Recursive N-Way Partial Least Squares for Brain-Computer Interface

Andrey Eliseyev*, Tetiana Aksenova

CLINATEC, CEA, Grenoble, France

Abstract

In the article tensor-input/tensor-output blockwise Recursive N-way Partial Least Squares (RNPLS) regression is considered. It combines the multi-way tensors decomposition with a consecutive calculation scheme and allows blockwise treatment of tensor data arrays with huge dimensions, as well as the adaptive modeling of time-dependent processes with tensor variables. In the article the numerical study of the algorithm is undertaken. The RNPLS algorithm demonstrates fast and stable convergence of regression coefficients. Applied to Brain Computer Interface system calibration, the algorithm provides an efficient adjustment of the decoding model. Combining the online adaptation with easy interpretation of results, the method can be effectively applied in a variety of multi-modal neural activity flow modeling tasks.

Citation: Eliseyev A, Aksenova T (2013) Recursive N-Way Partial Least Squares for Brain-Computer Interface. PLoS ONE 8(7): e69962. doi:10.1371/journal.pone.0069962

Editor: Derek Abbott, University of Adelaide, Australia

Received February 28, 2013; **Accepted** June 14, 2013; **Published** July 26, 2013

Funding: This work was partially supported by grant CE ICoBI from the Nanosciences Foundation, RTRA and a non-restricted research grant from the Edmond J. Safra Philanthropic Foundation. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: eliseyev.andrey@gmail.com

Introduction

The Brain Computer Interface (BCI) is a system which translates recordings of the brain's neural activity into commands for external devices. For neuronal signal decoding, a control model is adjusted to an individual brain during the BCI system learning. This is called calibration. The model allows the control of an external effector at the stage of the online execution. Multi-way analysis was recently reported [1–8] to be an efficient way to calibrate BCI systems by providing simultaneous signal processing in several domains (temporal, frequency and spatial). Multi-way analysis represents a natural approach for modalities fusion [9]. It is based on the tensor data representation. Several tensor-based approaches have been invented (PARAFAC [10], Tucker [11], Non-negative Tensor Factorization [12], Sparse Nonnegative Tucker Decomposition [13], General Tensor Discriminant Analysis [14], Regularized Tensor Discriminant Analysis [7], etc.). Among them, a Multi-way or N-way Partial Least Squares (NPLS) regression was invented [15,16] as a generalization of ordinary Partial Least Squares (PLS) [17] for the case of tensor variables.

One of the major problems in BCI studies is the variability of the neuronal signals, in particular, due to brain plasticity. The changes in the neuronal activity require recalibration of the BCI systems. The full system recalibration is a time and labor-consuming procedure. Adaptive calibration aims to provide a fast adjustment of the BCI system in response to mild changes of the signal. For adaptive modeling the online (consecutive) algorithms are applied. In addition, consecutive algorithms are efficient tools to treat the signal for a long period of observation. Let us note that the task of calibration of the BCI system analyzing the signal in several domains (temporal, frequency and spatial) is characterized by significant duration of observation as well as by huge feature space dimensions. For the high dimensional observations (more variables than observations) Partial Least Squares modeling is particularly suited [17]. A blockwise Recursive Partial Least Squares [18] allows online identification of Partial Least Squares regression. N-way PLS (NPLS) [15] provides a generalization of ordinary PLS to the case of tensor variables. Similarly to the generic algorithm, NPLS combines regression analysis with the projection of data into the low dimensional space of latent variables using tensor factorization. The blockwise Recursive PLS adapted to multi-way structured inputs (tensor-input scalar-output) was invented by the authors recently [19]. This article presents blockwise Recursive NPLS (RNPLS) extended to the most general case of tensor-input/tensor-output, its numerical study, testing and comparison. The algorithm is addressed to a data set of huge dimensions and conducts a sequential blockwise tensor-data processing. Unlike the multi-pass blockwise Iterative NPLS [8], which repeatedly runs through the entire data set, the new RNPLS algorithm performs a consecutive calculation and can be applied online. Moreover, in the case of non-stationary tensor-valued processes, RNPLS allows adaptive learning by introducing the forgetting factor.

Extending the results published in [19], the present article generalizes RPLS to the case of tensor-input and tensor-output variables. Moreover, it studies the convergence of regression coefficients as well as their variance and expectation and compares them with "true" ones. Finally, the computational efficiency and the generalization ability are compared for a set of PLS family methods (RNPLS, generic NPLS, multipass blockwise Iterative N-way Partial Least Squares (INPLS), Unfolded PLS (UPLS)). For this purpose, a set of computational experiments were carried out.

Finally, the generalization abilities of the algorithms are examined using recordings from real-life BCI preclinical experiments. In particular, RNPLS is applied to solve the tensor-input and scalar-output problem of binary events prediction in long-term experiments in freely moving animals.

Generalization of the algorithm allows us to consider another problem of continuous hand movement trajectory reconstruction from Electrocorticogramm (ECoG) recordings: simultaneous prediction of 3D coordinates of the hand at three points (wrist, elbow, shoulder).

Methods

The RNPLS algorithm unites the recursive calculation scheme of Recursive PLS regression with the multimodal data representation of NPLS.

Generic PLS

Ordinary PLS [17] is an iterative procedure to model a linear relationship between vector input and output variables on the basis of matrices of observations, $\mathbf{X}$ and $\mathbf{Y}$:$\mathbf{Y}=\mathbf{XC}+\mathbf{V}$, where $\mathbf{V}$ and $\mathbf{C}$ are noise and coefficient matrices respectively. PLS is a statistical method particularly suited to the high dimensional observation vectors. In order to build the model, the observations are projected into the low dimensional spaces of latent variables in such a way that the maximum variances of $\mathbf{X}$ and $\mathbf{Y}$ are explained simultaneously. At the first iteration, the matrices $\mathbf{X}$ and $\mathbf{Y}$ are represented as $\mathbf{X}=\mathbf{t}_1\mathbf{p}_1^{\mathrm{T}}+\mathbf{E}_1$, $\mathbf{Y}=\mathbf{u}_1\mathbf{q}_1^{\mathrm{T}}+\mathbf{F}_1$, where $\mathbf{t}_1$ and $\mathbf{u}_1$are the latent variables (score vectors), while $\mathbf{p}_1$ and $\mathbf{q}_1$ are the loading vectors. $\mathbf{E}_1$ and $\mathbf{F}_1$ are the matrices of residuals. The score vectors are calculated to maximize the covariance between $\mathbf{t}_1$ and $\mathbf{u}_1$. The coefficient b_1 of a regression $\mathbf{u}_1=\mathrm{b}_1\mathbf{t}_1+\mathbf{r}_1$ is calculated to minimize the norm of the residuals, $\mathbf{r}_1$. Then the procedure is applied iteratively F times to the residual matrices. It is demonstrated in [18] that the latent variables could be constructed as orthonormal:

$$\mathbf{T}^{\mathrm{T}}\mathbf{T}=\mathbf{I}_F, \tag{1}$$

where $\mathbf{T}=[\mathbf{t}_1,\ldots,\mathbf{t}_F]$, $\mathbf{I}_F$ is identity matrix.

Generic NPLS

NPLS is an algorithm of the PLS family adapted to multimodal data (tensor variables). Tensors, or multi-way arrays, are higher-order generalizations of vectors and matrices. Elements of a tensor $\underline{\mathbf{X}}\in\Re^{I_1\times I_2\times\ldots\times I_N}$ are denoted $x_{i_1,i_2,\ldots,i_N}$. Here, N is the order of the tensor, i.e., the number of dimensions, also known as ways or modes. Vectors and matrices are tensors of order one and two respectively. The number of variables I_i in the mode i shows the dimensionality of the mode (for more details, see [20]). The tensor $\underline{\mathbf{X}}\in\Re^{I_1\times I_2\times\ldots\times I_N}$ can be unfolded into a matrix along the *i-th* mode [20]. In this paper this operation of unfolding along the *i-th* mode is referred to as $(\underline{\mathbf{X}})_{(i)}$. The vectorization of tensor $\underline{\mathbf{X}}$ is denoted as $\mathrm{vec}(\underline{\mathbf{X}})$ [20].

NPLS models a linear relationship between input and output variables on the basis of tensors of observations $\underline{\mathbf{X}}\in\Re^{n\times I_1\times\ldots\times I_N}$ and $\underline{\mathbf{Y}}\in\Re^{n\times J_1\times\ldots\times J_M}$. The first dimension of the array corresponds to the points of observation, while other dimensions are related to the modes of analysis (e.g., time, frequency, and space).To project data into the low dimensional feature space, NPLS uses the tensors decomposition with a set of projectors corresponding to each modality. At the first iteration, the tensors $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$ are represented as.

$$\underline{\mathbf{X}}=\mathbf{t}_1\circ\mathbf{w}_1^1\circ\ldots\circ\mathbf{w}_1^N+\underline{\mathbf{E}}_1,$$

$$\underline{\mathbf{Y}}=\mathbf{u}_1\circ\mathbf{q}_1^1\circ\ldots\circ\mathbf{q}_1^M+\underline{\mathbf{F}}_1. \tag{2}$$

The operation "$\circ$" is the outer product [20]. The latent variables $\mathbf{t}_1\in\Re^n$ and $\mathbf{u}_1\in\Re^n$ are extracted from the first mode of the tensors $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$, providing maximum covariance between $\mathbf{t}_1$ and $\mathbf{u}_1$. In parallel, the algorithm forms the factors, i.e. the sets of vectors $\{\mathbf{w}_1^1\in\Re^{I_1},\ \ldots,\ \mathbf{w}_1^N\in\Re^{I_N}\}$, $\|\mathbf{w}_1^i\|_{i=1}^N=1$, and $\{\mathbf{q}_1^1\in\Re^{J_1},\ \ldots,\ \mathbf{q}_1^M\in\Re^{J_M}\}$ related to corresponding modes of $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$. They are found in such a way that projection of the tensor $\underline{\mathbf{X}}$ on the vectors $\{\mathbf{w}_1^1,\ \ldots,\ \mathbf{w}_1^N\}$ results in $\mathbf{t}_1$ and projection of the tensor $\underline{\mathbf{Y}}$ on the vectors $\{\mathbf{q}_1^1,\ \ldots,\ \mathbf{q}_1^M\}$ results in $\mathbf{u}_1$. The coefficient b_1 of regression $\mathbf{u}_1=b_1\mathbf{t}_1+\mathbf{e}_1$ is calculated by least squares. Next, factors are calculated in the same way by decomposing the residuals $\underline{\mathbf{E}}_1$ and $\underline{\mathbf{F}}_1$. NPLS latent variables $\mathbf{T}=[\mathbf{t}_1,\ldots,\mathbf{t}_F]$ are not orthogonal. A detailed description of NPLS can be found, for example, in [16].

