Dimensionality reduction for time series decoding and forecasting problems*

R. V. Isachenko[1], M. R. Vladimirova[2], V. V. Strijov[3]

Abstract: The paper is devoted to the problem of detecting the relation between independent and target variables. We propose to predict a multidimensional target vector instead of predicting one timestamp point. We consider the linear model of partial least squares (PLS). The method finds the matrix of a joint description for the design matrix and the outcome matrix. The description is low-dimensional and allows to build a simple, stable model. We conducted computational experiments on the real data of energy consumption and electrocorticograms signals (ECoG).

Keywords: time series decoding, forecast, partial least squares, dimensionality reduction

1 Introduction

The paper investigates the problem of dependence recovering between an input data and a model outcome. The proposed model is suitable for predicting a multidimensional target variable. In the case of the forecasting problem objects and target spaces have the same nature. To build the model we need to construct autoregressive matrices for input objects and target variables. The object is a local signal history, the outcome is signal values in the next timestamps. An autoregressive model makes a consumption that the current signal values depend linearly on the previous signal values.

In the case of time series decoding problem objects and target spaces are different in nature, the outcome is a system response to the input signal. The autoregressive design matrix contains the local history of the input signal. The autoregressive target matrix contains the local history of the response.

*The work was financially supported by the Russian Foundation for Basic Research (project 16-07-01155).

[1]Moscow Institute of Physics and Technology, isa-ro@yandex.ru

[2]Moscow Institute of Physics and Technology, vladimirova.maria@phystech.edu

[3]A. A. Dorodnicyn Computing Centre, Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, strijov@ccas.ru

The object space in time series decoding problems is high dimensional. Excessive dimensionality of the feature description leads to instability of the model. To solve this problem the feature selection procedures are used [1, 2].

The paper considers the partial least squares regression (PLS) model [3–5]. The PLS model reduces the dimensionality of the input data and extracts the linear combination of features which have the greatest impact on the response vector. Feature extraction is an iterative process in order of decreasing the influence on the response vector. PLS regression methods are described in detail in [6–8]. The difference between various PLS approaches, different kinds of the PLS regression could be found in [9].

The current state of the field and the overview of nonlinear PLS method modifications are described in [10]. A nonlinear PLS method extension was introduced in [11]. There has been developed the variety of PLS modifications. The proposed nonlinear PLS methods are based on smoothing splines [12], neural networks [13], radial basis functions [14], genetic algorithms [15].

The result of the feature selection is the dimensionality reduction and the increasing model stability without significant loss of the prediction quality. The proposed method is used on two datasets with the redundant input and target spaces. The first dataset consists of hourly time series of energy consumption. Time series were collected in Poland from 1999 to 2004.

The second dataset comes from the NeuroTycho project [16] that designs Brain-Computer съехало почему-то Interface (BCI) [17, 18] for information transmitting between brains and electronic devices. Brain-Computer Interface (BCI) system enhances its user's mental and physical abilities, providing a direct communication mean between the brain and a computer [19]. BCIs aim at restoring damaged functionality of motorically or cognitively impaired patients. The goal of motor imagery analysis is to recognize intended movements from the recorded brain activity. While there are various techniques for measuring cortical data for BCI [20, 21], we concentrate on the ElectroCorticoGraphic (ECoG) signals [22]. ECoG, as well as other invasive techniques, provides more stable recordings and better resolution in temporal and spatial domains than its non-invasive counterparts. We address the problem of continuous hand trajectory reconstruction. The subdural ECoG signals are measured across 32 channels as the subject is moving its hand. Once the ECoG signals are transformed into informative features, the problem of trajectory reconstruction is the autoregression problem. Feature extraction involves application of some spectro-temporal transform to the ECoG signals from each channel [23].

In papers, which are devoted to forecasting of complex spatial time series, the forecast is built pointwise [24,25]. If one need to predict multiple points simultaneously, it is proposed to compute forecasted points sequentially. During this process the previous predicted values are used to obtain a subsequent ones. The proposed method allows to obtain multiple predicted time series values at the same time taking into account hidden dependencies not only in the object space, but also in the target space. The proposed method significantly reduces the dimensionality of the feature space.

2 Problem statement

Given a dataset $\mathfrak{D} = (\mathbf{X}, \mathbf{Y})$, where $\mathbf{X} \in \mathbb{R}^{m \times n}$ is a design matrix, $\mathbf{Y} \in \mathbb{R}^{m \times r}$ is a target matrix. The examples of how to construct the dataset for a particular application task described in the Computational experiment.

We assume that there is a linear dependence between the objects $\mathbf{x} \in \mathbb{R}^n$ and the responses $\mathbf{y} \in \mathbb{R}^r$

$$\underset{1\times r}{\mathbf{y}} = \underset{1\times n}{\mathbf{x}} \cdot \underset{n\times r}{\mathbf{\Theta}} + \underset{1\times r}{\boldsymbol{\varepsilon}}, \tag{1}$$

where $\mathbf{\Theta}$ is the matrix of model parameters, $\boldsymbol{\varepsilon}$ is the vector of residuals.

