}^{(1)}; \cdots; \boldsymbol{Y}^{(C)}\right]$.
    Solve $\boldsymbol{H}_{(n)}^{-n^T}\boldsymbol{U}^{(n)} = \boldsymbol{Y}$ for $\boldsymbol{U}^{(n)}$.
    $\boldsymbol{U}^{(n)} = Gram - Schmidt\left(\boldsymbol{U}^{(n)}\right)$.
  end
until *Stop Criterion is met*;

character. We call the latter intensifications *target stimulations* and know that their responses contain a P300-like evoked potential [50], which is not present in the former intensifications (*non target stimulations*). Therefore a specific character can be determined through detecting the row and column with the largest P300 responses corresponding to the target stimulations. Each column/row intensifies 15 times. For each character we have $12 \times 15 = 180$ intensifications, from which $2 \times 15 = 30$ stimuli are target. For more information on the paradigm (e.g. inter-stimuli interval, duration of intensification, etc.) see [52].

The 64 channel EEG data were acquired at sampling frequency of 240 Hz and bandpass filtered in the band 0.1–60 Hz. The training set for each of the two subjects contains the EEG data of 85 characters and the test set includes 100 characters. The true characters of the test dataset is published with the results of competition in the competition website.[2]

For each channel, we have extracted the samples of 667 ms after each intensification onset as a trial. Each trial was bandpass filtered between 0.1 and 10 Hz with 8-order Chebyshev type I filter and then decimated to 20 Hz. In this way each trial consists of 14 samples (the same preprocessing has been done in [53]). Now the data should be tensorized: we have samples of K trials from 64 channels, each containing14 samples; so we can construct a third order tensor of size $64 \times 14 \times K$, where its modes are *Channel* $\times$ *Time* $\times$ *Trials*. This way of tensorizing EEG data including ERP has been used previously in [36,37]. For this kind of tensorization, each trial is a second order tensor (matrix) and all trials are concatenated in a third order tensor.

5.2. Selecting number of features

In this part we will provide our experimental analysis of finding best, smallest subspace dimension on data of P300 speller. To find the minimum dimension of HOSRDA subspace we assume $J \triangleq J_1 = J_2$ and select the smallest J between 1 and 5. For this purpose, we have used *Hold Out* cross validation method [46]. For each J, we select 45 random characters from the training set and call its data *hold out train set*, the data of remaining 40 characters build the cross validation test set (which we call *hold out test set*). Then we will run the Algorithm 1 with hold out train set and learn the HOSRDA subspace factor matrices, on which the hold out train and test set are mapped to get the discriminant features. Next, the accuracy of character detection in hold out test set is computed. For each J we repeat this procedure 10 times. Among the values for J with largest mean accuracy, the smallest one with smaller standard deviation, will be selected.The best values of J have been selected to be 3 for both subjects.

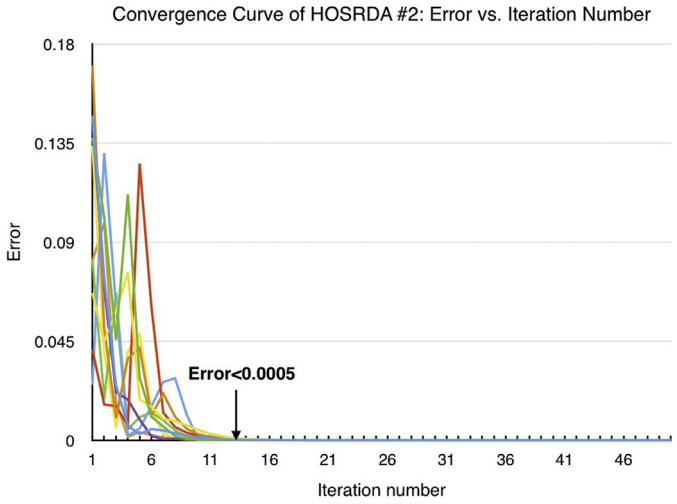

Fig. 2. The HOSRDA learning algorithm is executed for data of 10 randomly selected subsets of train set (data of 45 characters). Each line represents the convergence curve of one execution. It is apparently seen that the convergence trend does not depend on the specifications of the set used to train the learning stage.

