Biomedical Signal Processing and Control 87 (2024) 105379

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

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

Low-rank Tensor Restoration for ERP extraction

Zahra Sohrabi Bonab [*], Mohammad B. Shamsollahi

BiSIPL, Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Keywords:
Event related potentials
Low-rank tensor decomposition
ADMM
Multi-mode analysis

ABSTRACT

Event-related potential (ERP) data is essentially multi-dimensional, with correlated data in some spaces. Therefore, matrix and vector analysis results in structural information loss. Tensor decomposition can be used to explore the shared structural information of ERP signal among related conditions. Perceiving the decaying trends of the singular value changes of unfolding matrices indicate that they are low-rank matrices. Based on this assumption, in this work, a low-rank tensor restoration (LTR) method is proposed. An operator splitting method known as the alternating direction method of multipliers (ADMM) is adapted to tackle the proposed optimization problem with orthogonality and sparsity constraints. Accordingly, the problem is solved in a sequential fashion by computing the unconstrained and orthogonality constrained quadratic sub-problems with closed-form solutions. The algorithm is examined under three application areas, namely, noise removal, feature extraction and subject-to-subject transfer learning. The empirical evaluations on real P300-based ERP dataset demonstrate the robustness and effectiveness of the proposed method.

1. Introduction

ELECTROENCEPHALOGRAM (EEG) is the most common non-inva sive and simple signal that is used in brain–computer interface (BCI) systems. BCI is an emerging technology that uses brain control signals to directly link the brain with the outside world without the involvement of muscles and peripheral nerves. Event-related potentials (ERPs) are time-locked changes of neuronal activity of brain in response to external stimuli. Among all EEG features, ERP-based BCIs are more popular and they have many practical applications in both clinical and non-clinical areas. While ERP detection is the core part of many BCI systems, it suffers from low signal to noise ratio and trial-to-trial variability [1]. Habituation, refractoriness, fatigue, boredom, or even attention level of the subject can affect the ERPs [2]. Therefore, averaging despite its signal-to-noise (SNR) improvement, implies a loss of information related to trial-to-trial variability.

In order to extract ERP components, it is usually modeled as sum of invariant signals and random noises. The signal part accounts for the information processing stages which can be divided into perceptual (e.g., N100), cognitive (e.g., P300), and decision-making (e.g., N400 or P650) components while the noise part accounts for the non-stationary background EEG, internal noises like muscle activity and Gaussian noise [3–5].

The presence of noise confines the processing precision in successive ERP-based applications. Hence, ERP extraction is an important preprocessing step for almost all usages. Until now, a large number of extraction techniques has been developed for different ERP signal processing applications. Filtering based methods are commonly used because of their simplicity and time effectiveness [6]. Using the idea that discrimination of spatial topology and temporal template could differentiate distinct tasks, several spatio-temporal filtering algorithms are proposed [5,7–13]. However, most of these methods are template-based and their performance is limited for variable components.

Optimization-based techniques also have been applied in EEG classification, where, ERP denoising is considered as an optimization problem consisting of regularization and data-fidelity. In [14], single-trial ERPs are classified using multiple sparse discriminant vectors learned from ℓ_1-regularized least-squares regressions. Zhang et al. [15] introduced a regularized sparse Bayesian method where a sparse discriminant vector is learned with a Laplace prior in a hierarchical fashion. In [16], three types of regularizers are presented that induce different types of sparsity on the input signal matrix. However, in [17] the optimization problem is solved under smoothness constraint.

In recent years, with the expansion of neural network research, many neural network based algorithms are suggested by researchers to detect ERP signals [18–21]. While these algorithms benefit from concurrent processing property and high accuracy, they suffer from computation cost and require higher hardware facilities [21]. Recently, increasing number of studies have used tensor-based techniques in many applications including signal and image processing [22–26] and pattern recognition [27–29]. In this paper an ERP is represented as

[*] Corresponding author.
E-mail address: zsohrabi@ee.sharif.edu (Z. Sohrabi Bonab).

https://doi.org/10.1016/j.bspc.2023.105379
Received 24 October 2022; Received in revised form 30 June 2023; Accepted 14 August 2023
Available online 30 August 2023

Table 1
Notations and definitions.

$\mathcal{X}, X, x$	Tensor, Matrix, Vector
$\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$	An Nth-order tensor
$X_{(n)} = \text{unfold}_n(\mathcal{X})$	n-mode matricization of tensor $\mathcal{X}$
$rank_d(\mathcal{X})$	Rank of $X_{(n)}$
$\times_i$	Multiplication across the n-mode
$\mathcal{G} \times \{A\}$	$\mathcal{G} \times_1 A_1 \times_2 A_2 \times_3 \ldots \times_N A_N$
$\mathcal{G} \times_{-n} \{A\}$	$\mathcal{G} \times_1 A_1 \times_2 \ldots \times_{n-1} A_{n-1} \times_{n+1} A_{n+1} \times_{n+2} \ldots \times_N A_N$
$\langle \mathcal{X}, \mathcal{Y} \rangle$	Inner product of two tensors
$\|\mathcal{X}\|_F = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$	The Frobenius norm of the tensor $\mathcal{X}$
$\|X\|_* = \sum_i^{\min(m_1,m_2)} \sigma_i(X)$	The Nuclear norm, which is sum of singular values of $X \in \mathbb{R}^{m_1 \times m_2}$
$\|\mathcal{G}\|_0 = \|\{i : g_i \neq 0\}\|$	ℓ_0-norm defined as total number of nonzero elements

a multi-dimensional signal. Therefore, multi-linear analysis can be carried out based on these tensor space.

The focus of the present paper is on the concepts and application of low rank approximation framework based on well known tensor decomposition method, known as Tucker decomposition (TD) [30]. Although TD generally suffers from non-uniqueness solution, it offers extra degrees of freedom when compared with the other tensor decomposition methods [23]. With our prior knowledge about each class (ie. semi-class trials similarity, time course sparsity and spatial correlation), we restrict ERP tensor to be low-rank in each mode and a sparse regularization is induced on the core tensor allowing it to well describe the detailed information in the data [31].

A practicable Alternating Direction Method of Multipliers (ADMM) algorithm [32] is used to solve the proposed Low-rank tensor restoration (LTR) model. Computationally efficient and closed-form expressions are derived for updating each variable. By imposing additional sparseness and orthogonality constraints it is possible to exploit the prior knowledge simultaneously, namely, the local self-similarity of trial mode, the sparse time course of temporal mode and the neighbor electrodes recording correlation of spatial mode [31,33,34]. These constraints lead to learn a structured factor matrices where they are used to transfer the new subject's raw data into feature space. In this work, this property is called subject-to-subject transfer learning (STL). The aim of STL is to learn an objective function for a target subject with help of not only the target domain but also from other subjects domain. Classification is one of the most investigated applications of STL, where, it utilizes the knowledge implied in the source domain(s) to improve the performance of the learned decision functions and features on the target domain [35–37]. Due to low computational cost and acceptable accuracy, the results obtained in this work are very interesting considering real-world BCI application.