The task is to find the matrix of the model parameters $\mathbf{\Theta}$ given the dataset $\mathfrak{D}$. The optimal parameters are determined by error function minimization. Define the quadratic error function S for the dataset $\mathfrak{D}$:

$$S(\mathbf{\Theta}|\mathfrak{D}) = \left\| \underset{m\times n}{\mathbf{X}} \cdot \underset{n\times r}{\mathbf{\Theta}} - \underset{m\times r}{\mathbf{Y}} \right\|_2^2 = \sum_{i=1}^{m} \left\| \underset{1\times n}{\mathbf{x}_i} \cdot \underset{n\times r}{\mathbf{\Theta}} - \underset{1\times r}{\mathbf{y}_i} \right\|_2^2 \to \min_{\mathbf{\Theta}}. \tag{2}$$

The linear dependence of the columns of the matrix X leads to an instable solution for the optimization problem (2). To avoid the strong linear dependence one could use feature selection techniques.

3 Partial Least Squares method

To eliminate the linear dependence and reduce the dimensionality of the input space, the principal components analysis (PCA) is widely used. The main disadvantage of the PCA method is its insensitivity to the interrelation between the objects and the responses. The partial least squares (PLS) algorithm projects the design matrix $\mathbf{X}$ and the target matrix to the latent space $\mathbb{R}^l$ with low dimensionality ($l < r < n$). The PLS algorithm finds the latent space matrix $\mathbf{T} \in \mathbb{R}^{m \times l}$ that best describes the original matrices $\mathbf{X}$ and $\mathbf{Y}$.

The disign matrix $\mathbf{X}$ and the target matrix $\mathbf{Y}$ are projected into the latent space in the following way:

$$\underset{m\times n}{\mathbf{X}} = \underset{m\times l}{\mathbf{T}} \cdot \underset{l\times n}{\mathbf{P}^{\mathsf{T}}} + \underset{m\times n}{\mathbf{F}} = \sum_{k=1}^{l} \underset{m\times 1}{\mathbf{t}_k} \cdot \underset{1\times n}{\mathbf{p}_k^{\mathsf{T}}} + \underset{m\times n}{\mathbf{F}}, \tag{3}$$

$$\underset{m\times r}{\mathbf{Y}} = \underset{m\times l}{\mathbf{T}} \cdot \underset{l\times r}{\mathbf{Q}^{\mathsf{T}}} + \underset{m\times r}{\mathbf{E}} = \sum_{k=1}^{l} \underset{m\times 1}{\mathbf{t}_k} \cdot \underset{1\times r}{\mathbf{q}_k^{\mathsf{T}}} + \underset{m\times r}{\mathbf{E}}, \tag{4}$$

where $\mathbf{T}$ is a matrix of a joint description of the objects and the outcomes in the latent space, and the columns of the matrix $\mathbf{T}$ are orthogonal; $\mathbf{P}$, $\mathbf{Q}$ are transition matrices from the latent space to the original space; $\mathbf{E}$, $\mathbf{F}$ are residual matrices.

The pseudocode of the PLS regression algorithm is given in the algorithm 1. In each of the l steps the algorithm iteratively calculates columns $\mathbf{t}_k$, $\mathbf{p}_k$, $\mathbf{q}_k$ of the matrices $\mathbf{T}$, $\mathbf{P}$, $\mathbf{Q}$, respectively. After the computation of the next set of vectors, the one-rank approximations are subtracted from the matrices $\mathbf{X}$, $\mathbf{Y}$. This step is called a matrix deflation. In the first step one has to normalize the columns of the original matrices (subtract the mean and divide by the standard deviation). During the test mode we need to normalize the test data, compute the model prediction (1), and then perform the reverse normalization.