5.3. Convergence and stability

Like all iterative algorithms, HOSRDA needs a stop criterion, which guarantees a stable convergence of the algorithm. Different stop criteria have been proposed in literature of tensor-based algorithms [41,44,54]

Since our algorithm maximizes the Fisher ratio, we propose the following stop criterion:

$$Error = |Fisher\ Ratio(i) - Fisher\ Ratio(i-1)| < \varepsilon \tag{34}$$

where $FisherRatio(i)$ is the Fisher ratio of training data that is projected on the HOSRDA subspace basis factors in the ith iteration. In other words, this criterion causes to break the iterative loop of learning when the class separability of training data projections does not change any more.

Inspired by [41], we have conducted an experiment for data of subject B of P300 speller dataset to check the convergence and stability of this stop criterion on our data: we executed the HOSRDA learning algorithm for trials of 10 randomly selected subsets of train set (each subset contains data of 45 randomly selected characters) and plotted the graph of Error vs. Iteration number (Fig. 2). According to this graph, we can experimentally claim that the convergence trend does not depend on the specifications of the training set. Note that in our next experiments we selected $\varepsilon = 0.0005$.

5.4. Character detection performance

For a P300 speller, the most important result to be reported is the character detection performance: It means how many characters of test data can be truly predicted and it is in direct relation

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

Table 1
The accuracy of character detection of different methods applied on two subjects of dataset II of BCI competition III, when R repetitions per intensification is used. Dashes indicate that the related reference has not reported any results in that field.

	SubjectA			*SubjectB*			*mean*			Subject
	5	10	15	5	10	15	5	10	15	R
HOSRDA+LDA	63	84	96	**82**	**94**	**97**	72.5	**89**	**96.5**	
eSVM [53]	**72**	83	**97**	75	91	96	**73.5**	87	**96.5**	
CNN-1 [55]	61	**86**	**97**	79	91	92	70	88.5	94.5	
MCNN-1 [55]	61	82	**97**	77	92	94	69	87	95.5	
EFLD [56]	65	82	93	78	93	**97**	71.5	87.5	95	
SRDA	54	**86**	95	71	89	95	62.5	87.5	95	
STDA	56	82	95	80	93	95	68	87.5	95	
HODA+LDA	49	79	92	74	93	96	61.5	86	94	
SWLDA [57]	–	–	–	–	–	–	–	–	92.5	
Reg.+HODA+LDA [40]	–	–	–	–	–	–	–	–	92	

with P300 detection performance. We have applied our proposed algorithm for character detection in dataset II of BCI competition III and has reached mean accuracy equal to 96.5%, 89%, and 72.5% for 15, 10, and 5 repetitions per intensification, respectively. This result is comparable to other reported results on this dataset in Table 1, where R is the number of repetitions per intensification (note that the first R repetition of each intensification is used from the dataset.). It can be seen that for subject B, HOSRDA outperforms all the previously reported methods, while for subject A, its results are fairly comparable with other methods.

The winner group of the competition [53] has reported 96.5% of accuracy (average for the two subjects). They have used Ensemble of SVM classifier along with sequential channel selection. Cecotti et al. [55] proposed seven classifiers based on convolutional neural network (CNN) with the best result of 95.5% for a multi-classifier CNN (MCNN). Salvaris et al. [56] used discrete-wavelet transform (DWT) for feature reduction and an Ensemble of Fishers Linear Discriminants (EFLD) for classification and achieved average accuracy of 95% for the two subjects. Krusienski et al. [57] introduced the use of Stepwise Linear Discriminant Analysis (SWLDA) and reported accuracy of 92.5% . Onishi et al. exploited HODA for feature reduction from tensor of polynomial approximations of each trial [40]; they reached an average accuracy of 92% on this dataset (in Table 1 this method is cited as Reg.+HODA+LDA, in which "Reg." stands for "Regression" and indicates that the data tensors are built by using regression approximations of each trial). Note that we have included the results reported in original papers in Table 1; for having more information about the parameters and structures of the methods, we refer the readers to the original references.

We have also compared our proposed method with STDA [41] and HODA [7], as the state-of-the-art tensor-based discriminant analysis methods used for data of P300 speller paradigm. STDA is implemented by authors, and HODA is implemented using NFEA toolbox [58], by imposing the assumption of $J_1 = J_2$. From Table 1 it is clear that HOSRDA+LDA outperforms STDA and HODA+LDA in character detection. As explained before, STDA is a special formulation of HODA. The differences in performance of STDA and HODA+LDA may be due to some differences in implementations or stop criteria used for them due to the fact that NFEA is not an open source toolbox.