The remainder of the paper is organized as follows: In Section 2 briefly overviews tensor algebra. Section 3 presents the LTR model and the corresponding ADMM-based algorithm to solve the low-rank tensor restoration as an optimization problem. Experiments were performed on two real dataset with P300 ERP component while focusing on three applications and the results are presented in Section 4. Section 5 concludes this paper. Basic notations and definitions are shown in Table 1.

2. Background

Two common models in tensor data analysis are Tucker decomposition and PARAFAC/CP decomposition. In this section, a brief review of Tucker decomposition is presented. A Tucker model with orthogonality constraints on component matrices is a generalization of SVD from matrix to tensor which decomposes a tensor into a core tensor multiplied by a matrix along each mode [23], and it is formulated as:

$$\mathcal{X} = \mathcal{G} \times_1 A_1 \times_2 A_2 \times_3 \ldots \times_N A_N + \mathcal{E} \tag{1}$$

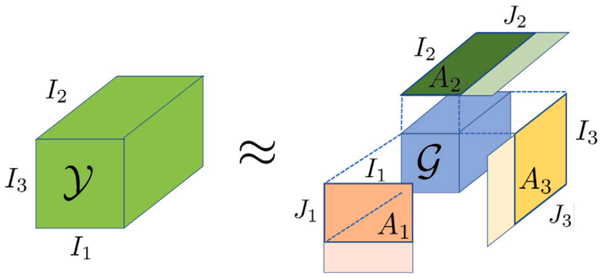


Fig. 1. Three dimensional tensor low-rank Tucker decomposition.

where $\mathcal{E} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is decomposition error and $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ is called the core tensor and it is similar to the singular values in SVD-based matrix decomposition with its intensity showing the level of interaction between different components. $A_i \in \mathbb{R}^{J_i \times I_i}$ is the column-wise orthonormal factor matrix and can be regarded as the principal components in each mode [33].

It is notable that, although Tucker decomposition suffers from core rotations, which results in non-unique solutions, it is preferred over other tensor analysis methods because of its flexibility in extracting a different number of components along each mode. Within this mathematical framework, every multidimensional data can be viewed as a whole data set which involves a joint processing along each mode.

The focus of this article is to solve the constrained Tucker decomposition of N dimensional tensor data such as ERP signals. An ERP tensor recorded from real experimental paradigms are always with an evident correlation along each of its modes. Meanwhile, the constructed tensor data often displays a low-rank structure due to significant correlations between neighboring electrodes and similar trials. Therefore with small number of latent factors, data variation and information are represent-able. Based on this assumption, the ERP data is modeled as a N-dimensional tensor $\mathcal{Y} = \mathcal{X} + \mathcal{N}$, where $\mathcal{Y}$ is a tensorized recorded EEG data and $\mathcal{X}$ is superposition of ERP components and it is a low-rank tensor. Moreover, $\mathcal{N}$ corresponds to the background EEG and additive noises. The least important tailing factor columns in each A_i and the proportional core slices in $\mathcal{G}$ along each dimension are discarded to construct a compressed approximation. For simplicity in Fig. 1 a three-dimensional tensor and its decomposed elements are illustrated.

To restore tensor from corrupted observations $\mathcal{Y}$ under low-rankness and sparsity constraint, the corresponding noise-free tensor $\mathcal{X}$ can be estimated by solving the following optimization problem:

$$\underset{\mathcal{X}}{\textbf{argmin}} \quad \frac{1}{2}\|\mathcal{Y} - \mathcal{X}\|_F^2 + \lambda \Phi(\mathcal{X}) \tag{2}$$

where $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the observation tensor and $\Phi(\mathcal{X})$ is the regularization term related to the rank of $\mathcal{X}$. $\Phi(\mathcal{X})$ identifies prior knowledge about the clean ERP tensor, and $\lambda > 0$ is a compromise parameter that controls trade-off between the two terms.

3. Proposed LTR model

In this section, first a general regularized minimization problem is solved and then its extension to ERP signal denoising and classification along with subject-to-subject transfer learning are introduced.

In order to incorporate structural information hidden in the signal, sparsity constrain is imposed on the core tensor along with low-rankness regularization over each factor matrix. To obtain this goal, both inner sparsity and subspace low-rankness are integrated into the problem. The regularization term is defined as: $\Phi(\mathcal{X}) = \|\mathcal{G}\|_0 + \alpha \prod_{d=1}^{N} rank_d(\mathcal{X})$, where $\mathcal{G}$ is TD core tensor, and $\|\mathcal{G}\|_0$ is ℓ_0-norm. The variable α is trade-off control parameter that determines the weight of each term.

Minimization of $\|\mathcal{G}\|_0$ is a nonconvex optimization problem and generally it is impossible to solve it when the signal dimension is large. A general alternative approach is to recast it to a linear problem as $\|\mathcal{G}\|_1 := \sum_i |g_i|$ corresponding to sum of all the core tensor elements.

Meanwhile, due to the non-convex and non-smoothness of the rank function $rank(\cdot)$, its calculation is usually NP-hard problem that cannot be solved within polynomial time. Therefore, the function $rank_d(\mathcal{X})$ is relaxed to one of its convex successors with a singular value penalty function $f(\cdot)$ as $rank_d(\mathcal{X}) := \min \sum_{i=1}^{\min\{m_1,m_2\}} f(\sigma_i(X_{(d)}))$, where $X_{(d)} \in \mathbb{R}^{m_1 \times m_2}$ denotes the mode-d unfolding matrix of the tensor $\mathcal{X}$, and $\sigma_i(X_{(d)})$ represents the ith singular value of $X_{(d)}$. With this definition, the penalty function $f(x) = x$ induces the nuclear norm. Nuclear norm is extensively used for matrix completion and rank relaxation. With these assumptions along with adding defined regularization term to the model in (2), the final model for ERP tensor restoration is given by:

$$\begin{aligned} \underset{\mathcal{G},A_1,\ldots,A_N}{\textbf{argmin}} \quad & \frac{1}{2}\|\mathcal{Y}-\mathcal{X}\|_F^2 + \lambda\|\mathcal{G}\|_1 + \beta \prod_{d=1}^{N} \sum_{i=1}^{M_d} \sigma_i(X_{(d)}) \\ \textbf{s.t.} \quad & \mathcal{X} = \mathcal{G} \times_1 A_1 \times_2 A_2 \times_3 \ldots \times_N A_N \\ & A_i^T A_i = I, \quad i = 1,\ldots,N \end{aligned} \tag{3}$$

where $\beta = \alpha\lambda$ with λ being the compromise parameter and M_d is the minimum dimension of mode-d unfolding matrix of $\mathcal{X}$. Additionally, the orthogonality constraint $A_i^T A_i = I$ is typically utilized to restrain the columns of A_i from becoming degenerate.