Algorithm 1 PLSR algorithm

Require: $\mathbf{X}, \mathbf{Y}, l$;
Ensure: $\mathbf{T}, \mathbf{P}, \mathbf{Q}$;
1: normalize matrices $\mathbf{X}$ и $\mathbf{Y}$ by columns
2: initialize $\mathbf{u}_0$ (the first column of $\mathbf{Y}$)
3: $\mathbf{X}_1 = \mathbf{X}; \mathbf{Y}_1 = \mathbf{Y}$
4: **for** $k = 1, \dots, l$ **do**
5:     **repeat**
6:         $\mathbf{w}_k := \mathbf{X}_k^\mathsf{T}\mathbf{u}_{k-1}/(\mathbf{u}_{k-1}^\mathsf{T}\mathbf{u}_{k-1}); \quad \mathbf{w}_k := \frac{\mathbf{w}_k}{\|\mathbf{w}_k\|}$
7:         $\mathbf{t}_k := \mathbf{X}_k\mathbf{w}_k$
8:         $\mathbf{c}_k := \mathbf{Y}_k^\mathsf{T}\mathbf{t}_k/(\mathbf{t}_k^\mathsf{T}\mathbf{t}_k); \quad \mathbf{c}_k := \frac{\mathbf{c}_k}{\|\mathbf{c}_k\|}$
9:         $\mathbf{u}_k := \mathbf{Y}_k\mathbf{c}_k$
10:     **until** $\mathbf{t}_k$ stabilizes
11:     $\mathbf{p}_k := \mathbf{X}_k^\mathsf{T}\mathbf{t}_k/(\mathbf{t}_k^\mathsf{T}\mathbf{t}_k), \ \mathbf{q}_k := \mathbf{Y}_k^\mathsf{T}\mathbf{t}_k/(\mathbf{t}_k^\mathsf{T}\mathbf{t}_k)$
12:     $\mathbf{X}_{k+1} := \mathbf{X}_k - \mathbf{t}_k\mathbf{p}_k^\mathsf{T}$
13:     $\mathbf{Y}_{k+1} := \mathbf{Y}_k - \mathbf{t}_k\mathbf{q}_k^\mathsf{T}$

The vectors $\mathbf{t}_k$ and $\mathbf{u}_k$ from the inner loop of the algorithm 1 contain information about the object matrix $\mathbf{X}$ and the outcome matrix $\mathbf{Y}$, respectively. The blocks of steps (6)–(7) and (8)-(9) are analogues of the PCA algorithm for the matrices $\mathbf{X}$ and $\mathbf{Y}$ [5]. Sequential repetition of the blocks takes into account the interaction between the matrices $\mathbf{X}$ and $\mathbf{Y}$.

The theoretical explanation of the PLS algorithm follows from the statements.

Утверждение 1. *The best description of the matrices $\mathbf{X}$ and $\mathbf{Y}$ taking into account their interrelation is achieved by maximizion the covariance between the vectors $\mathbf{t}_k$ and $\mathbf{u}_k$.*

The statement follows from the equation

$$\operatorname{cov}(\mathbf{t}_k, \mathbf{u}_k) = \operatorname{corr}(\mathbf{t}_k, \mathbf{u}_k) \cdot \sqrt{\operatorname{var}(\mathbf{t}_k)} \cdot \sqrt{\operatorname{var}(\mathbf{u}_k)}.$$

Maximization of variances of the vectors $\mathbf{t}_k$ and $\mathbf{u}_k$ corresponds to keeping information about original matrices, the correlation of these vectors corresponds to interrelation between $\mathbf{X}$ and $\mathbf{Y}$. ■

In the inner loop of the algorithm 1 the normalized weight vectors $\mathbf{w}_k$ and $\mathbf{c}_k$ are calculated. These vectors construct the matrices $\mathbf{W}$ and $\mathbf{C}$, respectively.

Утверждение 2. *The vector $\mathbf{w}_k$ and $\mathbf{c}_k$ are eigenvectors of the matrices $\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k$ and $\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k$, corresponding to the maximum eigenvalues.*

$$\mathbf{w}_k \propto \mathbf{X}_k^\mathsf{T}\mathbf{u}_{k-1} \propto \mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{c}_{k-1} \propto \mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{t}_{k-1} \propto \mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{w}_{k-1},$$
$$\mathbf{c}_k \propto \mathbf{Y}_k^\mathsf{T}\mathbf{t}_k \propto \mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{w}_k \propto \mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{X}_k^\mathsf{T}\mathbf{u}_{k-1} \propto \mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{c}_{k-1},$$

where the $\propto$ symbol means equality up to a multiplicative constant.

The statement follows from the fact that the update rule for vectors $\mathbf{w}_k$, $\mathbf{c}_k$ coincides with the iteration of the power method for the maximum eigenvalue.

Let a matrix $\mathbf{A}$ be diagonilized, $\mathbf{x}$ be some vector, then

$$\lim_{k\to\infty} \mathbf{A}^k\mathbf{x} = \lambda_{\max}(\mathbf{A}) \cdot \mathbf{v}_{\max},$$

where $\lambda_{\max}(\mathbf{A})$ is the maximum eigenvalue of the matrix $\mathbf{A}$, $\mathbf{v}_{\max}$ is the eigenvector of the matrix $\mathbf{A}$, corresponding to $\lambda_{\max}(\mathbf{A})$. ■

Утверждение 3. *The update rule for the vectors in steps (6)–(9) of the algorithm 1 corresponds to the maximization of the covariance between the vectors $\mathbf{t}_k$ and $\mathbf{u}_k$.*