Comparing the performance of SRDA and HOSRDA, we can observe that HOSRDA has a better performance.[3] However, It is well known that tensor-based discriminant analysis avoids the curse of dimensionality and does not suffer from Small Sample Size (SSS) problem [8,41] and therefore, the superiority of HOSRDA over SRDA becomes crystal clear when facing SSS problem, where small number of training samples are available. For comparing the performance of SRDA and HOSRDA in SSS problem, the character detection performances of HOSRDA and SRDA are computed when the number of training characters are varied from a small number to a rather large one, and the graphs in Fig. 3 are plotted. It can be seen that HOSRDA has a significantly better performance in facing SSS problem.

By comparing the proposed method with the other algorithms in terms of runtime complexity, the following results are obtained. The method of the winners (eSVM) have exploited advance ensemble of SVMs classification with channel selection [53], which has hours of training time. Additionally, from the Table 1 we can see that convolutional neural network's performance is very good. However, it has also a complex and high computational training procedure. Besides, the approach of tensor construction used in [40] is a time consuming process, while we construct our tensor from the raw data (bandpass filtered only). Generally, the tensor-based feature reduction techniques such as HOSRDA, HODA, and STDA require a very short training time to learn the subspace basis factors, for example, HOSRDA+LDA training stage takes 2.55 s in average for the two subjects in this P300 speller dataset. This can be regarded as a great advantage of HOSRDA+LDA over other complex methods, since it can extremely reduce the time of training stage of the BCI system. One may claim tensor techniques are complex tools; however, with a more precise glance at tensor algebraic operations, it will turn out to be simple concatenation of vectors and matrices and matrix multiplications.

5.5. Spatial projections of HOSRDA

An interesting point is the relation between spatial basis factors and the spatial properties of P300. Fig. 4(a) depicts the three spatial factor matrices of subject B. To evaluate how much these spatial projections are meaningful, we use [53] as a benchmark, where the importance of channels are marked on a scalp topography due to their ranking in a channel selection sequential procedure (Fig. 4(b)). Comparing Fig. 4(a) and (b), it is clear that HOSRDA has been successful in selecting the most important channels.

5.6. Complexity analysis

According to Sections 3.3.1 and 4.1, one can see in the case that the number of training samples is significantly large in comparison with the sizes of training data dimensions (i.e. I_1 and I_2), the computational complexity of HOSRDA and HODA are in the same order. However, if we are dealing with a small sample size problem, where the number of training samples are smaller than the

[3] Note that to implement SRDA in MATLAB the source codes provided by the first author of [45] on his website (http://www.cad.zju.edu.cn/home/dengcai/) are used.