Note that problem (3) is difficult to solve due to the interdependent nuclear norms. Accordingly, we perform variable splitting and allocate separate auxiliary variables to each unfolding of $\mathcal{X}$. Let $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N$ be new tensor variables equal to the reconstructed tensor $\mathcal{X}$. In another word, the new variable $\mathcal{T}_d \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined such that $T_{d(d)} = X_{(d)}$ for all $d \in \{1,2,\ldots,N\}$ and equivalently problem (3) is reformulated as:

$$\begin{aligned} \underset{\mathcal{G},A_1,\ldots,A_N}{\textbf{argmin}} \quad & \frac{1}{2}\|\mathcal{Y} - \mathcal{G} \times_1 A_1 \times_2 A_2 \times_3 \ldots \times_N A_N\|_F^2 \\ & + \lambda\|\mathcal{G}\|_1 + \beta \prod_{d=1}^{N} \sum_{i=1}^{M_d} \sigma_i(T_{d(d)}) \\ \textbf{s.t.} \quad & \mathcal{T}_d = \mathcal{G} \times_1 A_1 \times_2 A_2 \times_3 \ldots \times_N A_N \\ & A_d^T A_d = I, \quad d = 1,\ldots,N \end{aligned} \tag{4}$$

With these new variables $\mathcal{T}_d$s, the semi-augmented Lagrangian function of problem (4) is given as follows:

$$\begin{aligned} & \mathcal{L}(\mathcal{G}, A_1,\ldots,A_N, \mathcal{T}_1,\ldots,\mathcal{T}_N, \Lambda_1,\ldots,\Lambda_N, \mu) \\ & := \frac{1}{2}\|\mathcal{Y} - \mathcal{G} \times_1 A_1 \times_2 A_2 \times_3 \ldots \times_N A_N\|_F^2 \\ & + \lambda\|\mathcal{G}\|_1 + \beta \prod_{d=1}^{N} \sum_{i=1}^{M_d} \sigma_i(T_{d(d)}) \\ & + \sum_{d=1}^{N} \langle \mathcal{T}_d - \mathcal{G} \times_1 A_1 \times_2 A_2 \times_3 \ldots \times_N A_N, \Lambda_d \rangle \\ & + \sum_{d=1}^{N} \frac{\mu}{2}\|\mathcal{T}_d - \mathcal{G} \times_1 A_1 \times_2 A_2 \times_3 \ldots \times_N A_N\|_F^2 \\ & \textbf{s.t.} \quad A_d^T A_d = I, \quad d = 1,\ldots,N \end{aligned} \tag{5}$$

where Λ_ds are Lagrange dual variables and $\mu > 0$ is the step size. $\mathcal{L}$ is computed subject to $A_d^T A_d = I, \quad d = 1,\ldots,N$.

Estimation of multiple unknowns simultaneously and directly, is hard. One idea to solve these kinds of problems is to decompose them into m smaller subproblems so that each of these decomposed subproblems only involves one variable and thus the properties of this variable could be used effectively in algorithmic design. ADMM is an approximation to augmented Lagrangian method (ALM) by sequentially updating each of the primal variables that make use of the advantages that the generated sub-problems could have closed-form solutions [38, 39]. Problem (5) can be decomposed into several subproblems for each variable,where at each iteration, the objective function is solved according to one of the constraints while other constraints are fixed. In consequence, the core tensor and the factor matrices are updated alternatively. The solution of the model (5) can be established by iteratively optimizing the corresponding subfunctions.

(1) **Update $\mathcal{G}$**: With other parameters fixed, the core tensor $\mathcal{G}$ can be updated using:

$$\begin{aligned} & \underset{\mathcal{G}}{\textbf{argmin}} \quad \mathcal{L}(\mathcal{G}, A_1,\ldots,A_N, \mathcal{T}_1,\ldots,\mathcal{T}_N, \Lambda_1,\ldots,\Lambda_N, \mu) = \\ & \underset{\mathcal{G}}{\textbf{argmin}} \quad \lambda\|\mathcal{G}\|_1 + \frac{1}{2}\|\mathcal{Y} - \mathcal{G} \times_1 A_1 \times_2 \ldots \times_N A_N\|_F^2 \\ & \qquad + \sum_{d=1}^{N} \langle \mathcal{T}_d - \mathcal{G} \times_1 A_1 \times_2 \ldots \times_N A_N, \Lambda_d \rangle \\ & \qquad + \sum_{d=1}^{N} \frac{\mu}{2}\|\mathcal{T}_d - \mathcal{G} \times_1 A_1 \times_2 \ldots \times_N A_N\|_F^2 \end{aligned} \tag{6}$$

By doing some simple algebra problem (6) turns to following optimization problem:

$$\underset{\mathcal{G}}{\textbf{argmin}} \quad a\|\mathcal{G}\|_1 + \frac{1}{2}\|\mathcal{G} - \mathcal{K} \times_1 A_1^T \times_2 \ldots \times_N A_N^T\|_F^2 \tag{7}$$

where $a = \frac{\lambda}{1+N\mu}$ and $\mathcal{K} = \frac{\mathcal{Y}+\sum_{d=1}^{N}(\mu\mathcal{T}_d+\Lambda_d)}{1+N\mu}$. If we set $\mathcal{P} = \mathcal{K} \times_1 A_1^T \times_2 \ldots \times_N A_N^T$, then the problem (7) has a closed form solution as follows [40]:

$$\hat{\mathcal{G}} = \text{sign}(\mathcal{P})\max(0, |\mathcal{P}| - a\rho). \tag{8}$$

where ρ is pre-defined thresholding parameter.

(2) **Update $\mathbf{A_i}$**: In order to update each factor matrix A_i, all other variables are fixed. Ignoring the constant terms, the sub-problem for updating A_i is re-arranged as:

$$\begin{aligned} \underset{A_i}{\textbf{argmin}} \quad & \frac{1}{2}\|\mathcal{G} \times_1 A_1 \times_2 \ldots \times_N A_N - \mathcal{K}\|_F^2 \\ \textbf{s.t.} \quad & A_i^T A_i = I \end{aligned} \tag{9}$$

whose optimal solution can be obtained via employing the fact that $\|\mathcal{G} \times_n A\|_F = \|\mathcal{G}\|_F, \forall A^T A = I$:

$$\begin{aligned} \underset{A_i}{\textbf{argmin}} \quad & \frac{1}{2}\|A_i G_{(i)} - B_i\|_F^2 \\ \textbf{s.t.} \quad & A_i^T A_i = I \end{aligned} \tag{10}$$

where $B_i = \text{unfold}_i(\mathcal{K} \times_{-i} \{A^T\})$. It is simple to show that the optimization problem (10) is equivalent to:

$$\begin{aligned} \underset{A_i}{\textbf{argmin}} \quad & \frac{1}{2}\|A_i - G_{(i)}^T B_i\|_F^2 \\ \textbf{s.t.} \quad & A_i^T A_i = I \end{aligned} \tag{11}$$

Therefore, following theorem 2.1 of [41] quadratic problem (10) has closed-form solution:

$$\hat{A}_i = U_i V_i^T \tag{12}$$

where U_i and V_i are two orthogonal matrices satisfying the SVD factorization $G_{(i)}^T B_i = U_i D_i V_i^T$.