The maximum covariance between the vectors $\mathbf{t}_k$ and $\mathbf{u}_k$ is equal to the maximum eigenvalue of the matrix $\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k$:

$$\begin{aligned}
\max_{\mathbf{t}_k,\mathbf{u}_k} \operatorname{cov}(\mathbf{t}_k, \mathbf{u}_k)^2 &= \max_{\substack{\|\mathbf{w}_k\|=1\\ \|\mathbf{c}_k\|=1}} \operatorname{cov}\left(\mathbf{X}_k\mathbf{w}_k, \mathbf{Y}_k\mathbf{c}_k\right)^2 = \max_{\substack{\|\mathbf{w}_k\|=1\\ \|\mathbf{c}_k\|=1}} \operatorname{cov}\left(\mathbf{c}_k^\mathsf{T}\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{w}_k\right)^2 = \\
&= \max_{\|\mathbf{w}_k\|=1} \operatorname{cov}\left\|\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{w}_k\right\|^2 = \max_{\|\mathbf{w}_k\|=1} \mathbf{w}_k^\mathsf{T}\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{w}_k = \\
&= \lambda_{\max}\left(\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\right),
\end{aligned}$$

where $\lambda_{\max}(\mathbf{A})$ is the maximum eigenvalue of the matrix $\mathbf{A}$. Using the statement 2, we obtain the required result. ■

After the inner loop, the step (11) is to compute vectors $\mathbf{p}_k$, $\mathbf{q}_k$ by projection of the columns of the matrices $\mathbf{X}_k$ and $\mathbf{Y}_k$ to the vector $\mathbf{t}_k$. To go to the next step one has to deflate the matrices $\mathbf{X}_k$ and $\mathbf{Y}_k$ by the one-rank approximations $\mathbf{t}_k\mathbf{p}_k^\mathsf{T}$ and $\mathbf{t}_k\mathbf{q}_k^\mathsf{T}$

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{t}_k\mathbf{p}_k^\mathsf{T} = \mathbf{X} - \sum_k \mathbf{t}_k\mathbf{p}_k^\mathsf{T},$$
$$\mathbf{Y}_{k+1} = \mathbf{Y}_k - \mathbf{t}_k\mathbf{q}_k^\mathsf{T} = \mathbf{Y} - \sum_k \mathbf{t}_k\mathbf{q}_k^\mathsf{T}.$$

Each next vector $\mathbf{t}_{k+1}$ turns out to be orthogonal to all vectors $\mathbf{t}_i$, $i = 1, \dots, k$.

Let assume that the dimension of object, response and latent variable spaces are equal to 2 ($n = r = l = 2$). Fig. 1 shows the result of the PLS algorithm in this case. Blue and green dots represent the rows of the matrices $\mathbf{X}$ and $\mathbf{Y}$, respectively. The dots were generated from a normal distribution with zero expectation. Contours of the distribution covariance matrices are shown in red. Black contours are unit circles. Red arrows correspond to principal components for the set of points. Black arrows correspond to the vectors of the matrices $\mathbf{W}$ and $\mathbf{C}$ from the PLS algorithm. The vectors $\mathbf{t}_k$ and $\mathbf{u}_k$ are equal to the projected matrices $\mathbf{X}_k$ and $\mathbf{Y}_k$ to the vectors $\mathbf{w}_k$ and $\mathbf{c}_k$, respectively, and are denoted by black pluses. Taking into account the interaction between the matrices $\mathbf{X}$ and $\mathbf{Y}$ deviates the vectors $\mathbf{w}_k$ and $\mathbf{c}_k$ from the principal components - не знаю можно ли писать просто во множественном числе directions. The deviation of the vectors $\mathbf{w}_k$ is insignificant. In the first iteration, $\mathbf{c}_1$ is close to the principal component pc_1, but the vectors $\mathbf{c}_k$ in the next iterations could strongly correlate. The difference in the behaviour of the vectors $\mathbf{w}_k$ and $\mathbf{c}_k$ is associated with the deflation process. In particular, we subtract from $\mathbf{Y}$ the one-rank approximation found in the space of the matrix $\mathbf{X}$.