Recursive PLS

The recursive PLS algorithms [18,21,22] were invented to take into account time-dependent changes in the data and were designed for matrices of observations. In comparison to other methods, Qin's blockwise recursive algorithm is more efficient in updating the model with respect to the computational cost and memory [18]. To model a linear relationship between vector input and output variables $\mathbf{Y}=\mathbf{XC}+\mathbf{V}$ on the basis of matrices of observations, $\mathbf{X}$ and $\mathbf{Y}$ ($\mathbf{V}$, $\mathbf{C}$ are noise and coefficient matrices respectively), Qin's algorithm considers the decomposition of the matrices $\mathbf{X}$ and $\mathbf{Y}$ with orthonormal input latent variables $\mathbf{T}^{\mathrm{T}}\mathbf{T}=\mathbf{I}_F$:

$$\mathbf{X}=\mathbf{TP}^{\mathrm{T}}+\mathbf{E}_F,\quad \mathbf{Y}=\mathbf{UQ}^{\mathrm{T}}+\mathbf{F}_F=\mathbf{TBQ}^{\mathrm{T}}+\mathbf{F}_F,$$

where $\mathbf{P}=[\mathbf{p}_1,\ldots,\mathbf{p}_F]$, $\mathbf{Q}=[\mathbf{q}_1,\ldots,\mathbf{q}_F]$ are the matrices of loading vectors, $\mathbf{T}=[\mathbf{t}_1,\ldots,\mathbf{t}_F]$ and $\mathbf{U}=[\mathbf{u}_1,\ldots,\mathbf{u}_F]$ are the matrices of latent variables (score), and $\mathbf{B}=\mathrm{diag}\{b_1,\ldots,b_F\}$ are regression coefficients for latent variables. In addition, by construction latent variables $\mathbf{T}=[\mathbf{t}_1,\ldots,\mathbf{t}_F]$ are orthogonal to residuals:

$$\mathbf{T}^T\mathbf{E}_F=\mathbf{0}, \tag{3}$$

$$\mathbf{T}^T\mathbf{F}_F=\mathbf{0}. \tag{4}$$

It has been shown [18] that if F is large enough to provide $\mathbf{E}_F^{\mathrm{T}}\mathbf{E}_F=0$, then $\mathbf{X}^T\mathbf{X}=\mathbf{PP}^T$ and $\mathbf{X}^T\mathbf{Y}=\mathbf{PBQ}^T$. This yields that, for the new data matrices $\{\mathbf{X}_1,\mathbf{Y}_1\}$, regression on the joint data sets,

$$\left\{\begin{bmatrix}\mathbf{X}\\ \mathbf{X}_1\end{bmatrix},\begin{bmatrix}\mathbf{Y}\\ \mathbf{Y}_1\end{bmatrix}\right\}$$

is equivalent to the regression calculated using the loading matrices ($\mathbf{P}$ and $\mathbf{Q}$) and the matrix of coefficients $\mathbf{B}$ from previous observations:

$$\left\{\begin{bmatrix}\mathbf{X}\\ \mathbf{X}_1\end{bmatrix},\begin{bmatrix}\mathbf{Y}\\ \mathbf{Y}_1\end{bmatrix}\right\}\Leftrightarrow\left\{\begin{bmatrix}\mathbf{P}^{\mathrm{T}}\\ \mathbf{X}_1\end{bmatrix},\begin{bmatrix}\mathbf{BQ}^{T}\\ \mathbf{Y}_1\end{bmatrix}\right\}$$

Here, the equivalence of regressions $\{\mathbf{X}_1,\mathbf{Y}_1\}\Leftrightarrow\{\mathbf{X}_2,\mathbf{Y}_2\}$ means an equality of the least square estimates of regression coefficients: $\hat{\mathbf{C}}_1=\hat{\mathbf{C}}_2$, $\hat{\mathbf{C}}_i=\arg\min_{\mathbf{C}}\|\mathbf{Y}_i-\mathbf{X}_i\mathbf{C}\|=\left(\mathbf{X}_i^{\mathrm{T}}\mathbf{X}_i\right)^{+}\mathbf{X}_i^{\mathrm{T}}\mathbf{Y}_i$, $i=1,2$.

Thus the old data sets $\mathbf{X}$ and $\mathbf{Y}$ are captured by the loading matrices and regression coefficients, while the new data is added directly to them. As a result, the algorithm always keeps the size of the processing matrices.

Blockwise recursive PLS was applied for the adaptive modeling and monitoring of time-varying and non-stationary processes with slow and abrupt changes [18,23–25] by introducing the sliding window and/or forgetting factor.

$$\left\{\begin{bmatrix}\gamma\mathbf{P}^{\mathrm{T}}\\ \mathbf{X}_1\end{bmatrix},\begin{bmatrix}\gamma\mathbf{BQ}^{\mathrm{T}}\\ \mathbf{Y}_1\end{bmatrix}\right\}\tag{5}$$

The forgetting factor γ is either fixed over the entire range of 0 to 1, or estimated dynamically by assessing changes in the current data [24].

Recursive NPLS

To apply the recursive approach to the NPLS algorithm, orthonormality of latent variables $\mathbf{T}=[\mathbf{t}_1,\ldots,\mathbf{t}_F]$ as well as their orthogonality to residuals should be provided (see Equation 1, Equation 3 and Equation 4). Let us note that the NPLS latent variables do not satisfy these conditions. The modification of NPLS to meet the requirements of Equation 1, Equation 3, and Equation 4 integrated to a recursive algorithm is described below.

Similar to recursive PLS, RNPLS is a sequential blockwise algorithm. Data blocks are received and then processed. The first block of data, namely tensors $\underline{\mathbf{X}}\in\Re^{n\times I_1\times\ldots\times I_N}$ and $\underline{\mathbf{Y}}\in\Re^{n\times J_1\times\ldots\times J_M}$, is factorized with NPLS. The procedure results in the sets of projectors $\left\{\mathbf{w}_f^1\in\Re^{I_1},\ldots,\mathbf{w}_f^N\in\Re^{I_N}\right\}_{f=1}^{F}$, $\left\{\mathbf{q}_f^1\in\Re^{J_1},\ldots,\mathbf{q}_f^M\in\Re^{J_M}\right\}_{f=1}^{F}$, as well as the matrix of latent variables $\mathbf{T}_F\in\Re^{n\times F}$ and regression coefficients $\left\{\mathbf{b}_f\in\Re^{f}\right\}_{f=1}^{F}$ (Algorithm S1, steps 4–17). It provides the approximations $\underline{\mathbf{X}}=\sum_{f=1}^{F}\mathbf{t}_f\circ\mathbf{w}_f^1\circ\ldots\circ\mathbf{w}_f^N+\underline{\mathbf{E}}_F$ and $\underline{\mathbf{Y}}=\sum_{f=1}^{F}\mathbf{u}_f\circ\mathbf{q}_f^1\circ\ldots\circ\mathbf{q}_f^M+\underline{\mathbf{F}}_F$.

For the next steps, we use the matrix representation. It does not affect the proposed approach, but simplifies the notation. To convert to the matrix form, the tensors $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$ are unfolded along the first mode into the matrices $\mathbf{X}\in\Re^{n\times(I_1\ldots I_N)}$ and $\mathbf{Y}\in\Re^{n\times(J_1\ldots J_M)}$: $\mathbf{X}=(\underline{\mathbf{X}})_{(1)}$, $\mathbf{Y}=(\underline{\mathbf{Y}})_{(1)}$. At the same time, let us denote the vectorization of the tensors $\mathbf{w}_f=\mathrm{vec}\left(\mathbf{w}_f^1\circ\ldots\circ\mathbf{w}_f^N\right)$, $\mathbf{w}_f\in\Re^{I_1\ldots I_N}$, $\mathbf{W}_f=\left[\mathbf{w}_1|\ldots|\mathbf{w}_f\right]$ and $\mathbf{q}_f=\mathrm{vec}\left(\mathbf{q}_f^1\circ\ldots\circ\mathbf{q}_f^M\right)$, $\mathbf{q}_f\in\Re^{J_1\ldots J_M}$ (Algorithm S1, steps 18–21). In the matrix notation $\mathbf{X}=\mathbf{T}_F\mathbf{W}_F^{\mathrm{T}}+\mathbf{E}_F$, $\mathbf{Y}=\sum_{f=1}^{F}\mathbf{T}_f\mathbf{b}_f\mathbf{q}_f^{\mathrm{T}}+\mathbf{F}_F$.

In order to use Qin's approach, the orthogonality of latent variables $\mathbf{T}_F$ is required (see Equation 1). Since the NPLS latent variables $\mathbf{T}_F$ are not orthogonal, let us orthonormalize them $\mathbf{T}_F^{\perp}$: $\left(\mathbf{T}_F^{\perp}\right)^{T}\mathbf{T}_F^{\perp}=\mathbf{I}_F$. It is constructed in a way to provide additional orthogonalization of $\mathbf{T}_F^{\perp}$ to the residuals matrix $\left(\mathbf{T}_F^{\perp}\right)^{T}\mathbf{F}_F=\mathbf{0}$. $\mathbf{T}_F^{\perp}$ can be obtained, for example, from a Gauss orthogonalization procedure [26]. For the new orthonormal latent variables $\mathbf{X}=\mathbf{T}_F\mathbf{W}_F^{\mathrm{T}}+\mathbf{E}_F=\mathbf{T}_F^{\perp}\mathbf{P}_F^{\mathrm{T}}+\mathbf{E}_F$, where $\mathbf{P}_F^{\mathrm{T}}=\left(\mathbf{T}_F^{\perp}\right)^{T}\mathbf{T}_F\mathbf{W}_F^{\mathrm{T}}$ (see Algorithm S1, steps 22–25). After the orthonormalization, Equation 1 and Equation 4 are satisfied, i.e., $\left(\mathbf{T}_F^{\perp}\right)^{T}\mathbf{T}_F^{\perp}=\mathbf{I}_F$ and $\left(\mathbf{T}_F^{\perp}\right)^{T}\mathbf{F}_F=\mathbf{0}$.

To provide the orthogonality of the matrix $\mathbf{T}_F^{\perp}$ and the residual matrix $\mathbf{E}_F$ (Equation 3), let us subtract from the residual matrix its projection on all latent variables $\left\{\mathbf{t}_f^{\perp}\right\}_{f=1}^{F}$: $\widetilde{\mathbf{E}}_F=\mathbf{E}_F-\sum_{f=1}^{F}\mathbf{t}_f^{\perp}\left(\mathbf{t}_f^{\perp}\right)^{T}\mathbf{E}_F$. Thus, $\mathbf{X}=\mathbf{T}_F^{\perp}\widetilde{\mathbf{P}}^{T}+\widetilde{\mathbf{E}}_F$, with a new matrix of loadings $\widetilde{\mathbf{P}}=[\widetilde{\mathbf{p}}_1,\ldots,\widetilde{\mathbf{p}}_F]$, $\widetilde{\mathbf{p}}_f=\mathbf{p}_f+\mathbf{E}_F^{\mathrm{T}}\mathbf{t}_f^{\perp}$. The relation $\left(\mathbf{T}_F^{\perp}\right)^{\mathrm{T}}\widetilde{\mathbf{E}}_F=\mathbf{0}$ now holds. Let us define $\widetilde{\mathbf{Q}}^{T}=\left(\mathbf{T}_F^{\perp}\right)^{T}\sum_{f=1}^{F}\mathbf{T}_f\mathbf{b}_f\mathbf{q}_f^{\mathrm{T}}$ (see Algorithm S1, steps 26–30).