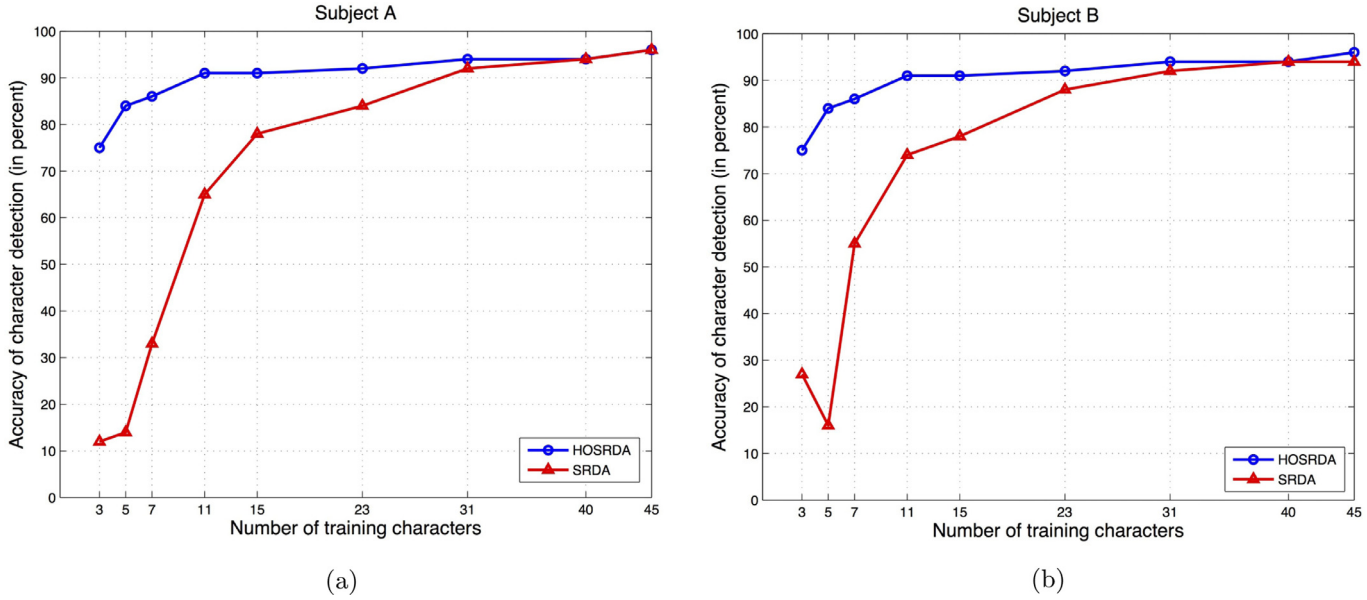

Fig. 3. Character detection performance of HOSRDA and SRDA for subject (a) A and (b) B in terms of number of training characters.

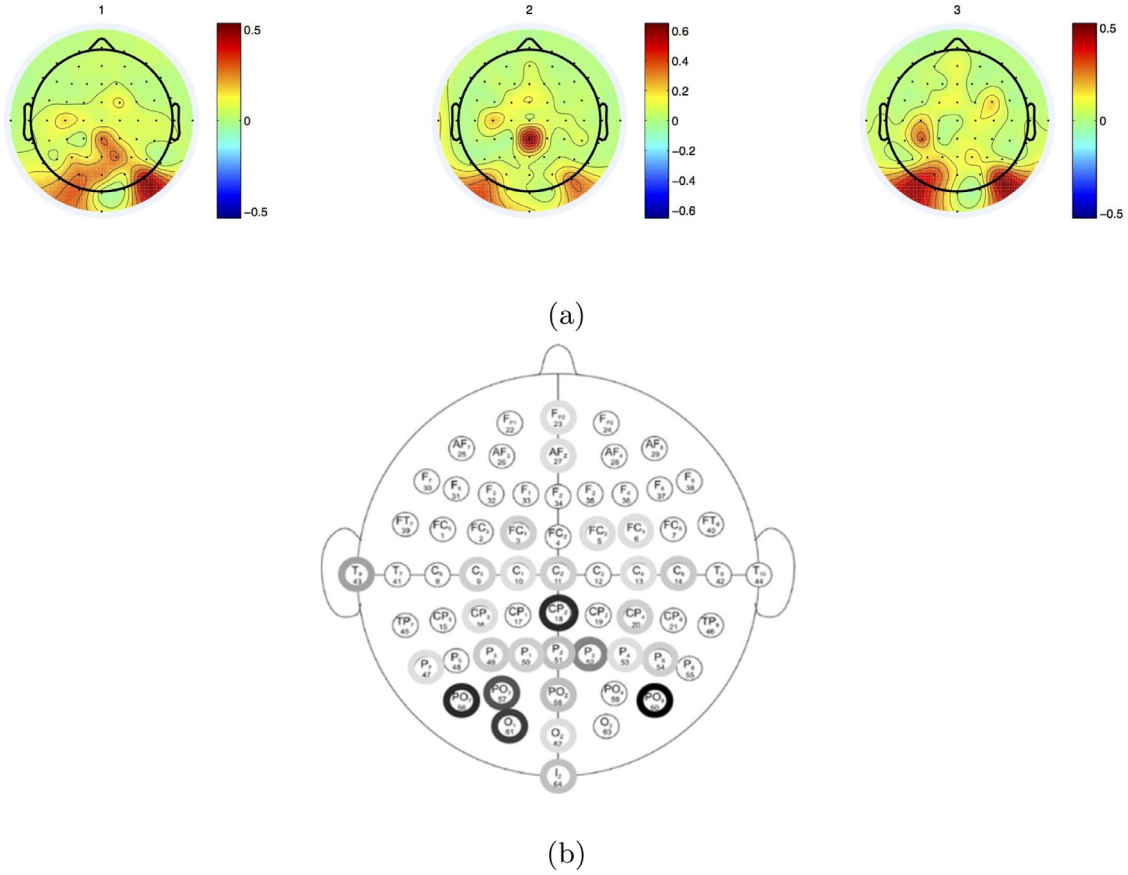

Fig. 4. (a) Spatial basis factors of HOSRDA for Subject B. (b) The results of channel ranking in [53] for Subject B. The darker the circle around a channel, the higher ranked the channel. (For interpretation of the colors of color bars, the readers are referred to the online version of the article.)

