Снижение размерности пространства зависимой переменной в задачах прогнозирования.*

Мария Владимирова
mrvladimirova@gmail.com
Московский физико-технический институт

Решается задача обнаружения способов зависимостей в прогнозируемой переменной. Используется набор гомогенных моделей, восстанавливающих прогноз по общему для всех переменных описанию объектов. Анализируется различие в пространстве параметров моделей. По результатам анализа выбирается оптимальная структура каждой модели.

Ключевые слова: *multitask learning, partial least squares, многозадачность, ...*

1. Introduction

Electric energy is a significant driving force for economic development, while the accuracy of demand forecasts is an important factor leading to the success of efficiency planning. For this reason, energy analysts need a guideline to better choose the most appropriate forecasting techniques in order to provide accurate forecasts of electricity consumption trends.

Concept named multitask learning (MTL) [1, 2] (*refer to an article, e.g. Caruana, R. Multitask learning.*) was proposed to use the redundancy information.

MTL can be carried out using machine learning methods yielding models with several independent outputs, such as neural networks, another type of inductive transfer, uses extra tasks to build the models, predictions of which are further used as extra inputs for the main task [3]. (*refer to chemometrics*)

There is the multitask learning method of giving extra information through the model's outputs in artificial neural networks, where extra outputs are trained in parallel with the main task outputs while using a shared representation [4](*Caruana, 2003*). Many learning methods do not have representation naturally shared between tasks. [1] (*Learning to learn*) presents an algorithm and results for MTL with case-based methods such as k-nearest neighbor and kernel regression, and sketches an algorithm for MTL in decision tree induction.

In this letter, we focus on the multitask problem using Partial Least Squares (PLS) regression method. The goal of PLS regression [5](*Abdi*) is to predict $\mathbf{Y}$ from $\mathbf{X}$ and to describe their common structure. When $\mathbf{Y}$ is a vector and $\mathbf{X}$ is a full rank matrix, this goal could be accomplished using ordinary multiple regression. When the number of predictors is large compared to the number of observations, $\mathbf{X}$ is likely to be singular and the regression approach is no longer feasible because of multicollinearity.

For the history of PLS and about PLS regression see [6, 7](*Geladi*) and (*Höskuldsson*). The difference between PLS and related technique, different varieties of PLS regression can be viewed in [8](*Bayesian PLS*). MTL using PLS regression method was proposed to address the multivariate calibration problems in the analytical chemistry in [9](*MTL using PLS*).

A nonlinear extension of the PLSR method is firstly introduced in [10]. A lot of PLS methods were developed in the literature. The activation functions of artificial neural networks are used in PLS method. Because the activation functions provide highly nonlinear transformations, they solve multicollinearity problem. Moreover, PLS method has non-linear modeling ability. Paper

Проект поддержан грантом РФФИ №16-07-01155.

proposes non-linear PLS method based on defferent methods: feed forward artificial neural networks [11], radial bases activation functions [12], logistic activation function and particle swarm optimization methods [13], differently use feed forward neural networks [14], Elman feedback artificial neural networks [15].

2. Problem statement

2.1 Partial Least Squares Regression

Partial Least Squares (PLS) regression finds components from $\mathbf{X}$ that are also relevant for $\mathbf{Y}$. Specifically, PLS regression searches for a set of components (called latent vectors) that performs a simultaneous decomposition of $\mathbf{X}$ and $\mathbf{Y}$ with the constraint that these components explain as much as possible of the covariance between $\mathbf{X}$ and $\mathbf{Y}$. This step generalizes PCA. It is followed by a regression step where the decomposition of $\mathbf{X}$ is used to predict $\mathbf{Y}$.

The general underlying model of multivariate PLS is

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathsf{T}} + \mathbf{E},$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^{\mathsf{T}} + \mathbf{F},$$

where $\mathbf{X}$ is an $n \times m$ rank matrix of predictors, $\mathbf{Y}$ is an $n \times p$ matrix of responses. $\mathbf{T}$ and $\mathbf{U}$ are $n \times l$ score matrices that are, respectively, projections of $\mathbf{X}$ and projections of $\mathbf{Y}$, $\mathbf{T}^{\mathsf{T}}\mathbf{T} = \mathbf{I}$. $\mathbf{P}$ and $\mathbf{Q}$ are, respectively, $m \times l$ and $p \times l$ orthogonal loading matrices. $\mathbf{B}$ is diagonal matrix with "regression weights" as diagonal elements and matrices $\mathbf{E}$ and $\mathbf{F}$ are the error terms, assumed to be independent and identically distributed random normal variables. The columns of $\mathbf{T}$ are called latent vectors. When their number is equal to the rank of $\mathbf{X}$, they perform an exact decomposition of $\mathbf{X}$. The decompositions of $\mathbf{X}$ and $\mathbf{Y}$ are made so as to maximise the covariance between $\mathbf{T}$ and $\mathbf{U}$. Specifically, the goal is to obtain a first pair of vectors $\mathbf{t} = \mathbf{X}\mathbf{p}$ and $\mathbf{u} = \mathbf{Y}\mathbf{q}$ with the constraints that $\mathbf{p}^{\mathsf{T}}\mathbf{p} = 1$, $\mathbf{t}^{\mathsf{T}}\mathbf{t} = 1$ and $\mathbf{t}^{\mathsf{T}}\mathbf{u}$ be maximal.