Since Equation 1, Equation 3 and Equation 4 are satisfied, then $\mathbf{X}^T\mathbf{X}=\widetilde{\mathbf{P}}\widetilde{\mathbf{P}}^{T}$, $\mathbf{X}^T\mathbf{Y}=\widetilde{\mathbf{P}}\widetilde{\mathbf{Q}}^{T}$. Hence, the least squares coefficients are equivalent for the data sets [18]:

$$\left\{\begin{bmatrix}\mathbf{X}\\ \mathbf{X}^{\mathrm{new}}\end{bmatrix},\begin{bmatrix}\mathbf{Y}\\ \mathbf{Y}^{\mathrm{new}}\end{bmatrix}\right\}\Leftrightarrow\left\{\begin{bmatrix}\widetilde{\mathbf{P}}^{\mathrm{T}}\\ \mathbf{X}^{\mathrm{new}}\end{bmatrix},\begin{bmatrix}\widetilde{\mathbf{Q}}^{\mathrm{T}}\\ \mathbf{Y}^{\mathrm{new}}\end{bmatrix}\right\}$$

In tensor notation (see Algorithm S1, steps 31–33):

$$\left\{\begin{bmatrix}\underline{\mathbf{X}}\\ \underline{\mathbf{X}}^{new}\end{bmatrix},\begin{bmatrix}\underline{\mathbf{Y}}\\ \underline{\mathbf{Y}}^{new}\end{bmatrix}\right\}\Leftrightarrow\left\{\begin{bmatrix}\underline{\widetilde{\mathbf{P}}}\\ \underline{\mathbf{X}}^{new}\end{bmatrix},\begin{bmatrix}\underline{\widetilde{\mathbf{Q}}}\\ \underline{\mathbf{Y}}^{new}\end{bmatrix}\right\}$$

Here, a tensor $\underline{\widetilde{\mathbf{P}}}$ is obtained from the matrix $\widetilde{\mathbf{P}}$, with F as the dimensionality of the first mode. The dimensions of the other modes are equal to the dimensions of the corresponding modes of $\underline{\mathbf{X}}$. Similarly $\underline{\widetilde{\mathbf{Q}}}$ is obtained from the matrix $\widetilde{\mathbf{Q}}$ according to the dimensions of the corresponding modes of $\underline{\mathbf{Y}}$.

Thus, the RNPLS algorithm inherits the tensor representation of the data and allows consecutive learning, which is a property of recursive PLS. Besides tensors $\underline{\widetilde{\mathbf{P}}}$ and $\underline{\widetilde{\mathbf{Q}}}$ for recursive calculations, the sets of projectors $\left\{\mathbf{w}_f^1\in\Re^{I_1},\ldots,\mathbf{w}_f^N\in\Re^{I_N}\right\}_{f=1}^{F}$, $\left\{\mathbf{q}_f^1\in\Re^{J_1},\ldots,\mathbf{q}_f^M\in\Re^{J_M}\right\}_{f=1}^{F}$ and coefficients $\left\{\mathbf{b}_f\in\Re^{f}\right\}_{f=1}^{F}$ are identified. They are used at the prediction stage to estimate the output $\hat{\underline{\mathbf{Y}}}$ in the same way as it is done in the traditional NPLS [15]. A graphical representation of the RNPLS algorithm is shown in Figure 1.

All schemes of adaptive modeling developed for the matrix RPLS can be directly applied to adaptive learning of RNPLS. For instance, similarly to Equation (5) we can introduce the forgetting factor $\gamma\in[0,1]$:

$$\left\{\begin{bmatrix}\gamma\underline{\widetilde{\mathbf{P}}}\\ \underline{\mathbf{X}}^{new}\end{bmatrix},\begin{bmatrix}\gamma\underline{\widetilde{\mathbf{Q}}}\\ \underline{\mathbf{Y}}^{new}\end{bmatrix}\right\}$$

Simulation Study

Both the generalization ability (prediction error) and the convergence of model coefficients were studied in simulation

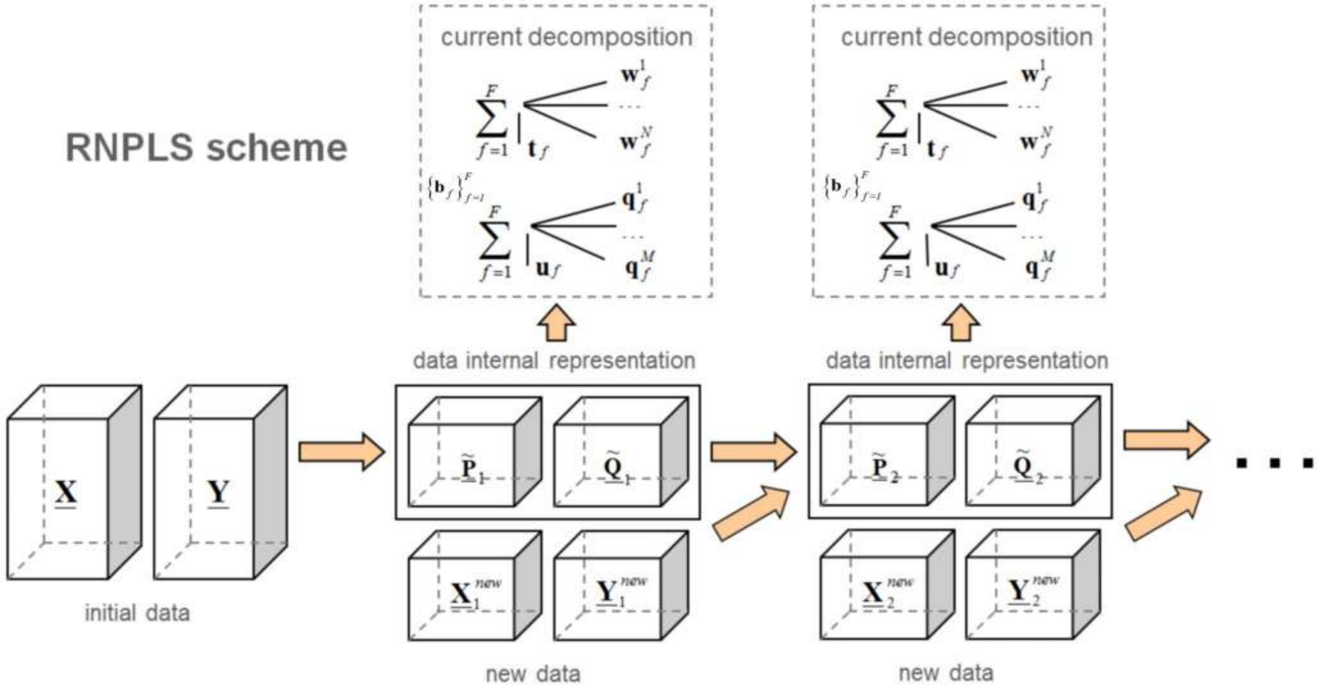
RNPLS scheme
current decomposition
$\sum_{f=1}^{F}$
$\mathbf{t}_f$
$\mathbf{w}_f^1$
...
$\mathbf{w}_f^N$
$\{\mathbf{b}_f\}_{f=1}^F$
$\mathbf{u}_f$
$\mathbf{q}_f^1$
$\mathbf{q}_f^M$
data internal representation
$\underline{\mathbf{X}}$
$\underline{\mathbf{Y}}$
initial data
$\underline{\widetilde{\mathbf{P}}}_1$
$\underline{\widetilde{\mathbf{Q}}}_1$
$\underline{\mathbf{X}}_1^{new}$
$\underline{\mathbf{Y}}_1^{new}$
$\underline{\widetilde{\mathbf{P}}}_2$
$\underline{\widetilde{\mathbf{Q}}}_2$
$\underline{\mathbf{X}}_2^{new}$
$\underline{\mathbf{Y}}_2^{new}$
new data
...

Figure 1. The RNPLS scheme. Information from the tensors of observation $\underline{\mathbf{X}}$ and $\underline{\mathbf{Y}}$ of order $(1+N)$ and $(1+M)$, respectively, is captured by their loading tensors $\underline{\widetilde{\mathbf{P}}}$ and $\underline{\widetilde{\mathbf{Q}}}$. On every iteration, the algorithm generates the current sets of the coefficient vectors $\{\mathbf{b}_f\}_{f=1}^F$, as well as the projection vectors $\left\{\mathbf{w}_f^1,\ldots,\mathbf{w}_f^N\right\}_{f=1}^F$ and $\left\{\mathbf{q}_f^1,\ldots,\mathbf{q}_f^M\right\}_{f=1}^F$ used to estimate the dependent variable $\hat{\underline{\mathbf{y}}}$ on the prediction stage.
doi:10.1371/journal.pone.0069962.g001

experiments. The resulted prediction errors of the proposed approach were compared with ones generated by other state of the art methods.

The computational experiments were performed on simulated datasets for matrix input and scalar output, the simplest case for simulation and comparison of tensor based algorithms. Keeping the essential properties, it simplifies the visualization and interpretation of results.

A set of PLS family offline algorithms for tensor data analysis, namely, generic NPLS, Iterative NPLS, and unfolded PLS, were compared with new sequential RNPLS. Let us note that all these algorithms generate the set of projectors and the set of coefficients of linear regression in low dimensional spaces of latent variables to predict the output variable from the input one. This model can be rewritten in the original variables. In the particular case of scalar $y(t)\in\Re$ and matrix $\mathbf{x}(t)\in\Re^{I_1\times I_2}$,

$$\hat{y}(t)=\sum_{i,j}x_{i,j}(t)c_{i,j},\quad \mathrm{C}=\left\{c_{i,j}\right\}_{i,j},$$

The matrix of coefficients $\mathbf{C}$ is estimated by each algorithm. To analyze the convergence properties of sequential blockwise RNPLS, the training data were split into K disjoint subsets. The RNPLS algorithm was consequently applied to these subsets, i.e., each following subset was used to adjust the solution from the previous step. $\mathbf{C}_k$ is the approximation of the regression coefficient matrix estimated at step $k=1,\ldots,K$.

The variance and expectation of the coefficients evaluated by the different methods were examined in comparison with "true" coefficients. Prediction errors of algorithms mentioned above were compared on simulated datasets. Furthermore, the adaptive properties of the RNPLS-based adaptive algorithm to abrupt changes in the observed process were evaluated.