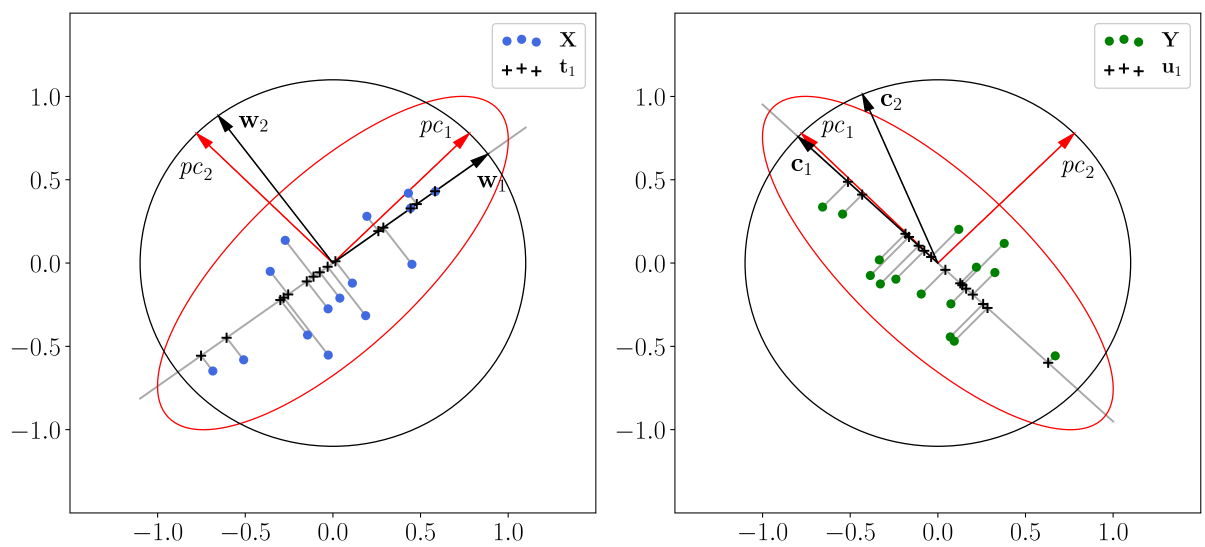

Figure 1: The result of the PLS algorithm for the case $n = r = l = 2$

.

To obtain the model predictions and find the model parameters, let multiply the both hand-sides of the equation (3) by the matrix $\mathbf{W}$. Since the rows of the residual matrix $\mathbf{E}$ are orthogonal to the columns of the matrix $\mathbf{W}$, we have

$$\mathbf{X}\mathbf{W} = \mathbf{T}\mathbf{P}^{\mathsf{T}}\mathbf{W}.$$

The linear transformation between objects in the input and latent spaces has the form

$$\mathbf{T} = \mathbf{X}\mathbf{W}^*, \tag{5}$$

where $\mathbf{W}^* = \mathbf{W}(\mathbf{P}^{\mathsf{T}}\mathbf{W})^{-1}$.

The matrix of the model parameters 1 could be found from equations (4), (5)

$$\mathbf{Y} = \mathbf{T}\mathbf{Q}^{\mathsf{T}} + \mathbf{E} = \mathbf{X}\mathbf{W}^*\mathbf{Q}^{\mathsf{T}} + \mathbf{E} = \mathbf{X}\mathbf{\Theta} + \mathbf{E}.$$

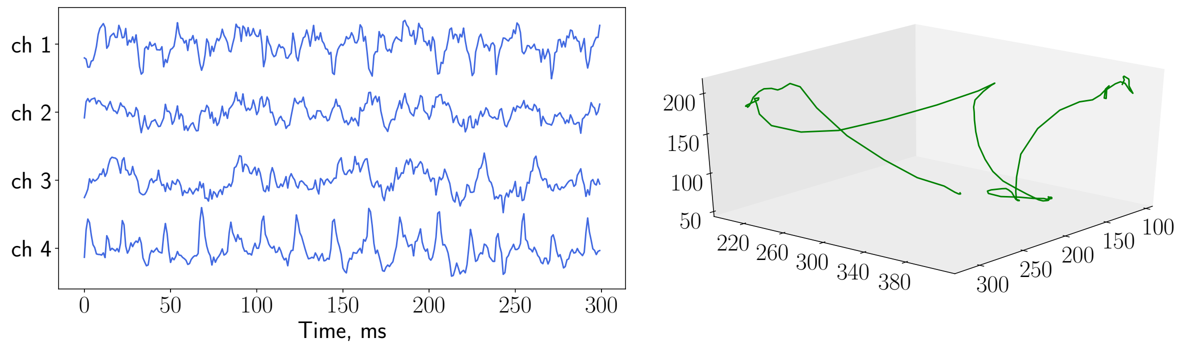

Figure 2: The ECoG data example. On the left the voltage data taken from multiple channels is shown, on the right there are coordinates of the hand along three axes.

Thus, the model parameters (1) are equal to

$$\boldsymbol{\Theta} = \mathbf{W}(\mathbf{P}^{\mathsf{T}}\mathbf{W})^{-1}\mathbf{Q}^{\mathsf{T}}. \tag{6}$$

To find the model predictions during the testing, we have to

- normalize the test data;
- compute the prediction of the model using the linear transformation with the matrix $\boldsymbol{\Theta}$ from (6);
- perform the inverse normalization.

4 Computational experiment

Time series of energy consumption contain hourly records (total of 52512 observations). A row of the matrix $\mathbf{X}$ is the local history of the signal for one week $n = 24 \times 7$. A row of the matrix $\mathbf{Y}$ is the local forecast of energy consumption for the next 24 hours $r = 24$. In this case, the matrices $\mathbf{X}$ and $\mathbf{Y}$ are autoregressive matrices.

In the case of the ECoG data, the matrix $\mathbf{X}$ consists of the spatial-temporal representation of voltage time series, and the matrix $\mathbf{Y}$ contains information about the position of the hand. The generation process of the matrix $\mathbf{X}$ from the voltage values described in [23]. Feature description in each time moment has dimension equal to 864. The hand position is described by the coordinates along three axes. An example of voltage data samples with the different channels and corresponding spatial coordinates of the hand are shown in Fig. 2. To predict the position of the hand in the next moments we used an autoregressive approach. One object consists of a feature description in a few moments. The answer is the hand position in the next moments of time. The task is to predict the hand position in the next few moments of time.