(3) **Update $\mathcal{T}_\mathbf{i}$**: With other parameters fixed, each auxiliary tensor $\mathcal{T}_i$ is updated via:

$$\begin{aligned} \underset{\mathcal{T}_i}{\textbf{argmin}} \quad & \beta' \sum_{j=1}^{M_d} \sigma_j(T_{i(i)}) + Tr(\Lambda^T(\mathcal{T}_i - (\mathcal{G} \times \{A\}))) \\ & + \frac{\mu}{2} Tr((\mathcal{T}_i - (\mathcal{G} \times \{A\})^T(\mathcal{T}_i - (\mathcal{G} \times \{A\})) \end{aligned} \tag{13}$$

where $\beta' = \beta \prod_{d \neq i}^{N} \sum_{j=1}^{M_d} \sigma_j(T_{d(d)})$. Problem (13) rearranges into squared from as follows:

$$\underset{\mathcal{T}_i}{\textbf{argmin}} \quad \frac{\beta'}{\mu} \sum_{j=1}^{M_d} \sigma_j(T_{i(i)}) + \frac{1}{2}\|\mathcal{T}_i - (\mathcal{G} \times \{A\}) - \frac{\Lambda_i}{\mu}\|_F^2 \tag{14}$$

Problem (14) is known as singular value thresholding which is proximity operator associated with the nuclear norm, and it has a closed form solution as follows [42]:

$$\hat{\mathcal{T}}_{i(i)} = U_i(\max(\Sigma_i - \frac{\beta'}{\mu}I, 0))V_i^T \tag{15}$$

where U_i and V_i are the singular vectors of SVD decomposition of $(\text{unfold}_i(\mathcal{G} \times \{A\}) + \frac{\Lambda_i}{\mu})$ associated with singular values greater than $\frac{\beta'}{\mu}$. Note that since only singular values greater than $\frac{\beta'}{\mu}$ is needed, to speed up the calculations, with appropriate algorithms (e.g. Lanczos algorithm) only first few singular values and singular vectors could be computed [42,43].

(4) **Update $\mathbf{\Lambda_i}$**: Lagrange multipliers is updated using gradient ascent.

$$\Lambda_i^{t+1} = \Lambda_i^t + \mu^t(\mathcal{T}_i^{t+1} - \mathcal{G}^{t+1} \times \{A^{t+1}\}) \tag{16}$$

Algorithm 1 LTR algorithm via ADMM

```
Input: Recorded noisy ERP tensor $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$, stopping criterion $\epsilon$
Output: The restored ERP tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$
    Initialization: core tensor $\mathcal{G}^0$, factor matrices $A_i^0$, and auxiliary tensor $\mathcal{T}_i^0 = \mathcal{Y}$ for $i = 1, 2, \ldots, N$, $\mu^0 > 0$, $\mu^{max}$, $\delta = 1.1$, maximum number of iterations $T$.
while not converge or $t < T$ do
    Update core tensor $\mathcal{G}^{t+1}$ by (8)
    Update all factor matrices $A_i^{t+1}$ by (12)
    Update all auxiliary tensors $\mathcal{T}_i^{t+1}$ by (15)
    Update Lagrange variable $\Lambda_i^{t+1}$ by (16)
    Update step size $\mu^{t+1}$ by $\mu^{t+1} = \min(\delta\mu^t, \mu^{max})$
    Check the convergence condition: $\frac{\|\mathcal{X}^{t+1} - \mathcal{X}^t\|_F^2}{\|\mathcal{Y}\|_F^2} < \epsilon$
    $t = t + 1$
end while
```

Since subproblems arising in the $\mathcal{G}$-update and A_i-update and $\mathcal{T}_i$-update are solvable with closed form solution, i.e., there exist $\hat{\mathcal{G}}$ and $\hat{A}_i$(12) and $\hat{\mathcal{T}}_i$(8), after each iteration of ADMM the sum of the primal and dual optimality gaps decreases. According to [32], this results is sufficient decrease condition to minimize the augmented Lagrangian. Therefore, in practice the ADMM converges to a modest accuracy solution within a few tens of iterations.

3.1. LTR for feature extraction

In order to extract significant features to maximize classification performance, simultaneous tensor decompositions is required. This problem can be considered as a generalization of Joint Approximative Diagonalization (JAD) in matrix algebra [23]. Accordingly, all training tensors are concatenated into one $K+1$ order training data tensor, and the Tucker-K decomposition is performed [44].

Algorithm 2 summarizes the training procedure of proposed LTR method for feature extraction and online classification. At step 2, LTR seeks to optimize factor matrices A_n under the constraint $A_n^T A_n = I$. By the end of the training stage, with the learned factor matrices A_n, the lower-dimensional tensor subspace representation $\mathcal{G}_{i,j}$ of each $\mathcal{Y}_{i,j}$ belonging to class i is computed as $\mathcal{G}_{i,j} = \mathcal{Y}_{i,j} \prod_{k=1}^{K} \times_k A_k^T$. In the classification framework, when a recorded tensor $\mathcal{Y}^t$ is received, first its low-rank tensor subspace representation is computed via the factor matrices found for training data named as projection matrices. Then the extracted feature is fed into a desired classifier and it is compared with train features. This procedure is illustrated in Fig. 2.

Algorithm 2 Training Procedure of LTD for classification

```
Input: Training tensor $\mathcal{Y}_{i,j}|_{1 \le i \le c}^{1 \le j \le n_i} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_K \times N}$, corresponding class labels $i \in \{1, 2, \ldots, c\}$, total number of training tensors $N = \sum_{i=1}^{c} n_i$ and reduced tensor subspace $J_1 \times J_2 \times \ldots \times J_K$, ADMM parameters, and $\epsilon$.
Output: Projection matrices $A_n \in \mathbb{R}^{I_n \times J_n}$ constrained by $A_n^T A_n = I$ and the projected tensor $\mathcal{G}_{i,j}|_{1 \le i \le c}^{1 \le j \le n_i} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_K}$.
while not converge do
    Optimize projection matrices $\hat{A}_n$ by solving problem (3) with ADMM method.
    $\mathcal{X}^t = \mathcal{G} \times \{A^t\}$
    Check the convergence condition: $\frac{\|\mathcal{X}^{t-1} - \mathcal{X}^t\|_F^2}{\|\mathcal{Y}\|_F^2} < \epsilon$
    $t = t + 1$
end while
$\mathcal{G}_{i,j} = \mathcal{Y}_{i,j} \prod_{k=1}^{K} \times_k A_k^T$
```

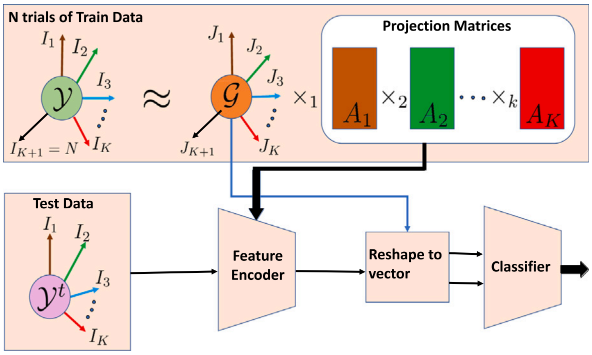


Fig. 2. Diagram of LTR model online classification procedure.

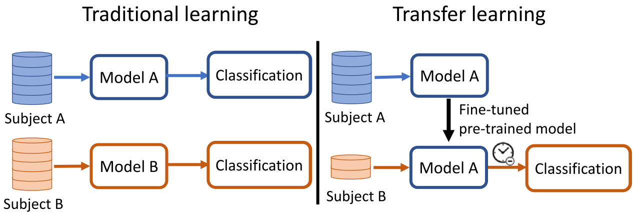


Fig. 3. Traditional vs. Transfer learning.