The tests were performed for the binary output variable which corresponds to the binary BCI experiment described below (see details in [8]). Despite its simplicity, this model makes it possible to explore the fundamental properties of the method. An artificial data set $\left\{\mathbf{x}_k\in\Re^{100\times 200},y_k\in\{0,1\}\right\}_k$was generated following [19]. Namely, binary outputs y_k were randomly generated (0 or 1) with equal probabilities. Matrices of input $\mathbf{x}_k$ (Figure 2A) were calculated according to $\mathbf{x}_k=\mathbf{c}(y_k)+\lambda\varepsilon_k$,

$$c_{ij}=\begin{cases}\cos(2.5\pi(i-0.4j)+2) & \text{if } y_k=0,\\ \sin(2.5\pi(i+0.4j)+1) & \text{if } y_k=1.\end{cases}$$

The random noise $\varepsilon_k\in\Re^{100\times 200}$ was drawn from a multivariate normal distribution $\mathcal{N}(\mathbf{0},\underline{\mathbf{\Sigma}})$. To take into account the covariance structure of the real data, the covariance tensor $\underline{\mathbf{\Sigma}}$ was defined as a Kronecker product of two matrices $\underline{\mathbf{\Sigma}}=\mathbf{\Sigma}_1\otimes\mathbf{\Sigma}_2$ where the matrices $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$ were estimated as covariance matrices of the temporal and frequency modalities of the real brain ECoG activity recordings (binary BCI is described below in the Section *Binary BCI*). Correlation matrices are shown in Figure 2B. Thus the artificial data inherit some of the statistical properties of the real recordings. The same noise distribution was applied for both classes in the simulated data. The noise ε_k was added to the templates with coefficients $\lambda=0.5$, 1, 5, 10. It has the same amplitude as the signal $\mathbf{c}(y_k)$ if $\lambda=1$.

In the particular case of this computational experiment, the matrix of regression coefficient $\mathbf{C}$ belongs to $\Re^{100\times 200}$.

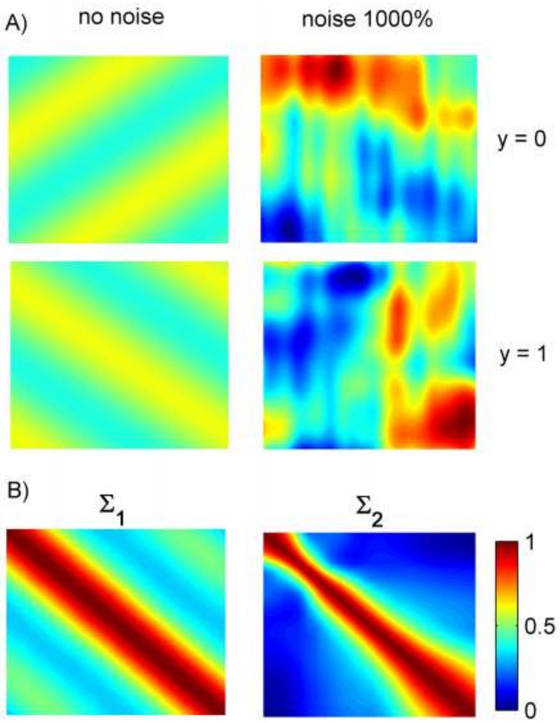

Figure 2. Artificial data. A) Points $\{\mathbf{x}, \mathbf{y}\}$ without noise and with 1000% of noise ($\lambda = 10$) from the simulated data set. B) Covariance tensor $\underline{\mathbf{\Sigma}} = \mathbf{\Sigma}_1 \otimes \mathbf{\Sigma}_2$ to generate samples of the simulated data set is defined as a Kronecker product of covariance matrices $\mathbf{\Sigma}_1$ and $\mathbf{\Sigma}_2$. Matrices Σ_1 and Σ_2 were estimated from experimental data (Section *Binary BCI*) as correlation matrix for the temporal and frequency modalities in the most informative time interval [0, −1.5] s and frequency band [50, 300] Hz.
doi:10.1371/journal.pone.0069962.g002

Convergence of RNPLS Regression Coefficients

Generally speaking, the convergence of the output variables y does not require the convergence of the regression coefficients $\mathbf{C}$ so far as the convergence of the regression coefficients is a stronger requirement.

To analyze the convergence of regression coefficients, we examined the distance between 2 matrices $\mathbf{C}_k$ and $\mathbf{C}_l$,

$$\rho(\mathbf{C}_k, \mathbf{C}_l) = \|\mathbf{C}_k - \mathbf{C}_l\|_{\mathrm{F}},$$

where $\|...\|_{\mathrm{F}}$ is the Frobenius norm.

In addition, to evaluate the RNPLS regression coefficients, they were compared with the "true" ones. The "true" coefficients are estimated using the simulated data without noise ($\lambda = 0$). In the particular case, the difference of the centers of the classes represents the matrix of regression coefficients (up to a scaling factor and a constant term). They could be evaluated, for instance, by the ordinary PLS.

To apply blockwise RNPLS, a simulated data set (800 points) was split into 40 disjoint subsets (blocks), each one containing 20 points. The level of noise was taken to be 1000% of the signal amplitude ($\lambda = 10$). The convergence of the regression coefficient matrices for the series of computational experiments (10 realizations) is represented in Figure 3 and Figure 4. The changes in $\mathbf{C}$ become insignificant after approximately 10–15 iterations. The number of factors, $F = 9$, was chosen in such a way as to minimize the average root mean squared error (RMSE) on the test dataset over 10 realizations of the training dataset.

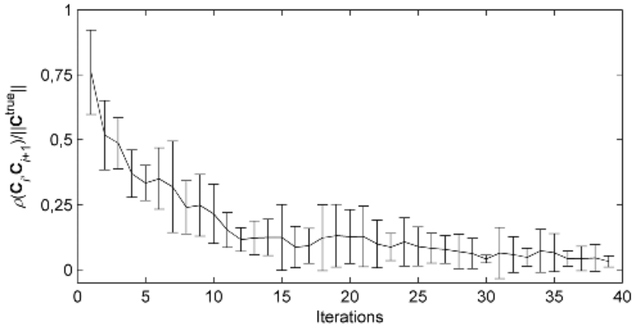

Figure 3. Convergence of the regression coefficients matrices. Convergence of the coefficients matrices versus iterations i (mean and standard deviation) of the RNPLS algorithm in the series of experiments (10 realizations, simulated dataset, level of noise 1000%). The distance ρ between two successive coefficient matrices decreases on ~70% (in ~3.5 times) after the first 12–15 iterations.
doi:10.1371/journal.pone.0069962.g003

Adaptivity

The second computational experiment was performed to verify the adaptive properties of the RNPLS. An abrupt change was introduced to the simulated data set (2000 points). Namely, the indexes of the classes were reversed from the point number 1000 for the rest of the data set. The level of the noise was chosen equal to 1000% ($\lambda = 10$). 10 realizations of the noise were generated.

To apply adaptive blockwise RNPLS, the data was split sequentially into blocks containing 50 points. The same number of factors ($F = 9$) was applied. For the first 20 blocks ($i = 1, \ldots ,20$), the algorithm successfully converged to "true" regression coefficients. For the next subsets ($i = 21, \ldots ,40$) the classes and, therefore, the "true" regression coefficients were reversed. Figure 5 demonstrates that the RNPLS algorithm adjusted its solution to the new situation in 15 iterations ($i = 21, \ldots ,35$) (forgetting factor $\gamma = 0.9$).

Comparison of Algorithms

To study whether the recursive scheme of the RNPLS algorithm decreases the accuracy of modeling, the proposed method was compared with generic offline NPLS, iterative NPLS and unfolded PLS on the simulated data. The percentage of prediction errors (RMSE) was estimated depending on the level of noise and the number of factors. The variance of the coefficients was studied for the set of algorithms. Finally these coefficients were compared with the "true" ones.

The entire simulated dataset (1600 points) was equally split into the training and the test datasets. The NPLS, iterative NPLS, and unfolded PLS algorithms processed the whole training set. For the recursive calculation, the training dataset was split into 20 disjoint windows, each one containing 40 points. For all conditions (different noise level and the number of factors), the experiment was repeated 10 times with new realizations of noise.

The percentage of prediction errors was estimated using 10 realizations of the test dataset depending on the level of noise and number of factors for all the algorithms. RNPLS demonstrated a significantly smaller number of factors which are necessary for efficient prediction (Table 1). Let us remember that the RNPLS showed better robustness, as the variation of the prediction errors was essentially smaller for the recursive algorithm for a small number of factors [19].

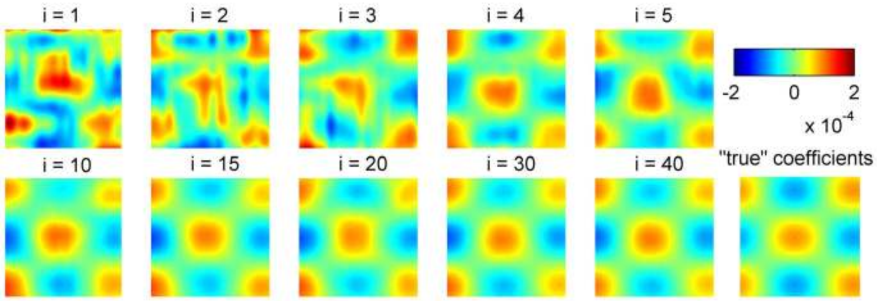

Figure 4. Examples of the regression coefficients. The regression coefficients $\mathbf{C}$ depending on the iteration number i of the RNPLS algorithm in one of the computational experiments for the simulated dataset, level of noise 1000%.
doi:10.1371/journal.pone.0069962.g004

To compare the variance of the estimated coefficients, RNPLS, NPLS, INPLS, and UPLS algorithms were applied to 10 realizations of the training dataset with the level of noise $\lambda = 10$ (1000%). The factor numbers were chosen equal to 9, 17, 17, and 13, for RNPLS, NPLS, INPLS, and UPLS, respectively, in order to minimize the average RMSE in the 10-fold cross-validation procedure. For this procedure, the training set is randomly partitioned into 10 equal size subsets. 9 subsets are used for training, whereas one is used for the test, i.e., RMSE calculation. The process is then repeated 10 times and the 10 results are averaged to produce a single RMSE estimation. The whole procedure is repeated for the different number of factors to find the one which minimizes RMSE. For each calibration approach, for the optimal number of factors, the images of the obtained regression coefficients $\left\{\mathbf{C}_k^{\mathrm{RNPLS}}\right\}_{k=1}^{10}$, $\left\{\mathbf{C}_k^{\mathrm{NPLS}}\right\}_{k=1}^{10}$, $\left\{\mathbf{C}_k^{\mathrm{INPLS}}\right\}_{k=1}^{10}$ and $\left\{\mathbf{C}_k^{\mathrm{unfolded}\ PLS}\right\}_{k=1}^{10}$ are represented in Figure 6. The variability of the unfolded PLS results ($\sigma_{\mathrm{unfolded}\ PLS} = 3.3 \cdot 10^{-5}$) is almost 5 times greater than the variability of the RNPLS results ($\sigma_{\mathrm{RNPLS}} = 0.7 \cdot 10^{-5}$), whereas the variabilities of INPLS and NPLS are $\sigma_{INPLS} = 0.8 \cdot 10^{-5}$ and $\sigma_{\mathrm{NPLS}} = 1.6 \cdot 10^{-5}$, respectively.