We introduce the mean-squared error for matrices $\mathbf{A} = [a_{ij}]$ and $\mathbf{B} = [b_{ij}]$

$$\mathrm{MSE}(\mathbf{A}, \mathbf{B}) = \sum_{i,j} (a_{ij} - b_{ij})^2.$$

To estimate the prediction quality, we compute the normalized MSE

$$\mathrm{NMSE}(\mathbf{Y}, \hat{\mathbf{Y}}) = \frac{\mathrm{MSE}(\mathbf{Y}, \hat{\mathbf{Y}})}{\mathrm{MSE}(\mathbf{Y}, \bar{\mathbf{Y}})}, \tag{7}$$

where $\hat{\mathbf{Y}}$ is the model outcome, $\bar{\mathbf{Y}}$ is the фмукфпу constant forecast over the columns of the matrix.

4.1 Energy consumption dataset

To find the optimal dimensionality l of the latent space, the energy consumption dataset was divided into training and validation parts. The training data consists of 700 objects, the validation data is of 370 ones. The dependence of the normalized mean-squared error (7) on the latent space with dimensionality l is shown in Fig. 3. First, the error drops sharply with increasing the latent space dimensionality and then changes slightly.

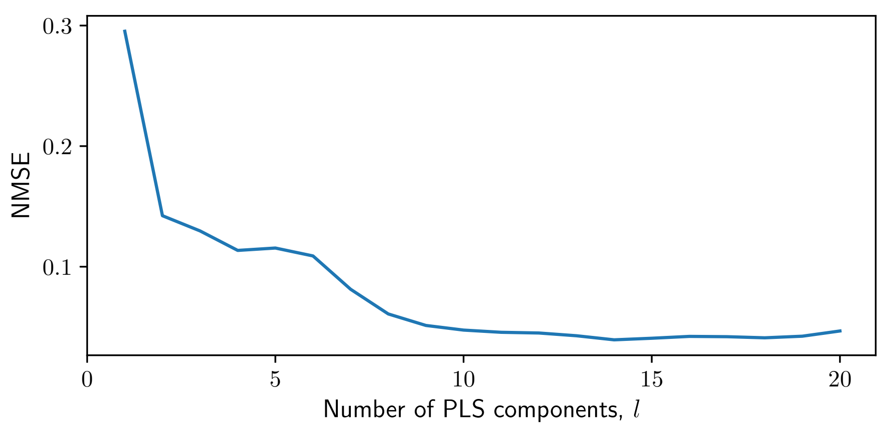

Figure 3: NMSE as a function of dimension l of latent space for energy consumption data.

The error achieves the minimum value for $l = 14$. Let build a forecast of energy consumption for a given l. The result is shown in Fig. 4. The PLS algorithm restored the autoregressive dependence and found the daily seasonality.

4.2 ECoG dataset

Fig. 5 illustrates the dependence of the normalized mean-squared error (7) on the latent space dimensionality l for ECoG dataset. The approximation error changes slightly for

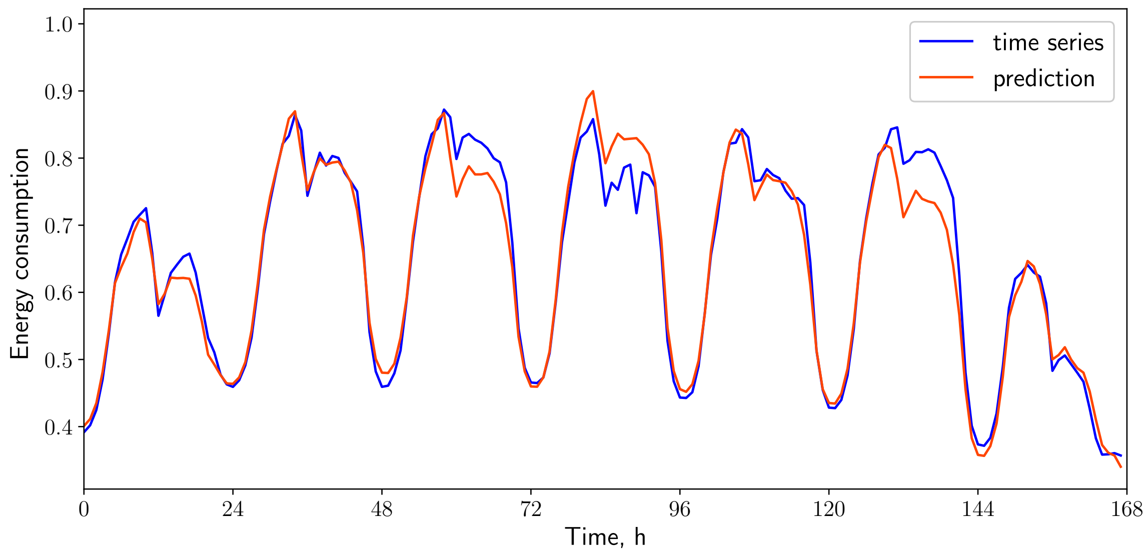

Figure 4: The energy consumption forecast by the PLS algorithm (the latent space dimensionality is equal to $l = 14$).