number of features, the computational complexity of HOSRDA will be considerably smaller due to the factor $O(LI^3)$ in complexity of HODA. Additionally, it is worth noting that if we exploit computationally low-cost, iterative algorithms for solving the least square problem (such as LSQR [59] as cited in [45]), it will need first order number of flops (instead of second order) in the L iterations of the iterative algorithm. This results in a notable drop in the computational complexity of HOSRDA even in the case when the number of training samples is large.

6. Conclusion

In this paper a new tensor-based feature reduction technique, called HOSRDA was proposed. HOSRDA is the higher order ex-

tension of SRDA and solves problem of eigendecomposition in HODA (higher order extension of LDA) via a regression problem. We exploited our proposed algorithm as a tool for feature reduction of P300 Speller data from BCI competition. HOSRDA+LDA reaches an accuracy of character detection of 96.5% (average for the two subject), which is better than almost all the published results on this dataset. We also compared the performance of HOSRDA+LDA with the state-of-the-art tensor-based discriminant analysis method used for classification in P300-based BCI, STDA, and showed that the proposed method outperforms it. The proposed methods' learning stage is in average 2.55 s for each subject, which is extremely less that other methods (for example eSVM needs hours of training). Since HOSRDA solves mainly a regression problem, further studies can be done to accompany regularization techniques (e.g. sparsity of factor matrices) with HOSRDA and its applications in other scopes of data analysis. Additionally, since the features extracted by HOSRDA are tensors, some tensor based classifiers may result in a better performance.

Acknowledgement

This work has been partly supported by Mowafaghian Grant from Djavad Mowafaghian Research Center of Intelligent Neuro-Rehabilitation Technologies, Sharif University of Technology, Tehran, Iran.