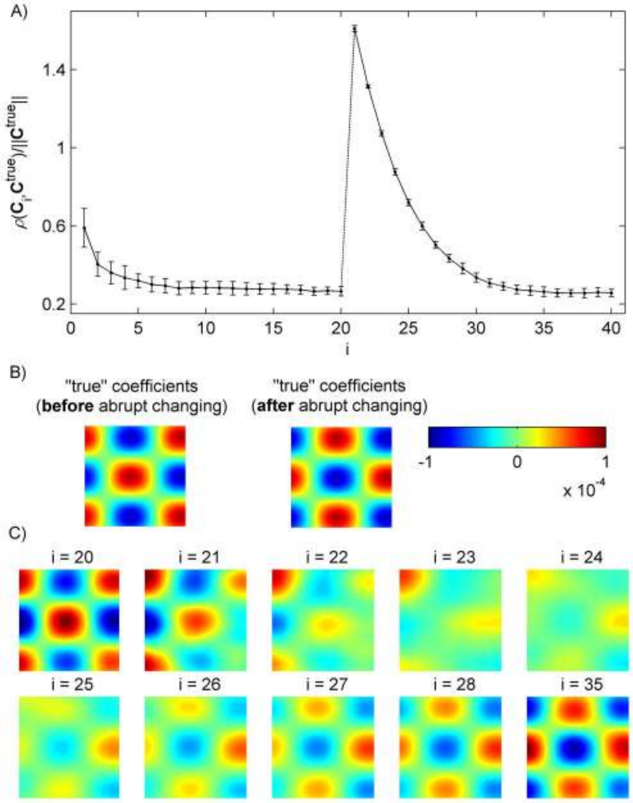

Figure 5. Adjustment of the RNPLS regression coefficients to abrupt changes in observations. A) The distance ρ (mean and standard deviation) between the RNPLS and "true" coefficients versus the iterations of the RNPLS algorithm in the series of experiments (10 realizations of the simulated dataset, level of noise 1000%). The solution of the RNPLS algorithm is adjusted in 15 iterations to the abrupt changes in observation (at *21st* iteration) with the forgetting factor $\gamma = 0.9$. B) "True" coefficients before after abrupt changes at *21st* iteration. C) Example of adjusting the regression coefficients over iterations $i = 20, \ldots, 35$.
doi:10.1371/journal.pone.0069962.g005

"True" coefficients (PLS without noise), the regression coefficients for RNPLS, NPLS, INPLS, and UPLS methods (averaged over 10 realizations, $\lambda = 10$) are shown in Figure 7. It can be seen that the result generated by RNPLS is closer to the "true" coefficients than the NPLS, INPLS, and UPLS solutions. To estimate the difference numerically, the mean values and standard deviations of $\rho\left(\mathbf{C}_{\lambda=0}^{\mathrm{true}}, \mathbf{C}_{\lambda>0}^{\mathrm{RNPLS}}\right)$ for different noise levels λ were calculated over the algorithm iterations (Figure 8). In parallel, the means and the standard deviations of $\rho\left(\mathbf{C}_{\lambda=0}^{\mathrm{true}}, \mathbf{C}_{\lambda>0}^{\mathrm{NPLS}}\right)$, $\rho\left(\mathbf{C}_{\lambda=0}^{\mathrm{true}}, \mathbf{C}_{\lambda>0}^{\mathrm{INPLS}}\right)$ and $\rho\left(\mathbf{C}_{\lambda=0}^{\mathrm{true}}, \mathbf{C}_{\lambda>0}^{\mathrm{unfolded}\ PLS}\right)$ were computed for every level of noise. All statistics were calculated for 10 realizations. As can be seen, the mean value of the distances between the RNPLS and "true" regression coefficients after 15 iterations (the first 20–100 points of the training set) is less than those between the NPLS, or UPLS and "true" regression coefficients computed for the whole training datasets (800 points), whereas RNPLS needs up to 14 iterations (280 points) to outperform the INPLS results. In addition, the standard deviations of the RNPLS results are generally less than those obtained by NPLS, INPLS, or UPLS.

The advantages of RNPLS can be explained by overfitting suppression.

Application

The particular goal of our study is to develop an efficient algorithm for BCI system calibration. The RNPLS algorithm was tested on real data sets recorded during BCI experiments.

Binary BCI

Both RNPLS and generic NPLS algorithms were applied in parallel to calibrate the binary BCI system. The data set was collected during the binary self-paced BCI experiments on a freely moving rat.

Table 1. Comparison of the optimal number of factors and RMSE for RNPLS, NPLS, INPLS, and UPLS.

	RNPLS		**NPLS**				**INPLS**				**UPLS**			
Noise, %	**Number of factors**	**RMSE, %**	**Number of factors**	**RMSE, %**	***t***	**$t_{0.01}$**	**Number of factors**	**RMSE, %**	***t***	**$t_{0.01}$**	**Number of factors**	**RMSE, %**	***t***	**$t_{0.01}$**
50	8	0.16±0.01	15	0.15±0.01	2.23	2.88	14	0.14±0.01	4.47	2.88	14	0.15±0.01	2.23	2.88
100	10	0.32±0.03	17	0.29±0.01	3.00	3.12	15	0.30±0.02	1.75	2.93	16	0.30±0.02	1.75	2.93
500	8	1.6±0.2	15	1.5±0.1	1.41	3.00	9	1.5±0.1	1.41	3.00	11	1.5±0.1	1.41	3.00
1000	9	3.1±0.2	17	2.8±0.1	4.24	3.00	17	3.0±0.3	0.88	2.93	13	2.9±0.2	2.24	2.88

The difference of RMSE for RNPLS and the other methods is not statistically significant ($t < t_{0.01}$) in the majority of the cases (10 from 12) according to the two-tailed Welch's *t*-test for 1% confidence level $t_{0.01}$.
doi:10.1371/journal.pone.0069962.t001

The experiments were carried out by the neurophysiological team at CEA, LETI, CLINATEC. Behavioral experiments were based on a reward-oriented task. A rat freely moving in a cage had the opportunity to push a pedal to activate a food dispenser. The BCI task consisted in detection (prediction) of the pushing event by analyzing the recording of the rat's neuronal activity (see details in [8]). During BCI experiments the Electrocorticogram (ECoG) (the signal from 14 electrodes located on the surface of the cortex) and the signal of the pedal were recorded simultaneously, down-sampled to 1.3 kHz and band-pass filtered between 0.5 Hz and 500 Hz. The Common Average Reference (CAR) filter was applied to eliminate a "common source" [27].

To form a feature tensor, each ECoG epoch was mapped to temporal-frequency-spatial space by the continuous wavelet transform (CWT). For each epoch j (determined by its final moment t), electrode c, frequency f and time shift τ, elements $x_{j,t,f,c}$ of the tensor $\underline{\mathbf{X}}$ were calculated as norm of CWT coefficients. The frequency band [10, 300] Hz with step $\delta f = 2$ Hz and sliding windows $[t - \Delta\tau,\ t]$, $\Delta\tau = 2$ s with step $\delta\tau = 0.04$ s, were considered for all electrodes $c = 1, 2, \ldots, 14$. The resulting dimension of a point was $(146 \times 51 \times 14)$. The binary dependent variable was set to one, $y_j = 1$, if the pedal was pressed at the moment t, and $y_j = 0$ otherwise.

For the model identification, the tensor of observations was formed from a 10 minute long recording. The tensor included 400 points corresponding to event-related epochs (50 pressing events repeated 8 times) and randomly selected 1000 non-event epochs. The tensor of observation was split into the training and test tensors. Namely, 1000 randomly selected points (700 "non-events" and 300 "events") formed the training dataset, while 400 points (300+100) were used for the test. The proportion of the classes is chosen to avoid a class imbalance problem (e.g. [28]). Generally, binary self-paced experiments [29] lead to imbalance classes: the "minority" class has a very small number of examples (class of events), while the "majority" class includes a large number of cases (class non-event). Most of the machine learning approaches are strongly biased toward the majority class (e.g. [28] while the correct classification of samples from the minority class is the most important. At the same time, the non-event class should be well presented for efficient algorithm learning. To avoid the problem, the classes were formed in a less biased way, following [8].

To test RNPLS, the training dataset was split into disjoint subsets with 10 and 100 points, marked as RNPLS (10) and RNPLS (100), respectively. Next, the projectors and the predictive models were identified. Figure 9A represents the first 2 factors calculated by RNPLS (10). The total number of factors was estimated to be 5 by the 10-fold cross-validation procedure. Each factor consists of 3 projectors, corresponding to frequency, temporal, and spatial modalities, respectively. Absolute values of the projector's elements define the influence of these elements on the identified model. The weights of factors in the decision rule are different. The relative weights of factors in the final decomposition (coefficients b_f^*, of the normalized model $\hat{\mathbf{y}} = \mathbf{T}\mathbf{b} + b_0\mathbf{1} = \mathbf{T}^*\mathbf{b}^* + b_0\mathbf{1}$, $\mathbf{t}_f^* = \mathbf{t}_f / \|\mathbf{t}_f\|$, $b_f^* = b_f\|\mathbf{t}_f\|$, $f = 1, 2, \ldots, F$) are demonstrated in Figure 9B. Figure 9C shows the summarized influence of the elements for different modalities on the predictive model according to Modality Influence (MI) analysis [30]. For instance, in the spatial modality the electrodes number 15 and 8 have the highest impact on the decision rule. In the frequency modality the main contribution is given by the frequencies band [150, 250] Hz. The most significant interval in the temporal domain is [−0.4, −0.1] s. Figure 10 shows predicted root mean squared errors (RMSE) as a function of the number of factors. With respect to NPLS, the RNPLS algorithm demonstrates minimal deterioration in the prediction quality for all numbers of factors: for RNPLS (100) it is about 0.1%, while for RNPLS (10) it is about 0.2%. Thus, the experiments demonstrated that the size of the analyzed data block could be significantly reduced without essential deterioration of the prediction quality.

Continuous BCI

The RNPLS method was also applied for decoding of the continuous three-dimensional hand trajectories from epidural ECoG signals of a Japanese macaque. The data is publicly available (http://neurotycho.org/data/20100802s1epidural-ecogfood-trackingbkentaroshimoda). The ECoG signals were recorded (Blackrock Microsystems, Salt Lake City, UT, USA) with a sampling rate of 1 kHz from the 64 electrodes implanted in the epidural space of the left hemisphere of the monkey, which was trained to retrieve foods using the right hand. The hand motion was recorded by an optical motion capture system (Vicon Motion Systems, Oxford, UK) with a sampling rate 120 Hz. The length of the experiments was 15 minutes. During the experiment, the monkey was seated in the chair with restricted head movement. The experimenter demonstrated foods at random locations at a distance of 20 cm for the monkey at random time intervals (3.8±1.0 times per minute), and the monkey grasped the foods. A precise description of the experiments can be found in [31]. In this work, the unfolded PLS was applied for the continuous three-dimensional hand trajectories reconstruction.