Iterative process:

1. $\mathbf{w} = \mathbf{X}^{\mathsf{T}}\mathbf{u}/(\mathbf{u}^{\mathsf{T}}\mathbf{u})$
2. $\|\mathbf{w}\| \to 1$
3. $\mathbf{t} = \mathbf{X}\mathbf{w}$
4. $\mathbf{c} = \mathbf{Y}^{\mathsf{T}}\mathbf{t}/(\mathbf{t}^{\mathsf{T}}\mathbf{t})$
5. $\|\mathbf{c}\| \to 1$
6. $\mathbf{u} = \mathbf{Y}\mathbf{c}$

By $\mathbf{T} = \mathbf{U}\mathbf{B}$, $\mathbf{P} \sim \mathbf{X}^{\mathsf{T}}\mathbf{U}$ and $\mathbf{Q} \sim \mathbf{Y}^{\mathsf{T}}\mathbf{T}$. To preserve relation $\mathbf{T} = \mathbf{U}\mathbf{B}$ between $\mathbf{T}$ and $\mathbf{U}$ one may use the following fused procedure: set $\mathbf{u}_0 = y_1,\ k = 1$ and while $\mathbf{t}_k$ changes, repeate:

$$\mathbf{p} = \mathbf{X}^{\mathsf{T}}\mathbf{t}/(\mathbf{t}^{\mathsf{T}}\mathbf{t}), \quad \mathbf{q} = \mathbf{Y}^{\mathsf{T}}\mathbf{u}/(\mathbf{u}^{\mathsf{T}}\mathbf{u})$$

Then we have

$$\mathbf{p} = \mathbf{X}^{\mathsf{T}}\mathbf{u} = \mathbf{X}^{\mathsf{T}}\mathbf{Y}\mathbf{q} = \mathbf{X}^{\mathsf{T}}\mathbf{Y}\mathbf{Y}^{\mathsf{T}}\mathbf{t} = \mathbf{X}^{\mathsf{T}}\mathbf{Y}\mathbf{Y}^{\mathsf{T}}\mathbf{X}\mathbf{p}/\|\mathbf{X}^{\mathsf{T}}\mathbf{Y}\mathbf{Y}^{\mathsf{T}}\mathbf{X}\|,$$

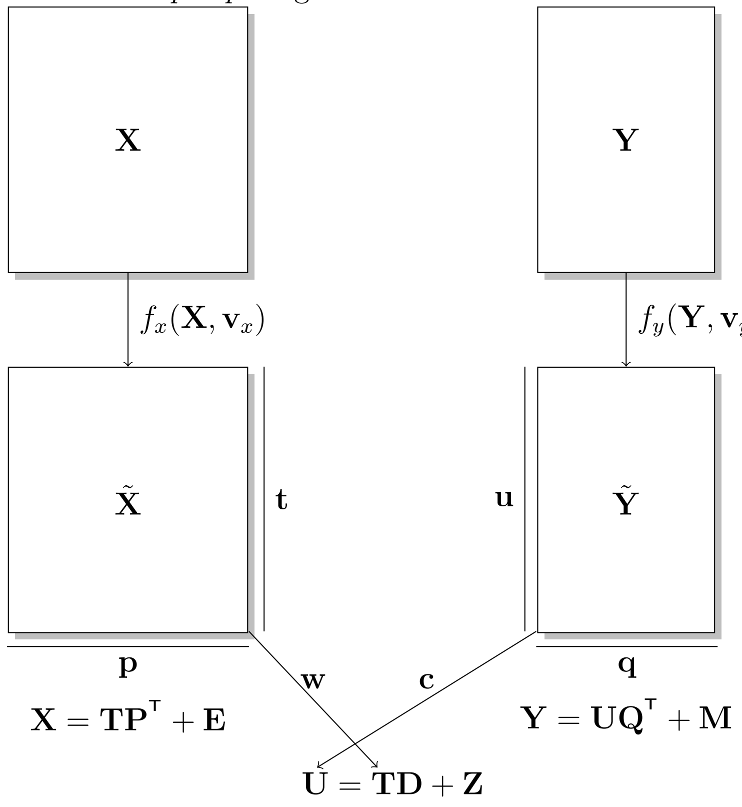
which is update rule for the eigenvector of the covariance matrix of $\mathbf{Y}^{\mathsf{T}}\mathbf{X}$. In [*Kee Siong Ng*] shown that $\mathbf{p}$ is is the solution to the optimisation problem

$$\arg\max_{\|\mathbf{p}\|=1} \mathbf{p}^{\mathsf{T}}\mathbf{X}^{\mathsf{T}}\mathbf{Y}\mathbf{Y}^{\mathsf{T}}\mathbf{X}\mathbf{p} = \arg\max_{\|\mathbf{p}\|=1} \mathrm{Var}(\mathbf{X}\mathbf{p})(\mathrm{Corr}(\mathbf{Y}, \mathbf{X}\mathbf{p}))^{\mathsf{T}}\mathrm{Corr}(\mathbf{Y}, \mathbf{X}\mathbf{p}) \tag{1}$$

