

Dimensionality reduction for time series decoding and forecasting problems*

R. V. Isachenko¹, M. R. Vladimirova², V. V. Strijov³

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

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 the object 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 the object and target spaces are different in nature, the outcome is a system response to the input signal. The autoregressive design matrix contains 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).

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

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

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

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

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

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

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

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

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

2 Problem statement

Given the 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 the 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 $\mathbf{y} \in \mathbb{R}^r$

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

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

The task is to find the matrix of the model parameters $\boldsymbol{\Theta}$ 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} :

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

The linear dependence of the columns of the matrix X leads to 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 dimension 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. Partial least squares (PLS) algorithm projects the design matrix \mathbf{X} and the target matrix in latent space \mathbb{R}^l with low dimension ($l < r < n$). 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 design matrix \mathbf{X} and the target matrix \mathbf{Y} is projected into the latent space in the following way:

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

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

is a matrix of 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} ,

79 respectively. After the computation of the next set of vectors, the one-rank approximations
80 are subtracted from the matrices \mathbf{X} , \mathbf{Y} . This step is called matrix deflation. The first step
81 one has to normalize the columns of the original matrices (subtract the mean and divide
82 by the standard deviation). During the test mode we need to normalize the test data,
83 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:      $\mathbf{w}_k := \mathbf{X}_k^\top \mathbf{u}_{k-1} / (\mathbf{u}_{k-1}^\top \mathbf{u}_{k-1}); \quad \mathbf{w}_k := \frac{\mathbf{w}_k}{\|\mathbf{w}_k\|}$ 
7:      $\mathbf{t}_k := \mathbf{X}_k \mathbf{w}_k$ 
8:      $\mathbf{c}_k := \mathbf{Y}_k^\top \mathbf{t}_k / (\mathbf{t}_k^\top \mathbf{t}_k); \quad \mathbf{c}_k := \frac{\mathbf{c}_k}{\|\mathbf{c}_k\|}$ 
9:      $\mathbf{u}_k := \mathbf{Y}_k \mathbf{c}_k$ 
10:   until  $\mathbf{t}_k$  stabilizes
11:    $\mathbf{p}_k := \mathbf{X}_k^\top \mathbf{t}_k / (\mathbf{t}_k^\top \mathbf{t}_k), \quad \mathbf{q}_k := \mathbf{Y}_k^\top \mathbf{t}_k / (\mathbf{t}_k^\top \mathbf{t}_k)$ 
12:    $\mathbf{X}_{k+1} := \mathbf{X}_k - \mathbf{t}_k \mathbf{p}_k^\top$ 
13:    $\mathbf{Y}_{k+1} := \mathbf{Y}_k - \mathbf{t}_k \mathbf{q}_k^\top$ 

```

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

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

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

91 The statement follows from the equation

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

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

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

97 **Утверждение 2.** The vector \mathbf{w}_k and \mathbf{c}_k are eigenvectors of the matrices $\mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k$
 98 and $\mathbf{Y}_k^\top \mathbf{X}_k \mathbf{X}_k^\top \mathbf{Y}_k$, corresponded to the maximum eigenvalues.