To test the RNPLS algorithm one recording (15 minutes, sampling rate: 1000 Hz) was chosen randomly from the data base. To train the algorithm, 5000 time epochs were randomly selected among the recorded time moments. As a result, the training dataset includes 0.5% of the entire recording. The test dataset

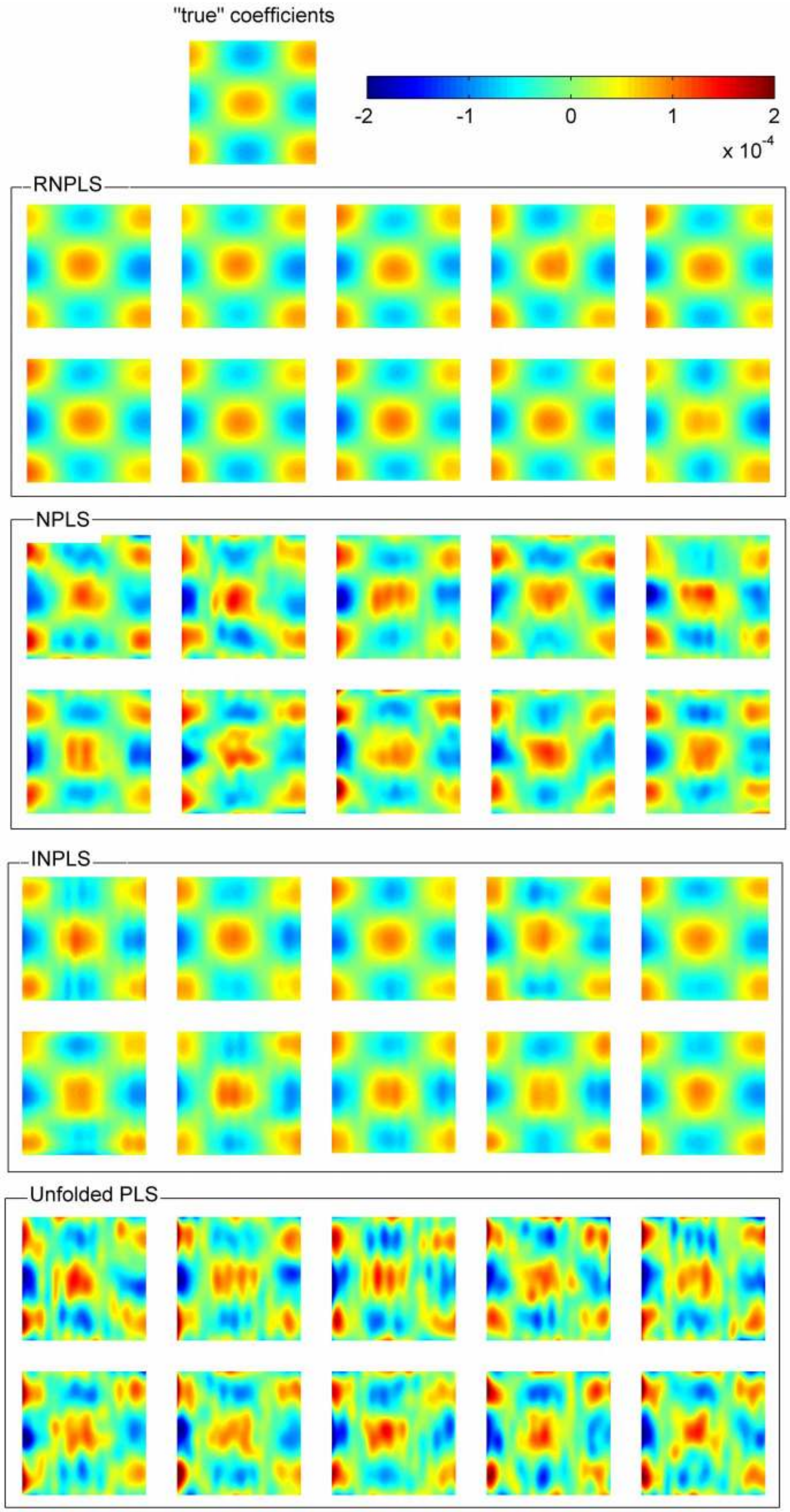

Figure 6. Examples of regression coefficients for different methods. RNPLS, NPLS, INPLS and UPLS regression coefficients for 10 realizations of the simulated training dataset (level of noise 1000%).
doi:10.1371/journal.pone.0069962.g006

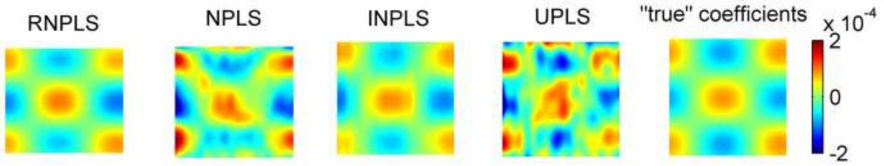

Figure 7. The RNPLS, NPLS, INPLS, UPLS and ''true'' regression coefficients. Comparison of the regression coefficients averaged over 10 realizations of noise in the simulated dataset (level of noise 1000%). RNPLS, NPLS, INPLS, UPLS and "true" coefficients.
doi:10.1371/journal.pone.0069962.g007

contains 4500 random epochs (0.45% of the entire recording) of the same file out of training.

To form a feature tensor, each ECoG epoch was mapped to temporal-frequency-spatial space by CWT. The frequency band consisted of 3 sub-bands, namely, [0.6, 7.8] Hz with step $\delta f = 0.2$ Hz, [8, 48] Hz with step $\delta f = 2$ Hz, and [50, 300] Hz with step $\delta f = 10$ Hz. Sliding windows $[t - \Delta\tau, t]$, $\Delta\tau = 1$ s with step $\delta\tau = 0.005$ s were considered for all electrodes $c = 1, 2, ..., 64$. The resulting dimension of a point is $(84 \times 201 \times 64) \approx 10^6$ vs. $(10 \times 10 \times 64) \approx 6400$ in [31] (frequency band Hz, $\delta f = 10$ Hz, time interval $\Delta\tau = 1$ s, $\delta\tau = 0.1$ s). Since the RNPLS algorithm allows recursive data set processing, the restriction on the memory consumption is less limiting.

Preprocessing techniques (chewing artifacts extraction, common average reference filter, etc.) were not applied. To identify the decoding model with RNPLS, the training set was split into 5 subsets of 1000 points. To validate the generalization ability, the model was applied to the test data set. Predicted wrist motion of the right hand was correlated with its real position. The resulting correlation for the test data as a function of the number of factors is shown in Figure 11. The best correlations 0.69, 0.82, and 0.79 for X-, Y-, and Z-positions are achieved for 150–175 factors. An example of the real and predicted trajectories is given in Figure 12.

To test the RNPLS approach in the general case of tensor input and tensor output data, we applied the proposed method for simultaneous prediction of the monkey's right shoulder, right elbow, and right wrist coordinates. Both the training and the test input tensors $\underline{\mathbf{X}}^{\text{training}}$ and $\underline{\mathbf{X}}^{\text{test}}$ (calculated from ECoG) were not changed, whereas the output tensors $\underline{\mathbf{Y}}^{\text{training}}$ and $\underline{\mathbf{Y}}^{\text{test}}$ acquired one more modality (shoulder, elbow, wrist). Thus, the training dataset was $\left\{\underline{\mathbf{X}}_k^{\text{training}}, \underline{\mathbf{Y}}_k^{\text{training}}\right\}_{k=1}^{5}$, $\underline{\mathbf{X}}_k^{\text{training}} \in \Re^{1000 \times 84 \times 201 \times 64}$, $\underline{\mathbf{Y}}_k^{\text{training}} \in \Re^{1000 \times 3 \times 3}$ and the test dataset was $\{\underline{\mathbf{X}}^{\text{test}}, \underline{\mathbf{Y}}^{\text{test}}\}$, $\underline{\mathbf{X}}^{\text{test}} \in \Re^{4500 \times 84 \times 201 \times 64}$, $\underline{\mathbf{Y}}^{\text{test}} \in \Re^{4500 \times 3 \times 3}$. The resulting correla-

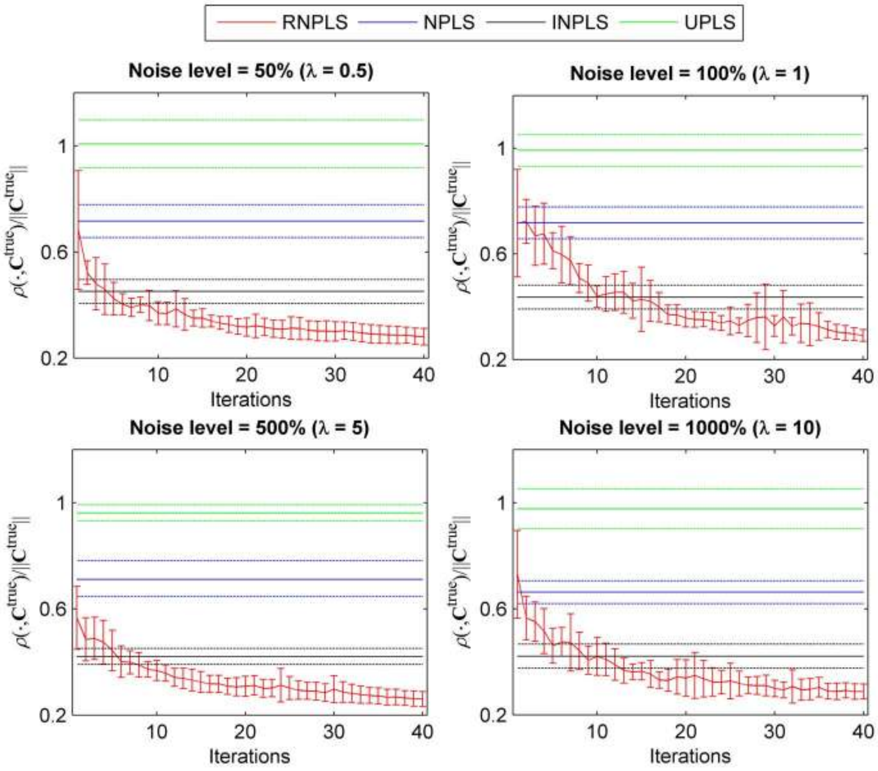

Figure 8. Comparison of regression coefficients for different levels of noise. The mean values and the standard deviations (10 realizations of random noise in the simulated dataset) of the distance between the "true" regression coefficients and RNPLS regression coefficients $\rho\left(\mathbf{C}^{true}, \mathbf{C}^{RNPLS}\right)$ over the RNPLS recursive iterations (red lines); between the "true" coefficients and regression coefficients generated by the NPLS applied to whole training datasets $\rho\left(\mathbf{C}^{true}, \mathbf{C}^{\text{NPLS}}\right)$ (blue lines); between the ''true'' coefficients and regression coefficients generated by INPLS (black lines) and by UPLS (green lines).
doi:10.1371/journal.pone.0069962.g008