$l > 5$. The joint spatial-temporal representation of objects and the position of the hand can be represented as a vector of dimensionality equal to $l \ll n$. Let us fix $l = 5$. An example of the approximation of the hand position is shown in Fig. 6. Solid lines represent the true coordinates of the hand along all axes, the dotted lines show the approximation by the PLS algorithm.

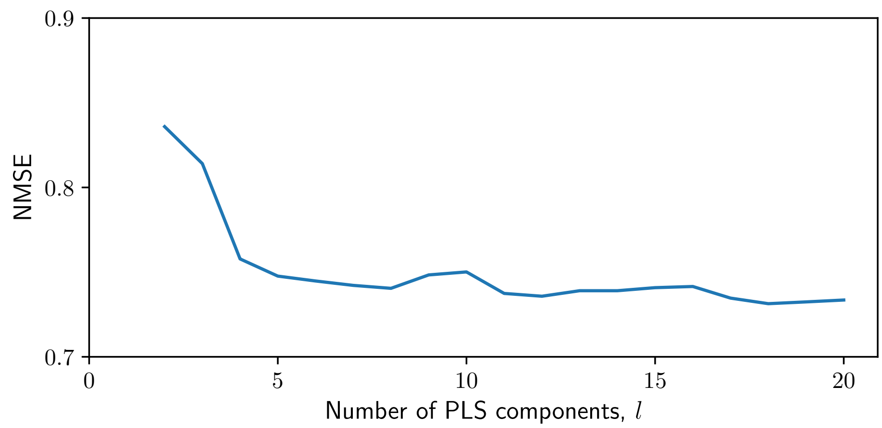

Figure 5: NMSE as a function of dimension l of latent space for the ECoG data.

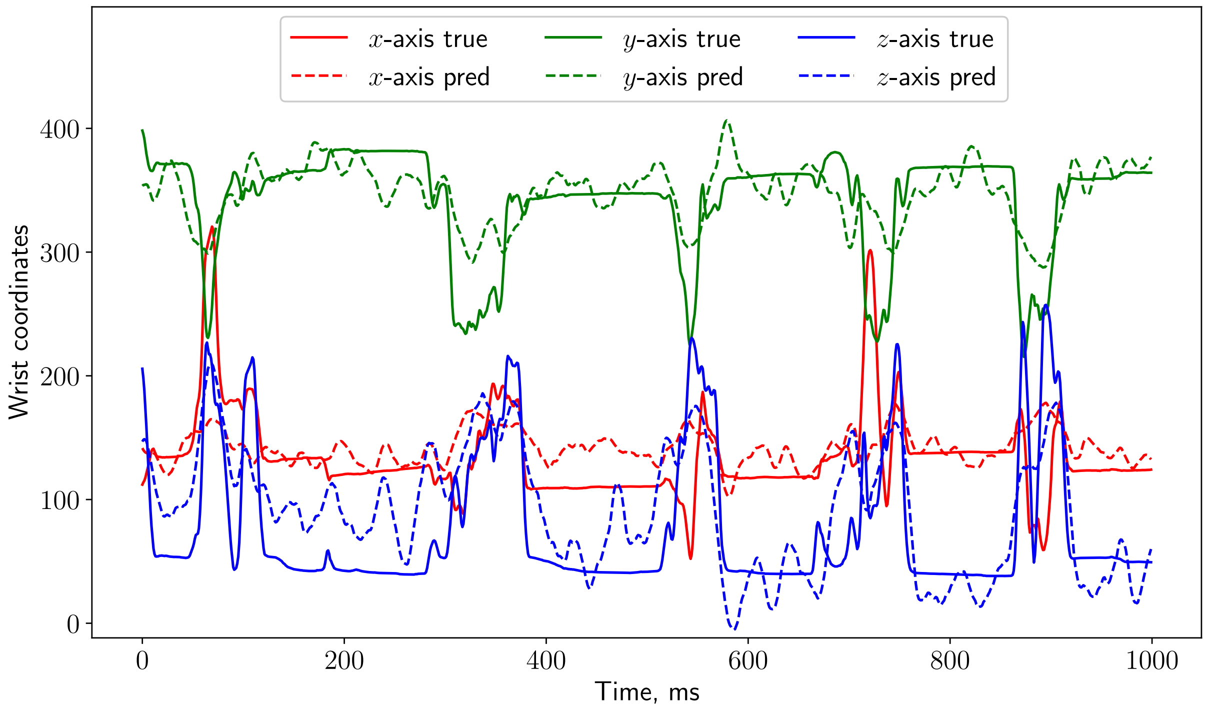

Figure 6: The hand motions predicted by the PLS algorithm (the latent space dimensionality is equal to $l = 5$).