References

[1] A. Cichocki, et al., Tensor decompositions for signal processing applications from two-way to multiway component analysis, 2014 arXiv preprint:1403.4462.
[2] A. Cichocki, et al., Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation, John Wiley & Sons, 2009.
[3] H. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, A survey of multilinear subspace learning for tensor data, Pattern Recognit. 44 (7) (2011) 1540–1551.
[4] F. Nie, et al., Extracting the optimal dimensionality for local tensor discriminant analysis, Pattern Recognit. 42 (1) (2009) 105–114.
[5] W. Zhang, et al., Tensor linear laplacian discrimination (TLLD) for feature extraction, Pattern Recognit. 42 (9) (2009) 1941–1948.
[6] X. Li, et al., Discriminant locally linear embedding with high-order tensor data, Syst., Man, Cybern., Part B, IEEE Trans. 38 (2) (2008) 342–352.
[7] A. Phan, A. Cichocki, Tensor decompositions for feature extraction and classification of high dimensional datasets, Nonlinear Theory Appl., IEICE 1 (1) (2010) 37–68.
[8] S. Yan, et al., Discriminant analysis with tensor representation, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, IEEE, 2005, pp. 526–532.
[9] G. Lechuga, et al., Discriminant analysis for multiway data, Springer Proc. Math. Stat. (2015).
[10] Q. Li, D. Schonfeld, Multilinear discriminant analysis for higher-order tensor data classification, IEEE Trans. Pattern Anal. Mach. Intell. 36 (12) (2014) 2524–2537.
[11] D. Tao, et al., General tensor discriminant analysis and gabor features for gait recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1700–1715.
[12] J. Li, L. Zhang, Regularized tensor discriminant analysis for single trial EEG classification in bci, Pattern Recognit. Lett. 31 (7) (2010) 619–628.
[13] Z. Lai, et al., Sparse tensor discriminant analysis, IEEE Trans. Image Process. 22 (10) (2013) 3904–3915.
[14] G. Zhou, et al., Group component analysis for multiblock data: common and individual feature extraction, IEEE Trans. Neural Netw. Learn. Syst. 27 (11) (2016) 2426–2439.
[15] F. Cong, et al., Tensor decomposition of EEG signals: a brief review, J. Neurosci. Methods 248 (2015) 59–69.
[16] S. Hajipour Sardouie, et al., Canonical polyadic decomposition of complex-valued multi-way arrays based on simultaneous schur decomposition, in: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, IEEE, 2013, pp. 4178–4182.
[17] M. De Vos, et al., Canonical decomposition of ictal scalp eeg reliably detects the seizure onset zone, NeuroImage 37 (3) (2007) 844–854.
[18] E. Acar, et al., Multiway analysis of epilepsy tensors, Bioinformatics 23 (13) (2007) i10–i18.
[19] W. Deburchgraeve, et al., Neonatal seizure localization using parafac decomposition, Clinical Neurophysiol. 120 (10) (2009) 1787–1796.
[20] B. Hunyadi, et al., Block term decomposition for modelling epileptic seizures, EURASIP J. Adv. Signal Process. 2014 (1) (2014) 139.
[21] E. Pippa, et al., EEG-based classification of epileptic and non-epileptic events using multi-array decomposition, Int. J. Monitoring Surveillance Technol. Res. (2017).
[22] A. Cichocki, et al., Noninvasive BCIs: multiway signal-processing array decompositions., IEEE Comput. 41 (10) (2008) 34–42.
[23] H. Lee, et al., Nonnegative tensor factorization for continuous EEG classification, Int. J. Neural Syst. 17 (04) (2007) 305–317.
[24] J. Li, et al., A prior neurophysiologic knowledge free tensor-based scheme for single trial EEG classification, IEEE Trans. Neural Syst. Rehabil. Eng. 17 (2) (2009) 107–115.
[25] Y. Zhang, et al., Multiway canonical correlation analysis for frequency components recognition in SSVEP-based BCIs, in: International Conference on Neural Information Processing, Springer, 2011, pp. 287–295.
[26] Y. Zhang, et al., L1-regularized multiway canonical correlation analysis for SSVEP-based BCI, IEEE Trans. Neural Syst. Rehabil. Eng. 21 (6) (2013) 887–896.
[27] Y. Zhang, et al., Frequency recognition in SSVEP-based BCI using multiset canonical correlation analysis, Int. J. Neural Syst. 24 (04) (2014) 1450013.
[28] Y. Zhang, et al., Sparse Bayesian multiway canonical correlation analysis for EEG pattern recognition, Neurocomputing 225 (2017) 103–110.
[29] G. Zhou, et al., Linked component analysis from matrices to high-order tensors: applications to biomedical data, Proc. IEEE 104 (2) (2016) 310–331.
[30] J. Möcks, Decomposing event-related potentials: a new topographic components model, Biol. Psychol. 26 (1) (1988) 199–215.
[31] J. Moecks, Topographic components model for event-related potentials and some biophysical considerations, IEEE Trans. Biomed. Eng. 6 (35) (1988) 482–484.
[32] A.S. Field, D. Graupe, Topographic component (parallel factor) analysis of multichannel evoked potentials: practical issues in trilinear spatiotemporal decomposition, Brain Topogr. 3 (4) (1991) 407–423.
[33] M. Mørup, et al., ERPWAVELAB: a toolbox for multi-channel analysis of time–frequency transformed event related potentials, J. Neurosci. Methods 161 (2) (2007) 361–368.
[34] M. Mørup, et al., Shift-invariant multilinear decomposition of neuroimaging data, NeuroImage 42 (4) (2008) 1439–1450.
[35] F. Cong, et al., Benefits of multi-domain feature of mismatch negativity extracted by non-negative tensor factorization from EEG collected by low-density array, Int. J. Neural Syst. 22 (06) (2012) 1250025.
[36] K. Vanderperren, et al., Single trial ERP reading based on parallel factor analysis, Psychophysiology 50 (1) (2013) 97–110.
[37] M. Niknazar, et al., Blind source separation of underdetermined mixtures of event-related sources, Signal Process. 101 (2014) 52–64.
[38] F. Cong, et al., Multi-domain feature extraction for small event-related potentials through nonnegative multi-way array decomposition from low dense array eeg, Int. J. Neural Syst. 23 (02) (2013) 1350006.
[39] F. Cong, et al., Feature extraction by nonnegative tucker decomposition from EEG data including testing and training observations, in: Neural Information Processing, Springer, 2012, pp. 166–173.
[40] A. Onishi, et al., Tensor classification for P300-based brain computer interface, in: Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on, IEEE, 2012, pp. 581–584.
[41] Y. Zhang, et al., Spatial-temporal discriminant analysis for ERP-based brain–computer interface, Neural Syst. Rehabil. Eng., IEEE Trans. 21 (2) (2013) 233–243.
[42] G. Zhou, et al., Nonnegative matrix and tensor factorizations: an algorithmic perspective, IEEE Signal Process. Mag. 31 (3) (2014) 54–65.
[43] Y. Zhang, et al., Fast nonnegative tensor factorization based on accelerated proximal gradient and low-rank approximation, Neurocomputing 198 (2016) 148–154.
[44] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM Rev. 51 (3) (2009) 455–500.
[45] D. Cai, et al., SRDA: an efficient algorithm for large-scale discriminant analysis, Knowl. Data Eng., IEEE Trans. 20 (1) (2008) 1–12.
[46] S. Theodoridis, K. Koutroumbas, Pattern recognition, Academic Press, 2003.
[47] L.N. Trefethen, D. Bau, Numerical linear algebra, 50, Siam, 1997.
[48] B.W. Bader, et al., Matlab tensor toolbox version 2.6, 2015.
[49] B.W. Bader, T.G. Kolda, Algorithm 862: MATLAB tensor classes for fast algorithm prototyping, ACM Trans. Math. Software 32 (4) (2006) 635–653, doi:10.1145/1186785.1186794.
[50] E. Donchin, et al., The mental prosthesis: assessing the speed of a p300-based brain-computer interface, Rehabil. Eng., IEEE Trans. 8 (2) (2000) 174–179.
[51] L.A. Farwell, E. Donchin, Talking off the top of your head: toward a mental prosthesis utilizing event-related brain potentials, Electroencephalogr. Clin. Neurophysiol. 70 (6) (1988) 510–523.
[52] B. Blankertz, et al., The BCI competition III: validating alternative approaches to actual BCI problems, Neural Syst. Rehabil. Eng., IEEE Trans. 14 (2) (2006) 153–159.
[53] A. Rakotomamonjy, V. Guigue, BCI competition III: dataset II-ensemble of SVMs for BCI P300 speller, Biomed. Eng., IEEE Trans. 55 (3) (2008) 1147–1154.
[54] G. Zhou, A. Cichocki., Matlab toolbox for tensor decomposition and analysis version 1.1., 2013.
[55] H. Cecotti, A. Gräser, Convolutional neural networks for P300 detection with application to brain-computer interfaces, Pattern Anal. Mach. Intell., IEEE Trans. 33 (3) (2011) 433–445.
[56] M. Salvaris, F. Sepulveda, Wavelets and ensemble of flds for p300 classification,