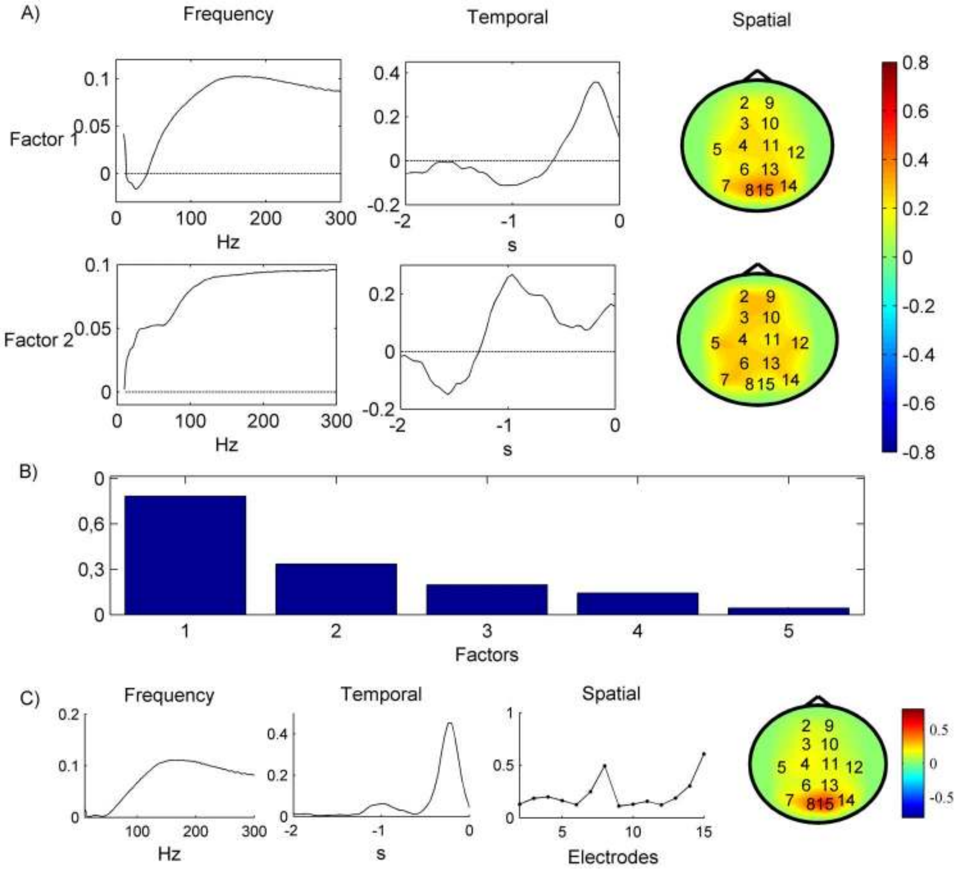

Figure 9. RNPLS (10) calibration. A) The first and the second factors (which are the most contributive out of 5): frequency, temporal and spatial projections. The values of elements of the spatial projector are shown in colors according to the color bar and positions of the electrodes are indicated by numbers. B) Factor weights in the final decomposition which are the coefficients b_i^* of the normalized model in the space of latent variables. C) Impact on the predictive model of different modalities' components according to MI analysis; the spatial modality is represented by the graph and the corresponding color map. Recordings from binary self-paced BCI experiments in freely moving rat**.**
doi:10.1371/journal.pone.0069962.g009

tions between the observed and predicted coordinates are represented in Table 2. The number of factors was equal to 200. Compared with the results obtained for the wrist only, we found that predictions of the Y- and Z-coordinates are improved while the X-coordinate became slightly worse.

In general, the quality of the prediction of the X-coordinates (left/right movement) for all analyzed parts of the right hand was essentially worse than prediction of Y-coordinates (backward/forward movement) and Z-coordinates (up/down movement). A possible explanation of this result is different types of muscular efforts, and therefore different levels of their representation in the

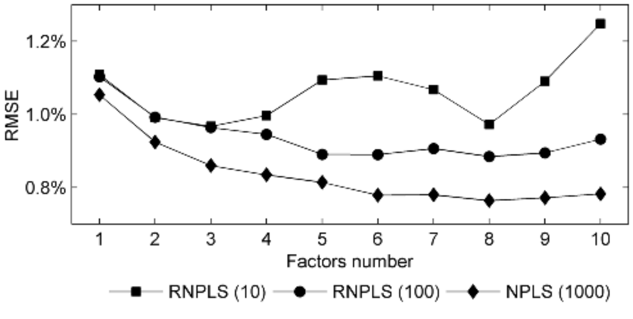

Figure 10. Comparison of prediction error RNPLS vs. NPLS. RMSE as a function of the number of factors for the test dataset. RNPLS (10) – the training set is split into 10-point disjoint subsets, RNPLS (100) – the training dataset is split into 100-point disjoint subsets, NPLS (1000) – generic NPLS using the whole training dataset. Recordings from binary self-paced BCI experiments in freely moving rat**.**
doi:10.1371/journal.pone.0069962.g010

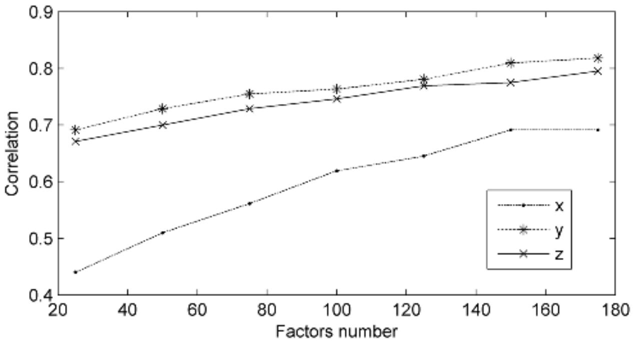

Figure 11. Correlation between the real and predicted coordinates. Correlation between the real and predicted coordinates as a function of the number of RNPLS factors for the test data of 3D wrist position of the monkey's right hand.
doi:10.1371/journal.pone.0069962.g011

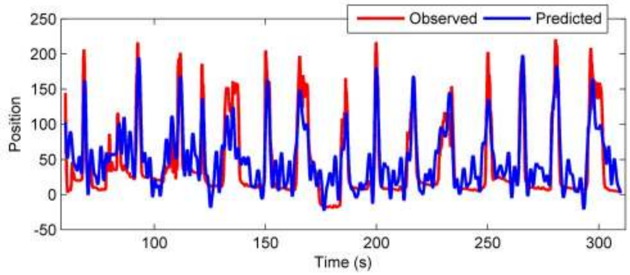

Figure 12. Example of the observed and predicted trajectories. Example of the observed and predicted trajectories for Y-coordinate of the monkey's right wrist. The predicted model is identified by the RNPLS approach applied for the case of tensor input and matrix output of 3D coordinates for wrist position.
doi:10.1371/journal.pone.0069962.g012

Table 2. Correlation between predicted and observed coordinates of the monkey's right hand (test dataset, N = 4500 samples).

	$corr_x$	t	$corr_y$	t	$Corr_z$	t
Shoulder	0.62	85.67	0.80	134.13	0.85	159.65
Elbow	0.54	72.67	0.84	153.67	0.83	148.19
Wrist	0.63	87.51	0.85	159.65	0.82	143.15

t-test demonstrates the significance of the correlation (1% confidence level $t_{0.01} = 2.58$).
doi:10.1371/journal.pone.0069962.t002

ECoG recordings.

Discussion

Neuronal signal decoding represents a challenging task. Recent promising results in the BCI area (e.g., [32–36]) have brought clinical applications to a realistic task. For real-life applications, the need for an easy to use BCI system is one of the crucial problems. The recursive algorithm presented in this article contributes towards solving the problem of adaptive fast and easy BCI system calibration.

The proposed RNPLS performs multimodal data analysis. While several tensor-based methods were recently efficiently applied in BCI studies, all of them are offline methods and require access to the entire data set. RNPLS unites the recursive calculation scheme with multimodal data representation and can be applied online. It uses a single-pass blockwise tensor-data processing in contrast to multipass blockwise Iterative NPLS [8], which runs over the entire recording to estimate each latent variable and corresponding set of projectors. Moreover, RNPLS allows adaptive learning by introducing the forgetting factor.

One of the important problems of sequential algorithms could be the degradation of results caused by consecutive calculation. To examine this question, the prediction accuracy of the proposed RNPLS algorithm was studied and compared with generic offline NPLS as well as two other algorithms of the PLS family (INPLS and Unfolded PLS) on artificial and real data sets. In both cases the generalization abilities of tested algorithms are comparable. We have not observed significant degradation in prediction error caused by recursive calculation.

At the same time, over the set of computational experiments recursive blockwise RNPLS provides the best approximation of the "true" model coefficients and with minimal variance. It is interesting to note that although coefficient estimates of all algorithms of the PLS family are biased, both blockwise INPLS and RNPLS provide more stable and less biased coefficients in comparison to NPLS and Unfolded PLS. While the "true" model is not necessarily the best in the sense of expectation of prediction error (e.g., [37]), the appropriate coefficient estimation is nevertheless of great importance for the interpretation of results.

Finally, the RNPLS approach demonstrates fast coefficients convergence for all tested levels of noise and is considerably noise-steady.

The recursive algorithm was compared to the generic NPLS in terms of accuracy and convergence rate on the real data of BCI experiments. In binary self-paced BCI in freely moving animals the RNPLS algorithm demonstrated minimal deterioration in the prediction quality with respect to NPLS: 0.2% using the 10 point blockwise algorithm and 0.1% using the 100 point blockwise algorithm. Treatment of smaller blocks of data does not significantly affect the generalization ability of the model. At the same time, a lower number of relevant factors was extracted by RNPLS. The 10-fold cross-validation procedure indicated 8 as the optimal number of factors for generic NPLS and only 5 factors for RNPLS. Nevertheless, taking into account the requirements for computation resources (memory), the blockwise method is favorable. The factors extracted by RNPLS algorithms are very similar to the factors reported in [8]. The same electrode has the greatest impact on the decision rule (~16% of the extracted information for the generic approach and ~19% for the recursive algorithm). In both cases, high frequencies [100, 300] Hz provided the main contribution to the decision (~86% and ~91%). In the time domain, the interval [−0.5, 0] s before the event is the most significant (~68% and ~78% of the influence). Let us note that less informative variables are better suppressed by the recursive algorithm. Namely, the time interval of more than 1 s before a movement is almost completely inhibited by RNPLS, which better corresponds to the physiology. Moreover, the computational efficiency of the resultant model is acceptable for real-time applications [8].

Another real-life problem in testing the recursive algorithm was the decoding of the continuous three-dimensional hand trajectory from epidural ECoG signals. The movement of the hand was represented by 3D positions of the shoulder, the elbow, and the wrist of the monkey. Thus, the motion was characterized by 9 degrees of freedom. Rigorous comparison of the proposed approach with results reported by other authors is complicated by variation of testing data. Nevertheless, the accuracy of prediction (correlations 0.63, 0.85, 0.82 for X-, Y-,Z- positions of wrist) is comparable with results reported by other authors using the same or a similar data base: 0.47±0.12, 0.56±0.10, 0.68±0.06 with Unfold-PLS [31]; 0.52±0.13, 0.67±0.04, 0.74±0.04 with Higher-Order Partial Least Squares [3,35]; 0.50±0.13, 0.67±0.04, 0.73±0.04 with NPLS [35]; 0.50±0.12, 0.67±0.04, 0.74±0.03 with Unfold-PLS [35]. The good accuracy may result from high resolution of data analyses which is achieved due to consecutive calculation. The large number of factors (150–175 vs. 36±14 in [31]) can be explained by the fact that presentation of data by tensors of order one (used in NPLS algorithms) required more factors than matrix decomposition with PLS. The continuous structure of the output signal could contribute to the significant number of factors. For instance, for the case of the binary response variables, the number of factors for the RNPLS approach did not exceed 10.