$$\begin{aligned}\mathbf{w}_k &\propto \mathbf{X}_k^\top \mathbf{u}_{k-1} \propto \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{c}_{k-1} \propto \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{t}_{k-1} \propto \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{w}_{k-1}, \\ \mathbf{c}_k &\propto \mathbf{Y}_k^\top \mathbf{t}_k \propto \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{w}_k \propto \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{X}_k^\top \mathbf{u}_{k-1} \propto \mathbf{Y}_k^\top \mathbf{X}_k \mathbf{X}_k^\top \mathbf{Y}_k \mathbf{c}_{k-1},\end{aligned}$$

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

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

102 Let a matrix \mathbf{A} be diagonalized, \mathbf{x} be some vector, then

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

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

105 **Утверждение 3.** The update rule for the vectors in steps (6)–(9) of the algorithm 1
 106 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 $\mathbf{X}_k^\top \mathbf{Y}_k \mathbf{Y}_k^\top \mathbf{X}_k$:

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

107 where $\lambda_{\max}(\mathbf{A})$ is the maximum eigenvalue of the matrix \mathbf{A} . Using the statement 2, we
 108 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 и \mathbf{Y}_k to the vector \mathbf{t}_k . To go to the next step one has to deflate the matrices \mathbf{X}_k и \mathbf{Y}_k by the one-rank approximations $\mathbf{t}_k \mathbf{p}_k^\top$ и $\mathbf{t}_k \mathbf{q}_k^\top$

$$\begin{aligned}\mathbf{X}_{k+1} &= \mathbf{X}_k - \mathbf{t}_k \mathbf{p}_k^\top = \mathbf{X} - \sum_k \mathbf{t}_k \mathbf{p}_k^\top, \\ \mathbf{Y}_{k+1} &= \mathbf{Y}_k - \mathbf{t}_k \mathbf{q}_k^\top = \mathbf{Y} - \sum_k \mathbf{t}_k \mathbf{q}_k^\top.\end{aligned}$$

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

110 Let assume that the dimension of objects, responses and latent variables spaces are
 111 equal to 2 ($n = r = l = 2$). Fig. 1 shows the result of the PLS algorithm in this case.