in: Neural Engineering, 2009. NER'09. 4th International IEEE/EMBS Conference on, IEEE, 2009, pp. 339–342.
[57] D.J. Krusienski, et al., Toward enhanced P300 speller performance, J. Neurosci. Methods 167 (1) (2008) 15–21.
[58] A.H. Phan, Nfea: tensor toolbox extraction and applications, 2011. (Available online at http://www.bsp.brain.riken.jp/~phan/nfea/nfea.html).
[59] C.C. Paige, M.A. Saunders, LSQR: an algorithm for sparse linear equations and sparse least squares, ACM Trans. Math. Software 8 (1) (1982) 43–71.

Mina Jamshidi Idaji received her B.Sc. in electrical engineering and pure mathematics from Isfahan University of Technology, Isfahan, Iran in 2014, and her M.Sc. degree of biomedical engineering from Sharif University of Technology, Tehran, Iran in 2016. Her research interests are in signal and image processing, particularly with biomedical applications, brain-computer interface, multilinear (tensor) algebra, and graph algorithms.

Mohammad Bagher Shamsollahi received the B.Sc. degree in electrical engineering from Tehran University, Tehran, Iran, in 1988, the M.Sc. degree in electrical engineering, telecommunications, from the Sharif University of Technology, Tehran,Iran, in 1991, and the Ph.D. degree in electrical engineering, biomedical signal processing, from the University of Rennes 1, Rennes, France, in 1997. He is currently a Professor with the Department of Electrical Engineering, Sharif University of Technology. His research interests include biomedical signal processing, brain-computer interface, and time-scale and time-frequency signal processing.

Sepideh Hajipour Sardouie received the Ph.D. degree in Electrical Engineering from Sharif University of Technology, Iran, in co-tutelle with University of Rennes 1, France in 2014. She is currently an assistant professor with the Department of Electrical Engineering at Sharif University of Technology. Her research interests focus on biomedical signal processing, specially EEG signal processing.