3.2. LTR for subject-to-subject transfer learning

In machine learning, TL is a method that utilizes data or knowledge from relevant situation to facilitate learning for a new desired problem. In the proposed work, subject to subject transfer learning (STL) is implemented in order to facilitate BCI applications. As depicted in Fig. 3, in this framework the required training time is remarkably reduced. Only a small amount of calibration data is required from a new subject. The work is based on the hypothesis that the similarity of brain dynamics in ERP paradigms among individuals is predictable. In order to validate the efficiency of STL model with the proposed LTR method, an experiment on real data is performed. The results suggest a practical way toward online ERP detection and can set a light to numerous real-world BCI applications.

When solving cross-subject classification problem via STL, a target subject classifier is designed by using the labeled data from the related subjects. A practical solution to this problem is to find the common latent features through feature transformation and use them as a bridge to transfer knowledge [35]. $\mathcal{Y}_L^T$ are labeled target subject's data from some calibration trials, and $\mathcal{Y}_U^T$ are unlabeled data to be recognized in subsequent trials.

In each iteration, LTR-STL has the following steps.

(1) Common Factor Matrix Extraction: For each source subject (denoted by S_k with $k = 1, \dots, m$), the source domain trials $\mathcal{Y}^{S_k}$ are separately input to the LTR algorithm to produce common latent feature extractor space $A_i^{S_k}$. Common latent factor A_i is obtained by averaging latent factors of all subjects $A_i = \text{mean}(A_i^{S_1}, \dots, A_i^{S_m})$.

(2) Specific Feature Extraction: For each target subject, the extracted common latent factor is fed to the target-specific feature extractor, which results in target feature space. Feature encoder function is defined as:

$$f(\mathcal{X}) = \mathcal{X}(\prod_{k=1}^{K} \times_k A_k^T) \tag{17}$$

(3) Data Classification: The feature extraction rule is subsequently applied to the test set. The output of the target specific feature extractor is input to the preferred classifier (e.g. LDA).

4. Experiments and results

The LTR method can effectively use the correlation between different modes and suppress the background noise. It may be widely applied to many types of EEG signal processing. This section discusses the application of proposed method in the detection of EEG signal components in one of widely used paradigms, named P300 speller.

4.1. Application I: Noise removal

In order to study noise removal capability of our method we used the real data with adding different levels of background Gaussian noise. The experimental data set provided in [45] was applied for the comparative analysis. Although our research purpose differs from the reference, the dataset nonetheless provides relevant experimental material for our research. The dataset is based on the P300 response and applied to both disabled and healthy subjects. It contains a total of 8 subjects' experimental data (Subject 1–4, 6–9). Due to reasons stated in [45], Subject 5 is not considered in this paper. It contains four recording sessions per subject, while each session includes six runs. In each session one of the images is set as target and the participants should count how many times this image is appeared in the screen. The group of six images randomly flashed is called one block. The number of blocks was chosen randomly between 20 and 25. In our experiment we use 15 blocks to train data for all subjects. Hence, the training data for one subject consisted of 1620 trials.

Before applying our method, the data was bandpass filtered between 1 Hz and 12 Hz by a 6th order Butterworth filter to attenuate large drifts and irrelevant high frequency noise. Then, the signal was further downsampled to 256 Hz from 2048 Hz to reduce the unnecessary dimensionality. For every single trial 1 s of EEG data after stimulus onest is extracted. Therefore, our 32-channel input tensor dimension is [32; 256; 1620]. A four-fold cross validation was applied to test the algorithm performance. One session's data is left for testing and the remaining three sessions are used for training. This procedure was repeated four times so each session served once for validation.

The optimal reduced dimension equal to [8, 8, 100] was chosen based on the results of the noise-free premier experiment, and applied to the online experiment. We preferred to find a common low-rank for all subjects to reduce the number of parameters, thus making the algorithm more general across subjects. However, it could be also possible to obtain a subject-specific optimal dimension based on the individual training dataset. The trained classifier was then applied for the testing session. The incoming data X_{new} was preprocessed and noise added in the same way as the training data. Since LDA is one of the most commonly used method for ERP signal analysis, a trained LDA is used to classify each segment into one of the two classes. It is notable that LDA was performed using the $fitcdiscr()$ function in MATLAB 2016b, with the '*DiscrimType*' option set to '*linear*' and the regularization parameter '*Gamma*' set to 0.1.

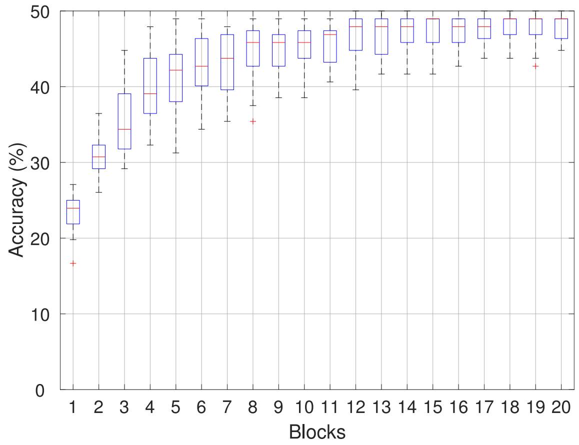


Fig. 4. Classification accuracies obtained by the LTR algorithm for SNR = −7 dB vs. different numbers of training data blocks.

The method performance is validated in different signal-to-noise ratios (SNR) defined as $20log(\frac{\sigma_x}{\sigma_n})$ [dB], where σ_x and σ_n are respectively the standard deviation of the signal and noise throughout the simulations. We considered the SNR values to be −15 and −7 dB. Fig. 5 shows the classification accuracies obtained by LDA and LTR-LDA, averaged over the generated twenty noisy signal tensors, for different numbers of training data blocks. For all subjects LTR-LDA yielded higher average accuracy than the solely LDA . As the SNR decreased, a more significant superiority was achieved by LTR-LDA method. Moreover, let us analyze the accuracy of the LTR method versus time, in terms of both mean and standard deviation of the error. In this study, the results are reported for the average statistics derived over the available subjects. As seen in Fig. 4, the plot depicts that the accuracy of the proposed LTR method increases with less standard deviation.

4.2. Application II: Feature extraction

In order to check feature extraction capability of our method, the publicly available P300 speller dataset from BCI Competition III Dataset II[1] is used. The dataset denoted A and B are from 2 healthy volunteers and they are recorded using the standard 10–20 EEG montage. The 64 channel data were acquired at sampling frequency of 240 Hz. Similar to [28], only 667 ms after each trial's stimulus onset is considered. Each extracted trial is bandpass filtered in the band 0.1 Hz to 12 Hz with 8th order Butterworth filter.