5 Conclusion

In the paper we proposed the approach for solving the problem of time series decoding and forecasting. The algorithm of partial least squares allows to build a vector target variable prediction -> target vector prediction. The latent space contains information about the objects and the responses and dramatically reduces the dimensionsionalies of the input matrices. The computational experiment demonstrated the applicability of the proposed method to the tasks of electricity consumption forecasting and brain-computer interface designing.

References

[1] AM Katrutsa and VV Strijov. Stress test procedure for feature selection algorithms. *Chemometrics and Intelligent Laboratory Systems*, 142:172–183, 2015.

[2] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *arXiv preprint arXiv:1601.07996*, 2016.

[3] Jacob A Wegelin et al. A survey of partial least squares (pls) methods, with emphasis on the two-block case. *University of Washington, Department of Statistics, Tech. Rep*, 2000.

[4] Hervé Abdi. Partial Least Squares (PLS) Regression. *Encyclopedia for research methods for the social sciences*, pages 792–795, 2003.

[5] Paul Geladi and Bruce R Kowalski. Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17, 1986.

[6] Paul Geladi. Notes on the history and nature of partial least squares (PLS) modelling. *Journal of Chemometrics*, 2(January):231–246, 1988.

[7] Agnar Höskuldsson. PLS regression. *Journal of Chemometrics*, 2(August 1987):581–591, 1988.

[8] Sijmen De Jong. Simpls: an alternative approach to partial least squares regression. *Chemometrics and intelligent laboratory systems*, 18(3):251–263, 1993.

[9] Roman Rosipal and Nicole Kramer. Overview and Recent Advances in Partial Least Squares. *C. Saunders et al. (Eds.): SLSFS 2005, LNCS 3940*, pages 34–51, 2006.

[10] Roman Rosipal. Nonlinear partial least squares: An overview. *Chemoinformatics and Advanced Machine Learning Perspectives: Complex Computational Methods and Collaborative Techniques*, pages 169–189, 2011.

[11] Svante Wold, Nouna Kettaneh-Wold, and Bert Skagerberg. Nonlinear pls modeling. *Chemometrics and Intelligent Laboratory Systems*, 7(1-2):53–65, 1989.

[12] Ildiko E. Frank. A nonlinear PLS model. *Chemometrics and Intelligent Laboratory Systems*, 8(2):109–119, 1990.

[13] S Joe Qin and Thomas J McAvoy. Nonlinear pls modeling using neural networks. *Computers & Chemical Engineering*, 16(4):379–391, 1992.

[14] Xuefeng F. Yan, Dezhao Z. Chen, and Shangxu X. Hu. Chaos-genetic algorithms for optimizing the operating conditions based on RBF-PLS model. *Computers and Chemical Engineering*, 27(10):1393–1404, 2003.

[15] Hugo Hiden, Ben McKay, Mark Willis, and Gary Montague. Non-linear partial least squares using genetic. In *Genetic Programming 1998: Proceedings of the Third*, pages 128–133. Morgan Kaufmann, 1998.

[16] Project tycho http://neurotycho.org/food-tracking-task.

[17] José del R Millán, Rüdiger Rupp, Gernot Mueller-Putz, Roderick Murray-Smith, Claudio Giugliemma, Michael Tangermann, Carmen Vidaurre, Febo Cincotti, Andrea Kubler, Robert Leeb, et al. Combining brain–computer interfaces and assistive technologies: State-of-the-art and challenges. *Frontiers in Neuroscience*, 4:161, 2010.

[18] SG Mason, A Bashashati, M Fatourechi, KF Navarro, and GE Birch. A comprehensive survey of brain interface technology designs. *Annals of biomedical engineering*, 35(2):137–169, 2007.

[19] José del R Millán, Frédéric Renkens, Josep Mouriño, and Wulfram Gerstner. Brain-actuated interaction. *Artificial Intelligence*, 159(1-2):241–259, 2004.

[20] Luis Fernando Nicolas-Alonso and Jaime Gomez-Gil. Brain computer interfaces, a review. *Sensors*, 12(2):1211–1279, 2012.

[21] Setare Amiri, Reza Fazel-Rezai, and Vahid Asadpour. A review of hybrid brain-computer interface systems. *Advances in Human-Computer Interaction*, 2013:1, 2013.

[22] Andrey Eliseyev and Tetiana Aksenova. Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ecog) recording. *PloS one*, 11(5):e0154878, 2016.

[23] Motrenko A. Gasanov I. Creation of approximating scalogram description in a problem of movement prediction. *Journal of Machine Learning and Data Analysis*, 3(2):160–169, 2017.

[24] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[25] G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.