Limitations of the Present Study and Future Work

The RNPLS algorithm applies a blockwise recursive calculation schema of Qin's RPLS algorithm and inherits its limitations. From the theory of Qin's recursive PLS, the number of factors F should provide a full rank of matrix of latent variables (the residuals of tensor $\underline{\mathbf{X}}$ factorization should be essentially zero: $\left\|\mathbf{E}_F\right\| \le \varepsilon$, see [18]). Otherwise it could result in numerical error. RNPLS inherits this point of Qin's algorithm. Similar to RPLS, the number of factors in Recursive NPLS should be large enough to extract the essential information. It could be chosen with cross-validation [18].

Some informative features could be weakly represented and finally lost due to insufficient size of blocks. To prevent this situation, the size of the data fragments should be adjusted depending on the number of informative factors.

In addition, the forgetting factor should be determined in accordance with the variability of the analyzed data.

The perspective of this study is preclinical (in the monkey) and clinical BCI applications. Simulation of decoding of neuronal activity demonstrates the compatibility of the predictive model with real time requirements. After efficient implementation, the algorithm will be used in the CLINATEC® BCI project which includes the realization of a fully implantable device, WIMAGINE®, to measure and transmit ECoG data wireless [38] to a terminal, and the means for a tetraplegic subject to pilot effectors, such as fragments of exoskeleton. Blockwise RNPLS allows higher resolution (frequency and temporal) of the BCI predictive model than other approaches including tensor based methods [31,35]. In addition, it will help to take into account session-to-session BCI signal variability with fast system adjustment instead of full recalibration and to consider intra-session signal variations (degradation due to fatigue, changes caused by brain plasticity, disturbance from the outside world, etc.) The study of the algorithm's adaptation to session-to-session, subject-to-subject variability with publicly available monkey ECoG recordings is our nearest perspective.

Conclusion

In the current paper we consider the method of the adaptive multimodal data flow processing in the most general case of tensor-input/tensor-output data, whose properties were studied in detail for different aspects. A detailed pseudocode is provided which substantially facilitates the understanding and implementation of the proposed approach.

Contrary to other multimodal data treatment methods, such as unfolded PLS, NPLS, INPLS, etc., the RNPLS allows sequential blockwise adaptive learning. Due to significant variability of the neural data, this property is of great importance for the real-time BCI applications, where complete recalibration of the system is time-consuming and cannot be done frequently.

For the obvious demonstration of the convergence properties, they were explored on the artificial datasets. The experiment demonstrated fast and stable convergence. Tested with noisy data (the level of the noise was up to 1000%), the algorithm showed robustness and significant noise steadiness.

At the same time, the performance of the sequential algorithm is demonstrated to be comparable with the offline algorithms of the PLS family. In both simulated and real experiments, RNPLS allows one-pass online regression reconstruction without loss of generality. Note that all PLS family algorithms provide biased estimates of regression coefficients [39]. However, the recursive algorithm demonstrated better stability, lower variability and less biased coefficients, regarding the "true" ones.

RNPLS method demonstrates promising results in BCI studies. Generalization of the algorithm to the case of tensor-input/tensor-output data allows efficient continuous movement reconstruction in several points (e.g. wrist, elbow, shoulder) simultaneously.

The algorithm can be efficient in many other applications for the adaptive modeling of high dimensional tensor flows.

Supporting Information

Algorithm S1 RNPLS.
(DOC)

Acknowledgments

The authors are grateful to Prof. A.-L. Benabid, Drs. A. Wyss, C. Moro, N. Torres, T. Costecalde, G. Charvet, F. Sauter, J. Porcherot, A. Serpollet and C. Mestais, as well as to all members of the CEA, LETI, CLINATEC team, who organized and realized the experiments as well as providing the recordings used in these studies.

Author Contributions

Conceived and designed the experiments: AE TA. Performed the experiments: AE TA. Analyzed the data: AE TA. Wrote the paper: AE TA.

References

1. Nazarpour K, Sanei S, Shoker L, Chambers JA (2006) Parallel space-time-frequency decomposition of EEG signals for brain computer interfacing. Proc. 14th European Signal Processing Conf. (EUSIPCO 2006).
2. Zhao Q, Caiafa CF, Cichocki A, Zhang L, Phan AH (2009) Slice oriented tensor decomposition of EEG data for feature extraction in space, frequency and time domains. Neural Information Processing: 16th Int'l Conf. (ICONIP 2009), LNCS 5863, 221–228.
3. Zhao Q, Caiafa CF, Mandic DP, Zhang L, Ball T, et al. (2011) Multilinear subspace regression: An orthogonal tensor decomposition approach. NIPS 2011: 1269–1277.
4. Chao ZC, Nagasaka Y, Fujii N (2010) Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkeys. Frontiers in Neuroengineering 3 (3): 1–10.
5. Phan AH, Cichocki A, Vu-Dinh T (2010) A tensorial approach to single trial recognition for brain computer interface. Proc. 2010 Int'l Conf. Advanced Technologies for Comm., (ATC 2010), 138–141.
6. Li J, Zhang L, Tao D, Sun H, Zhao Q (2009) A prior neurophysiologic knowledge free tensor-based scheme for single trial EEG classification. IEEE Trans. Neural Systems and Rehabilitation Eng. 17 (2): 107–115.
7. Li J, Zhang L (2010) Regularized tensor discriminant analysis for single trial EEG classification in BCI. Pattern Recognition Letters 31 (7): 619–628.
8. Eliseyev A, Moro C, Costecalde T, Torres N, Gharbi S, et al. (2011) Iterative N-way Partial Least Squares for a binary self-paced brain-computer interface in freely moving animals. J. Neural Eng. 8 (4) 046012: 1–14.
9. Martínez-Montes E, Valdés-Sosa PA, Miwakeichi F, Goldman RI, Cohen MS (2004) Concurrent EEG/fMRI analysis by multiway Partial Least Squares. J. Neuroimage 22 (3): 1023–1034.
10. Harshman RA (1970) Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. UCLA Working Papers in Phonetics (Ann Arbor: University Microfilms) 16: 84. No. 10,085.
11. Tucker LR (1966) Some mathematical notes on three-mode factor analysis. Psychometrika 31 (3): 279–311. doi: 10.1007/BF02289464.
12. Carroll JD, De Soete G; Pruzansky S (1989) Fitting of the latent class model via iteratively reweighted least squares CANDECOMP with nonnegativity constraints. In: Coppi R, Bolasco S, editors, Multiway data analysis. Elsevier, Amsterdam, The Netherlands. 463–472.
13. Mørup M, Hansen LK, Arnfred SM (2008) Algorithms for sparse nonnegative Tucker decomposition. Neural Computation 20: 2112–2131.
14. Tao D, Li X, Wu X, Maybank SJ (2007), General tensor discriminant analysis and Gabor features for gait recognition. IEEE Trans. Pattern Anal. Mach. Intell. 29 (10): 1700–1715.
15. Bro R (1996) Multiway Calibration. Multilinear PLS. J. Chemometrics 10 (1): 47–61.

16. Bro R (1998) Multi-way analysis in the food industry: Models, algorithms and applications. PhD Thesis. University of Amsterdam (NL) and Royal Veterinary and Agricultural University (DK).
17. Geladi P, Kowalski BR (1986) Partial least-squares regression: A Tutorial. Analytica Chimica Acta 185: 1–17.
18. Qin S J (1998) Recursive PLS algorithms for adaptive data modelling. Computers & Chemical Eng. 22 (4–5): 503–514.
19. Eliseyev A, Benabid A L, Aksenova T (2011) Recursive multi-way PLS for adaptive calibration of brain computer interface system. Lecture Notes in Computer Science, 6792/2011: 17–24.
20. Kolda TG, Bader BW (2009) Tensor decompositions and applications. SIAM Review 51 (3): 455–500.
21. Helland K, Berntsen HE, Borgen OS, Martens H (1992) Recursive algorithm for partial least squares regression. Chemometrics & Intelligent Laboratory Systems 14 (1–3): 129–137.
22. Dayal BS, MacGregor JF (1997) Recursive exponentially weighted PLS and its applications to adaptive control and prediction. J. Process Control 7 (3): 169–179.
23. Wang X, Kruger U, Lennox B (2003) Recursive partial least squares algorithms for monitoring complex industrial processes. Control Eng. Practice 11 (6): 613–632.
24. Vijaysai P, Gudi RD, Lakshminarayanan S (2003) Identification on demand using a blockwise recursive partial least-squares technique. Industrial & Eng. Chemistry Research 42 (3): 540–554.
25. Lee HW, Lee MW, Park JM (2007) Robust adaptive partial least squares modeling of a full-scale industrial wastewater treatment process. Industrial & Eng. Chemistry Research 46 (3): 955–964.
26. Golub GH, van Loan CF (1989) Matrix Computations. (2nd edition), Baltimore, MD: The Johns Hopkins University Press.
27. Ludwig KA, Miriani RM, Langhals NB, Joseph MD, Anderson DJ, et al. (2009) Using a common average reference to improve cortical neuron recordings from microelectrode arrays. J. Neurophysiology 101 (3): 1679–1689.
28. Japkowicz N, Stephen S (2002) The class imbalance problem: A systematic study. Intell. Data Anal. 6 (5): 429–449.
29. Mason SG, Birch GE (2000) A brain-controlled switch for asynchronous control applications. IEEE Trans. Biomed. Eng. 47: 1297–1307.
30. Cook RD, Weisberg S (1982) Residuals and influence in regression. London: Chapman and Hall.
31. Shimoda K, Nagasaka Y, Chao ZC, Fujii N (2012) Decoding continuous three-dimensional hand trajectories from epidural electrocorticographic signals in Japanese macaques. J. Neural Eng. 9: 036015.
32. Wolpaw JR, Birbaumer N, McFarland DJ, Pfurtscheller G, Vaughan TM (2002) Brain-computer interfaces for communication and control. Clinical Neurophysiology 113 (6): 767–791.
33. Velliste M, Perel S, Spalding MC, Whitford AS, Schwartz AB (2008) Cortical control of a prosthetic arm for self-feeding. Nature 453: 1098–1101.
34. Müller-Putz GR, Kaiser V, Solis-Escalante T, Pfurtscheller G (2010) Fast set-up asynchronous brain-switch based on detection of foot motor imagery in 1-channel EEG. Medical & Biological Eng. & Computing 48 (3): 229–233.
35. Zhao Q, Caiafa CF, Mandic DP, Chao ZC, Nagasaka Y, et al. (2012) Higher-order partial least squares (HOPLS): A generalized multi-linear regression method. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI). In press.
36. Hochberg LR, Bacher D, Jarosiewicz B, Masse NY, Simeral JD, et al. (2012) Reach and grasp by people with tetraplegia using a neurally controlled robotic arm. Nature 485 (7398): 372–375.
37. Aksenova TI, Jurachkovskiy YP (1988) Characterisation at unbiased structure and conditions of their J-optimality. J. Optimality, Soviet Journal of Automation and Information Science 21 (4): 36–42.
38. Foerster M, Porcherot J, Bonnet S, Van Langhenhove A, Robinet S, et al. (2012, November) Integration of a state of the art ECoG recording ASIC into a fully implantable electronic environment. In: Biomedical Circuits and Systems Conference (BioCAS), 2012 IEEE (pp. 232–235). IEEE.
39. Helland IS (2001) Some theoretical aspects of partial least squares regression. Chemometrics and Intelligent Laboratory Systems 58 (2): 97–107.