During the signal collection stage, the subject was seated in front of a computer screen, and a 6 × 6 character matrix was displayed as a stimulus. Out of 6 rows and 6 columns, one row and one column contain the desired character. The character matrix was shown for a 2.5 s period during each character spelling interval. The rows and columns of this matrix were intensified for 100 ms and remains blank for 75 ms. These 12 intensifications are referred as one trial. For a single character, 15 trials of data is collected. 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 available in the competition website[2]. The number of corresponding train and test epochs for each subject are $85 \times 12 \times 15 = 15\,300$ and $100 \times 12 \times 15 = 18\,000$, respectively. After this preprocessing stage, each

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

[2] Available online: http://www.bbci.de/competition/iii/

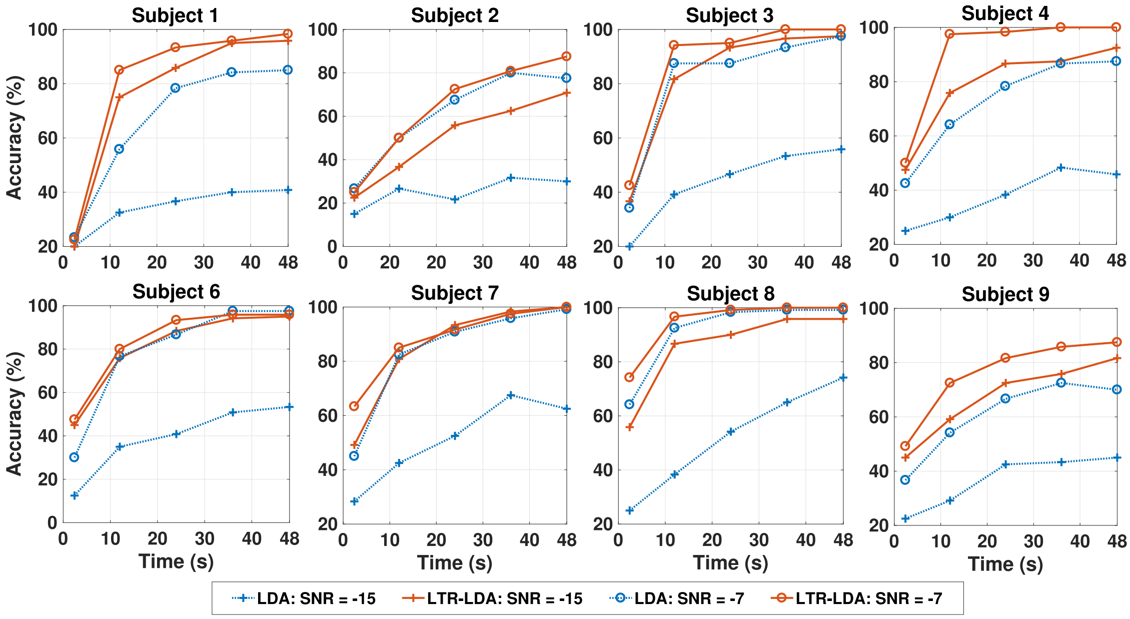


Fig. 5. Classification accuracies obtained by the LDA and LTR-LDA algorithms for all subjects of EPFL Dataset [45] with different additive noise level vs. training data time.

trial is assumed to be a second order tensor ($Channels \times Time$) and all trials are concatenated in the third mode. Therefore, the final tensor dimension is $64 \times 160 \times K$, where its modes are $Channel \times Time \times Trials$ with K indicating the number of training or testing trials.

4.2.1. Parameter setting

The whole optimization procedure for the proposed LTR model for tensor classification can be summarized as Algorithm 2. It is required to balance the trade-off between different variations in the augmented Lagrange function (5) by tuning the parameters. The investigation is made based on grid-search strategy and for each analysis, one parameter is changing while the others are fixed. We choose λ and β from the set $\{0.1, 0.2, \dots, 10.0\}$ and select initial μ from $\{0, 0.5, \dots, 10\}$. In our experiments, the initialized parameter μ is updated by $\mu^{t+1} = \delta\mu^{t}$ with $\delta = 1.2$ (where this value is chosen from the similar works), under the ADMM framework in an iteration processing. The parameter β controls the effect of factor matrices low rankness, while λ controls compressed core tensor's sparsity regularization. Through all the experiments, the parameter β shows little effect on both subjects, and empirically it is set to 1.8. Moreover, λ is set to 0.5 and $\mu_0 = 10$. The input signal is of dimension $[Channel; Time] = [64; 160]$. In our experiment based on cross validation the best reduced rank for Subject A is extracted to be $[33; 14]$, while for Subject B is $[37; 7]$.

4.2.2. Recognition accuracy:

The character recognition accuracy is shown in Table 2. Bold numbers indicate the highest accuracy of the proposed algorithm along the columns. The comparison methods include: higher order spectral regression discriminant analysis (HOSRDA) [28] representing state-of-the art for the tensor-based approach, SVNN [21], MsCNN [19], CapsNet [20], and CNN-1 [18] representing state-of-the-arts for neural network and deep learning based feature extraction approach. Ds-Reg [16] representing state-of-the-art for regularized method. Finally, eSVM [46] the winner group of the main competition. It can be seen that LTR has an acceptable performance in facing the classification problems.

Table 2
Character recognition of different methods applied on two subjects of dataset II of BCI competition III with different number of trials repetitions.

Method	Subject A				Subject B			
	15	10	5	1	15	10	5	1
LTR	**99**	**88**	55	**20**	**99**	**94**	72	35
HOSRDA [28]	96	86	63	17	97	**94**	**82**	**46**
SVNN [21]	98	88	71	17	96	**94**	76	31
MsCNN [19]	89	81	46	16	96	**94**	74	37
CapsNet [20]	98	87	68	16	96	**94**	81	45
CNN-1 [18]	97	86	61	16	92	91	79	35
DS-Reg [16]	**99**	86	71	17	94	90	79	35
eSVM [46]	97	83	**72**	16	96	91	75	35

4.3. Application III: Subject-to-subject transfer learning

One of BCI applications main challenge is obtaining sufficient training data and performing real-time processing. Transferring one subject's data model to another is a solution for this problem. Therefore, with minimized computational cost an acceptable accuracy is obtained. In subject-to-subject transfer framework, the parameters of the pre-trained LTR model are transferred and fine-tuned by the new subject's dataset. Hence, by learning the shared structure of datasets, the classification time is reduced for the new subject.

The efficiency of the proposed LTR-STL algorithm is examined by classification performance. BCI competition III dataset is used for this experiment. One of the subjects is chosen as LTR model train subject. The other subject is set as the test one. 40 characters of the test subject's training data is given to the algorithm to adapt the bias of the classifier. The feature extraction rule is subsequently applied to the test set, followed by a linear LDA classifier. In order to study the effect of fine-tuning a comparative study is done between pre-trained model without any fine-tuning train data from new subject and 10 characters and 45 characters fine-tuned model (see Fig. 6).