 112 Blue and green dots represent the rows of the matrices \mathbf{X} and \mathbf{Y} , respectively. The dots

113 were generated from a normal distribution with zero expectation. The covariance matrices
 114 contours of the distributions are shown in red. The black contour is a unit circle. Red
 115 arrows correspond to principal components for the set of points. Black arrows correspond
 116 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
 117 equal to the projected matrices \mathbf{X}_k and \mathbf{Y}_k on the vectors \mathbf{w}_k and \mathbf{c}_k , respectively, and are
 118 denoted by black pluses. Taking into account the interaction between the matrices \mathbf{X} and \mathbf{Y}
 119 deviates the vectors \mathbf{w}_k and \mathbf{c}_k from the principal components directions. The deviation of
 120 the vectors \mathbf{w}_k is insignificant. In the first iteration, \mathbf{c}_1 is close to the principal component
 121 pc_1 , but the vectors \mathbf{c}_k in the next iterations could strongly correlate. The difference in the
 122 behaviour of the vectors \mathbf{w}_k and \mathbf{c}_k is associated with the deflation process. In particular
 we subtract from \mathbf{Y} one-rank approximation found in the space of the matrix \mathbf{X} .

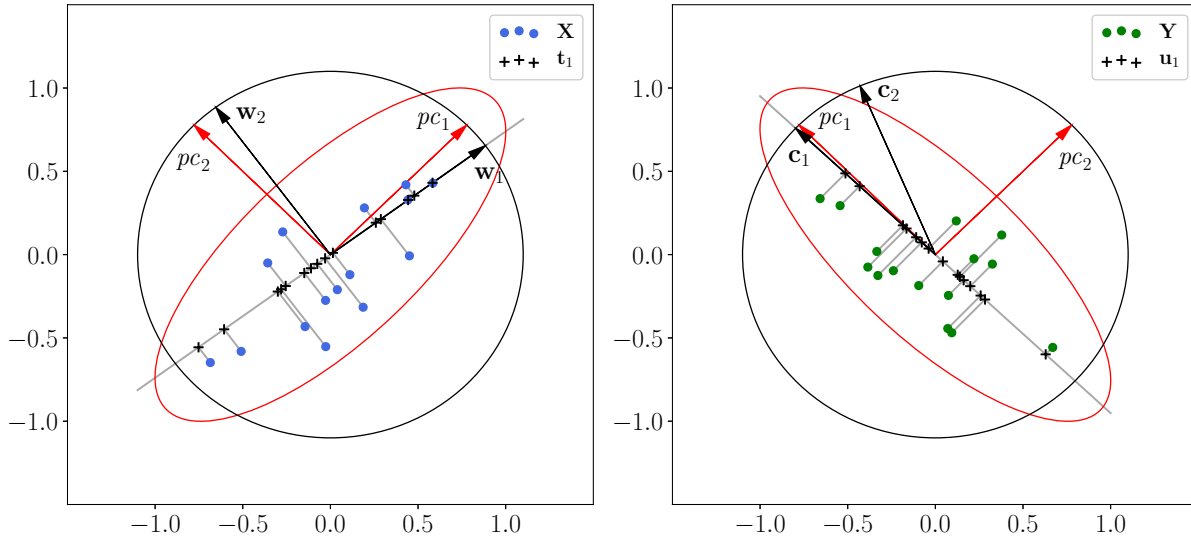


Figure 1: The result of the PLS algorithm for the case $n = r = l = 2$

123
 124 To obtain the model predictions and find the model parameters, let multiply the both
 125 sides of the equation (3) by the matrix \mathbf{W} . Since the rows of the residual matrix \mathbf{E} are
 126 orthogonal to the columns of the matrix \mathbf{W} , we have

$$\mathbf{XW} = \mathbf{TP}^T \mathbf{W}.$$

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

$$\mathbf{T} = \mathbf{XW}^*, \quad (5)$$

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

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

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

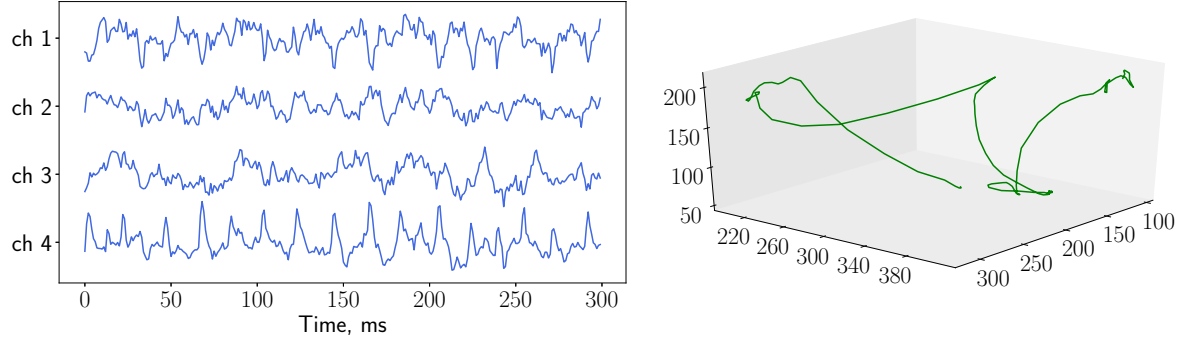


Figure 2: The ECoG data example. On the left voltage data taken from multiple channels is shown, on the right there are coordinates of the hand along three axes.

Thus, the model parameters (1) are equal to

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

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 Θ 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 $\mathbf{B} = [b_{ij}]$

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

To estimate the prediction quality, we compute the normalized MSE

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

where $\hat{\mathbf{Y}}$ is the model outcome, $\bar{\mathbf{Y}}$ is the $\phi\mu\gamma\kappa\phi\pi\gamma$ 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.

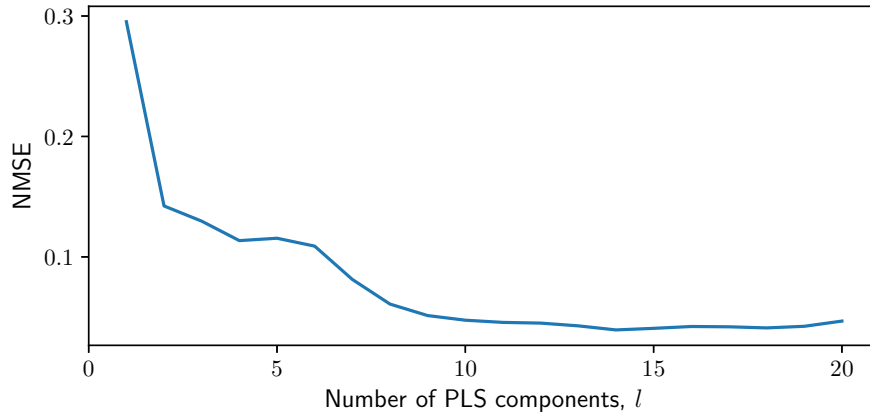


Figure 3: NMSE as a function of dimension l of latent space for energy consumption data.

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

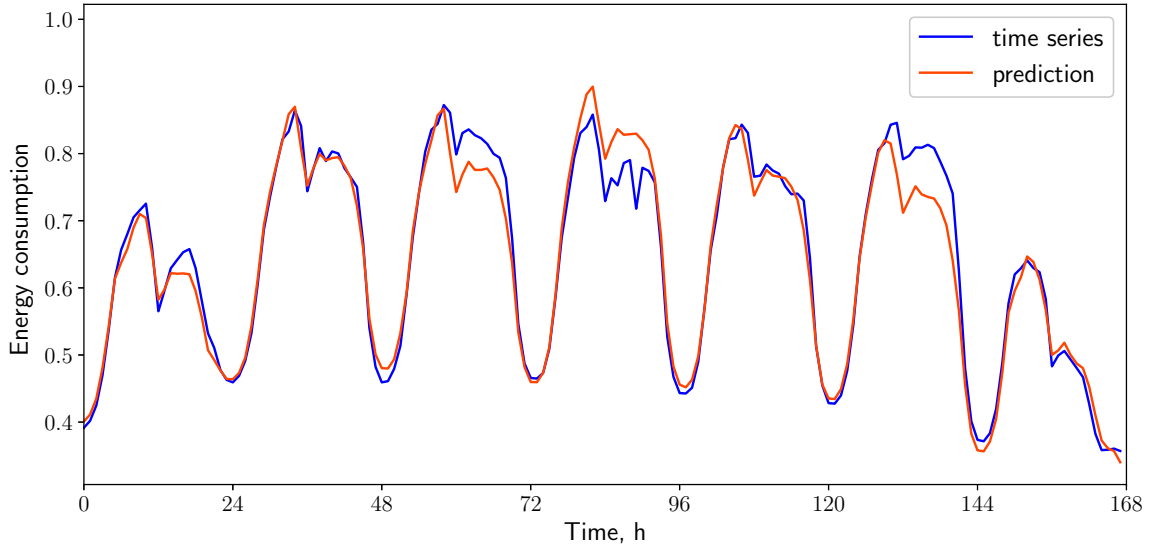


Figure 4: The energy consumption forecast by the PLS algorithm (the latent space dimensionality is equal to $l = 14$).

166 $l > 5$. The joint spatial-temporal representation of objects and the position of the hand
 167 can be represented as a vector of dimensionality $l \ll n$. Let us fix $l = 5$. An example of
 168 the approximation of the hand position is shown in Fig. 6. Solid lines represent the true
 169 coordinates of the hand along all axes, the dotted lines show the approximation by the
 170 PLS algorithm.

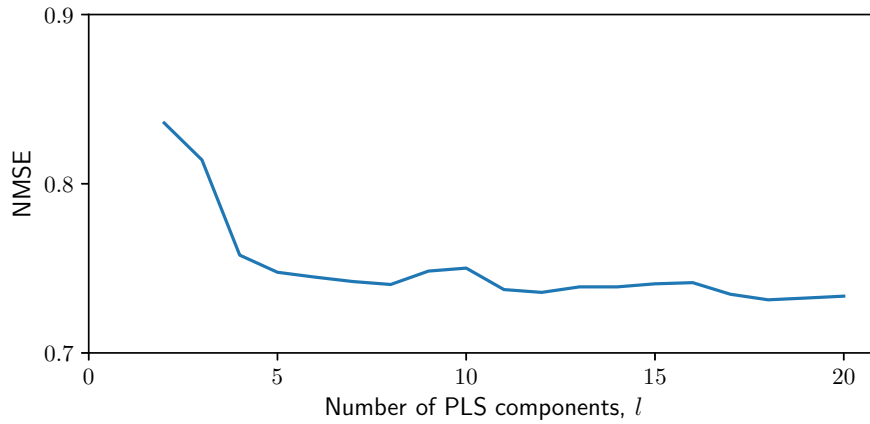


Figure 5: NMSE as a function of dimension l of latent space for the ECoG data.

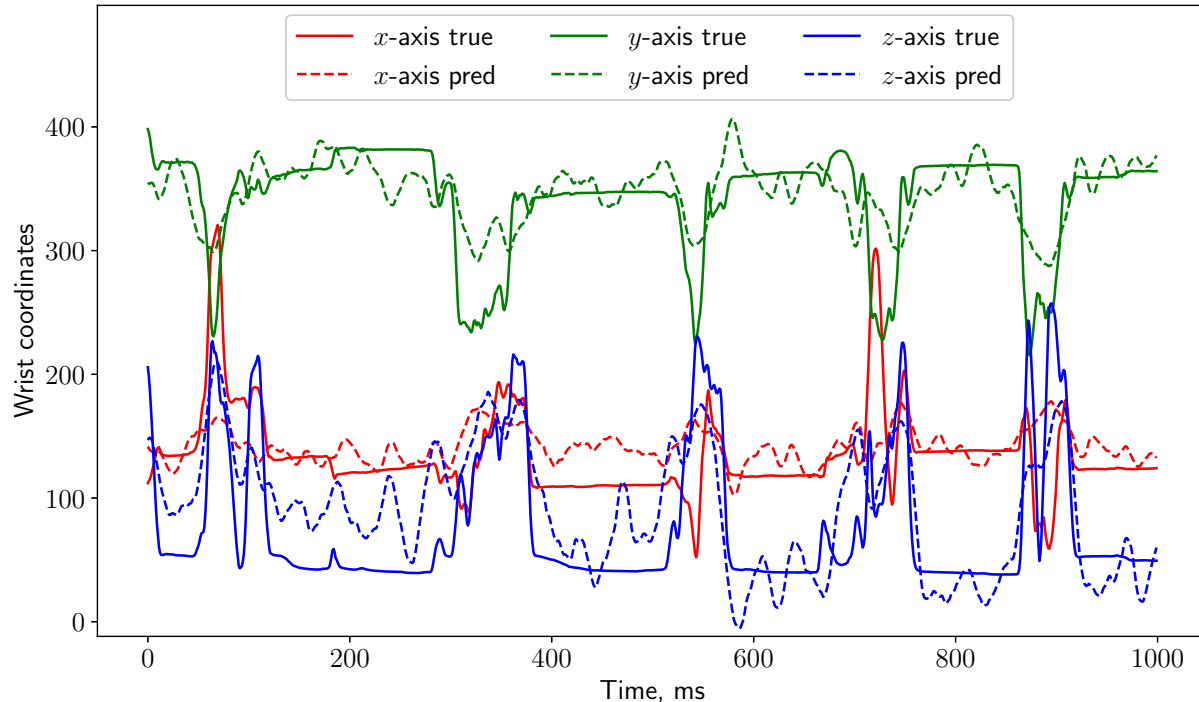


Figure 6: The hand motions predicted by the PLS algorithm (the latent space dimensionality is equal to $l = 5$).

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. The latent space contains information about the objects and the answers and dramatically reduces the dimensions 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, Suhan 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 Elishev 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.