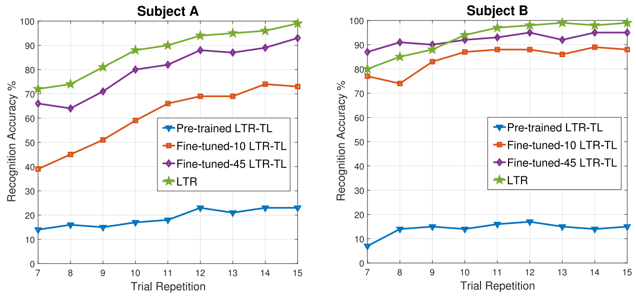
A comparison between character recognition performance of the proposed LTR method with MsCNN and HOSRDA algorithms and their STL version is shown in Table 3. MsCNN algorithm uses SVM classifier, while both LTD and HOSRDA utilize LDA classifier. The proposed LTR-STL technique achieves acceptable performance with limited training data and exponentially decreased classification time, which is depicted in Fig. 7.

5. Conclusion

Taking across-trial variability into account is important for physiological and clinical studies. In this paper, a tensor-based model,

Table 3
Quantitative comparison of different methods for traditional and transfer learning. Forty characters are selected to fine-tune the transfer learning method.

Method	Subject								Trials							
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
LTR	A	20	31	49	54	55	65	72	74	81	88	90	94	95	96	99
	B	35	54	54	66	72	80	80	85	88	94	97	98	99	98	99
	Mean	27.5	42.5	51.5	60	63.5	72.5	76	79.5	84.5	91	93.5	96	97	97	99
LTR-STL	A	23	31	50	53	56	62	72	72	78	80	85	84	87	92	93
	B	36	62	67	73	80	82	87	91	90	92	93	95	92	95	95
	Mean	26	45.5	54.5	60	68.5	72.5	76.5	78	80.5	85	87	91.5	89.5	92	94
MsCNN	A	16	16	39	38	46	49	65	69	78	81	82	87	88	89	89
	B	37	58	65	73	74	83	87	88	91	94	92	92	93	96	96
	Mean	26.5	37.0	52.0	55.5	60.0	66.0	76.0	78.5	84.5	87.5	87.0	89.5	90.5	92.5	92.5
MsCNN-STL	A	24	38	46	50	60	70	72	79	84	86	89	89	92	94	96
	B	40	59	67	74	79	84	90	92	94	97	96	98	97	97	96
	Mean	32.0	48.5	56.5	62.0	69.5	77.0	81.0	85.5	89.0	91.5	92.5	93.5	94.5	95.5	96.0
HOSRDA	A	17	33	50	60	63	70	70	75	83	84	88	91	92	94	96
	B	46	64	71	76	82	88	92	93	94	94	95	96	95	96	97
	Mean	31.5	48.5	60.5	68	72.5	79	81	84	88.5	89	91.5	93.5	93.5	95	96.5
HOSRDA-STL	A	10	14	23	26	31	37	41	40	44	52	54	60	61	64	66
	B	10	19	26	40	41	40	44	49	51	58	56	59	60	60	66
	Mean	10	16.5	24.5	33	36	38.5	42.5	44.5	47.5	56	55	59.5	60.5	62	66



(a) (b)

Fig. 6. Fine-tuning effect for subject A (a) and subject B (b)

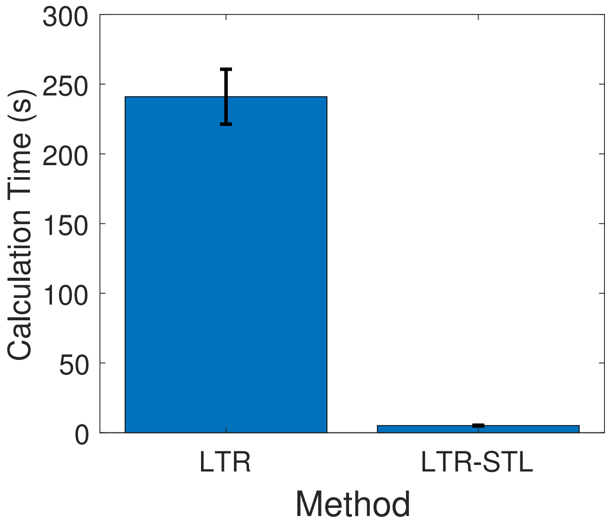


Fig. 7. LTR and LTR-STL Model Training time comparison.

named LTR, for ERP extraction from multichannel EEG is proposed. The multi-dimensional ERP signal was decomposed and redeemed with low-rank Tucker model constrained with sparsity regularization on its core tensor. The optimization problem is solved via ADMM algorithm with closed-form solutions.

It is shown that LTR is a reliable algorithm to identify and characterize ERP events. The experimental results showed that the proposed method can achieve noticeable SNR improvement and it boosts the classification accuracy. Given its increased detection performance, LTR lighten the road to a more efficient modeling of the ERP component extraction and detection, where it becomes possible to transfer one subject's model to train another one with much lower computation time. Therefore, it could provide new significant insights for BCI applications and explanation of brain mechanisms in many different conditions.

References

[1] W. Wu, C. Wu, S. Gao, B. Liu, Y. Li, X. Gao, Bayesian estimation of ERP components from multicondition and multichannel EEG, NeuroImage 88 (2014) 319–339.

[2] S.J. Luck, An Introduction to the Event-Related Potential Technique, MIT Press, 2014.

[3] U. Ghani, N. Signal, I.K. Niazi, D. Taylor, A novel approach to validate the efficacy of single task ERP paradigms to measure cognitive workload, Int. J. Psychophysiol. 158 (2020) 9–15, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167876020302191.

[4] M.D. Rugg, M.G. Coles, Electrophysiology of Mind: Event-Related Brain Potentials and Cognition, Oxford University Press, 1995.

[5] X. Xiao, M. Xu, J. Jin, Y. Wang, T.-P. Jung, D. Ming, Discriminative canonical pattern matching for single-trial classification of ERP components, IEEE Trans. Biomed. Eng. 67 (8) (2019) 2266–2275.

[6] N. Das, J. Vanthornhout, T. Francart, A. Bertrand, Stimulus-aware spatial filtering for single-trial neural response and temporal response function estimation in high-density EEG with applications in auditory research, Neuroimage 204 (2020) 116211.

[7] R. Li, J.C. Principe, M. Bradley, V. Ferrari, A spatiotemporal filtering methodology for single-trial ERP component estimation, IEEE Trans. Biomed. Eng. 56 (1) (2009) 83–92.

[8] D. Jarchi, S. Sanei, J.C. Principe, B. Makkiabadi, A new spatiotemporal filtering method for single-trial estimation of correlated ERP subcomponents, IEEE Trans. Biomed. Eng. 58 (1) (2011) 132–143.

[9] M.K.I. Molla, N. Morikawa, M.R. Islam, T. Tanaka, Data-adaptive spatio-temporal ERP cleaning for single-trial BCI implementation, IEEE Trans. Neural Syst. Rehabil. Eng. (2018).

[10] Y. Si, F. Li, K. Duan, Q. Tao, C. Li, Z. Cao, Y. Zhang, B. Biswal, P. Li, D. Yao, P. Xu, Predicting individual decision-making responses based on single-trial EEG, NeuroImage 206 (2020) 116333, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1053811919309243.

[11] M.J. Monesi, S.H. Sardouie, Extended common spatial and temporal pattern (ECSTP): A semi-blind approach to extract features in ERP detection, Pattern Recognit. 95 (2019) 128–135.

[12] A. Mobaien, N. Kheirandish, R. Boostani, A new approach based on principal ERPs and LDA to improve P300 mind spellers, in: 2021 7th International Conference on Signal Processing and Intelligent Systems, ICSPIS, IEEE, 2021, pp. 01–04.

[13] A. Mobaien, R. Boostani, N. Kheirandish, Signal activity detection in white Gaussian noise: Application to P300 detection, in: 2021 7th International Conference on Signal Processing and Intelligent Systems, ICSPIS, IEEE, 2021, pp. 01–05.

[14] Y. Zhang, G. Zhou, J. Jin, Q. Zhao, X. Wang, A. Cichocki, Aggregation of sparse linear discriminant analyses for event-related potential classification in brain-computer interface, Int. J. Neural Syst. 24 (01) (2014) 1450003.

[15] Y. Zhang, G. Zhou, J. Jin, Q. Zhao, X. Wang, A. Cichocki, Sparse Bayesian classification of EEG for brain–computer interface, IEEE Trans. Neural Netw. Learn. Syst. 27 (11) (2015) 2256–2267.

[16] R. Tomioka, K.-R. Müller, A regularized discriminative framework for EEG analysis with application to brain–computer interface, NeuroImage 49 (1) (2010) 415–432.

[17] A. Mobaien, R. Boostani, M. Mohammadi, S. Sanei, ERP detection based on smoothness priors, IEEE Trans. Biomed. Eng. (2022).

[18] H. Cecotti, A. Graser, Convolutional neural networks for P300 detection with application to brain-computer interfaces, IEEE Trans. Pattern Anal. Mach. Intell. 33 (3) (2010) 433–445.

[19] S. Kundu, S. Ari, MsCNN: a deep learning framework for P300-based brain–computer interface speller, IEEE Trans. Med. Robotics Bionics 2 (1) (2020) 86–93.

[20] R. Ma, T. Yu, X. Zhong, Z.L. Yu, Y. Li, Z. Gu, Capsule network for ERP detection in brain-computer interface, IEEE Trans. Neural Syst. Rehabil. Eng. 29 (2021) 718–730.

[21] Z. Zhang, G. Chen, S. Chen, A support vector neural network for P300 EEG signal classification, IEEE Trans. Artif. Intell. 3 (2) (2021) 309–321.

[22] K. Vanderperren, B. Mijović, N. Novitskiy, B. Vanrumste, P. Stiers, B.R. Van den Bergh, L. Lagae, S. Sunaert, J. Wagemans, S. Van Huffel, et al., Single trial ERP reading based on parallel factor analysis, Psychophysiology 50 (1) (2013) 97–110.

[23] A. Cichocki, R. Zdunek, A.H. Phan, S.-i. Amari, Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation, John Wiley & Sons, 2009.

[24] F. Cong, G. Zhou, P. Astikainen, Q. Zhao, Q. Wu, A.K. Nandi, J.K. Hietanen, T. Ristaniemi, A. Cichocki, Low-rank approximation based non-negative multi-way array decomposition on event-related potentials, Int. J. Neural Syst. 24 (08) (2014) 1440005.

[25] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, S. Yan, Tensor robust principal component analysis with a new tensor nuclear norm, IEEE Trans. Pattern Anal. Mach. Intell. 42 (4) (2019) 925–938.

[26] X. Gong, W. Chen, J. Chen, A low-rank tensor dictionary learning method for hyperspectral image denoising, IEEE Trans. Signal Process. 68 (2020) 1168–1180.

[27] Q. Shi, Y.-M. Cheung, Q. Zhao, H. Lu, Feature extraction for incomplete data via low-rank tensor decomposition with feature regularization, IEEE Trans. Neural Netw. Learn. Syst. 30 (6) (2018) 1803–1817.

[28] M.J. Idaji, M.B. Shamsollahi, S.H. Sardouie, Higher order spectral regression discriminant analysis (HOSRDA): A tensor feature reduction method for ERP detection, Pattern Recognit. 70 (2017) 152–162.

[29] N. Hao, M.E. Kilmer, K. Braman, R.C. Hoover, Facial recognition using tensor-tensor decompositions, SIAM J. Imaging Sci. 6 (1) (2013) 437–463.

[30] L.R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31 (3) (1966) 279–311.

[31] Q. Xie, Q. Zhao, D. Meng, Z. Xu, S. Gu, W. Zuo, L. Zhang, Multispectral images denoising by intrinsic tensor sparsity regularization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 1692–1700.

[32] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2011) 1–122.

[33] Y. Chang, L. Yan, X.-L. Zhao, H. Fang, Z. Zhang, S. Zhong, Weighted low-rank tensor recovery for hyperspectral image restoration, IEEE Trans. Cybern. 50 (11) (2020) 4558–4572.

[34] Q. Xie, Q. Zhao, D. Meng, Z. Xu, Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery, IEEE Trans. Pattern Anal. Mach. Intell. 40 (8) (2017) 1888–1902.

[35] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A comprehensive survey on transfer learning, Proc. IEEE 109 (1) (2020) 43–76.

[36] D. Wu, Y. Xu, B.-L. Lu, Transfer learning for EEG-based brain–computer interfaces: A review of progress made since 2016, IEEE Trans. Cogn. Dev. Syst. 14 (1) (2020) 4–19.

[37] J. Li, S. Qiu, Y.-Y. Shen, C.-L. Liu, H. He, Multisource transfer learning for cross-subject EEG emotion recognition, IEEE Trans. Cybern. 50 (7) (2019) 3281–3293.

[38] X. Wu, B. Zhou, Q. Ren, W. Guo, Multispectral image denoising using sparse and graph Laplacian Tucker decomposition, Comput. Vis. Media 6 (3) (2020) 319–331.

[39] H. Zhang, L. Liu, W. He, L. Zhang, Hyperspectral image denoising with total variation regularization and nonlocal low-rank tensor decomposition, IEEE Trans. Geosci. Remote Sens. 58 (5) (2019) 3071–3084.

[40] M.V. Afonso, J.M. Bioucas-Dias, M.A.T. Figueiredo, An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems, IEEE Trans. Image Process. 20 (3) (2011) 681–695.

[41] R. Lai, S. Osher, A splitting method for orthogonality constrained problems, J. Sci. Comput. 58 (2) (2014) 431–449.

[42] J.-F. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optim. 20 (4) (2010) 1956–1982.

[43] M. Tao, X. Yuan, Recovering low-rank and sparse components of matrices from incomplete and noisy observations, SIAM J. Optim. 21 (1) (2011) 57–81.

[44] A. Cichocki, Tensor decompositions: new concepts in brain data analysis? J. Soc. Instrum. Control Eng. 50 (7) (2011) 507–516.

[45] U. Hoffmann, J.-M. Vesin, T. Ebrahimi, K. Diserens, An efficient P300-based brain–computer interface for disabled subjects, J. Neurosci. Methods 167 (1) (2008) 115–125, Brain-Computer Interfaces (BCIs), [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165027007001094.

[46] A. Rakotomamonjy, V. Guigue, BCI competition III: dataset II-ensemble of SVMs for BCI P300 speller, IEEE Trans. Biomed. Eng. 55 (3) (2008) 1147–1154.