The most frequently used variant of linear PLS is based on two assumptions i) the score vectors $\mathbf{t}_{i\,i=1}^{p}$ are good predictors of $\mathbf{Y}$, and ii) a linear inner relation between the scores vectors $\mathbf{t}$ and $\mathbf{u}$ exists; that is,

$$\mathbf{U} = \mathbf{T}\mathbf{D} + \mathbf{H} \tag{2}$$

where $\mathbf{D}$ is the $p \times p$ diagonal matrix and $\mathbf{H}$ denotes the matrix of residuals.


$\mathbf{X}$
$\mathbf{Y}$
$f_x(\mathbf{X}, \mathbf{v}_x)$
$f_y(\mathbf{Y}, \mathbf{v}_y)$
$\tilde{\mathbf{X}}$
$\mathbf{t}$
$\mathbf{u}$
$\tilde{\mathbf{Y}}$
$\mathbf{p}$
$\mathbf{w}$
$\mathbf{c}$
$\mathbf{q}$
$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathsf{T}} + \mathbf{E}$
$\mathbf{Y} = \mathbf{U}\mathbf{Q}^{\mathsf{T}} + \mathbf{M}$
$\mathbf{U} = \mathbf{T}\mathbf{D} + \mathbf{Z}$

3 Pre-training

Let us consider monotone functions **??**.

Response form	Function	Parameters	x'	y'
Very fast growth	$g = \exp(a + bx)$	$b > 0$	x	$\ln g$
Fast growth	$g = \exp(a + b \ln x)$	$b > 1$	$\ln x$	$\ln g$
Slow growth	$g = \exp(a + b \ln x)$	$0 < b < 1$	$\ln x$	$\ln g$
Very slow growth	$g = a + b \ln x$	$b > 0$	$\ln x$	g
Slow stabilization	$g = a + b/x$	$b \neq 0$	$1/x$	g
Fast stabilization	$g = a + b \exp(-x)$	$b \neq 0$	$\exp(-x)$	g
Sigmoid	$g = 1/(a + b \exp(-x))$	$b > 0$	$\exp(-x)$	$1/g$

Таблица 1. Monotone functions

3.0 The dependent variable transformation

$$\mathbf{Y} \xrightarrow{g_y(\mathbf{Y}, \mathbf{v}_y)} \breve{\mathbf{Y}} \xrightarrow{h_y(\breve{\mathbf{Y}})} \tilde{\mathbf{Y}}$$

Firstly, consider curvelinear transformation with vector of parameters $\mathbf{v}_y$

$$\breve{\mathbf{Y}} = g_y(\mathbf{Y}, \mathbf{v}_y) = \mathbf{G}_y(\mathbf{Y}, \mathbf{v}_y), \tag{3}$$

then, nonparametric transformation

$$\tilde{\mathbf{Y}} = h_y(\breve{\mathbf{Y}}) = h_y(g_y(\mathbf{Y}, \mathbf{v}_y)) = f_y(\mathbf{Y}, \mathbf{v}_y)$$

The second-order Taylor expansion of (3) has the form

$$\check{\mathbf{y}} = \check{\mathbf{y}}_0 + \frac{\partial g_y}{\partial \mathbf{v}_y}\Big|_0 \Delta \mathbf{v}_y,$$

where $\check{\mathbf{y}}_0 = g_y(\mathbf{t})$ is the value of g_y at the known value of $\mathbf{y}$. The following steps to compute $\Delta\mathbf{v}$ were suggested. The approach considers the difference $\check{\mathbf{y}} - \check{\mathbf{y}}_0 = \frac{\partial g_y}{\partial \mathbf{v}_y}\Big|_0 \Delta\mathbf{v}$ at the first step. Next, this leads to the following definition of a mismatch $\mathbf{e}$

$$\mathbf{e} = \check{\mathbf{y}} - \check{\mathbf{y}}_0 = \frac{\partial g_y}{\partial \mathbf{v}_y}\Big|_0 \Delta \mathbf{v}_y = \mathbf{J}_y \Delta \mathbf{v}_y,$$

where $\mathbf{J}_y$ consists of the partial derivatives $\left\{ \frac{\partial g}{\partial v_i}\Big|_0 \right\}_{i=1}^{N}$. From this mismatch relation $\Delta\mathbf{v}_y$ can be directly computed by regressing the mismatch $\mathbf{e}$ on $\mathbf{J}_y$

$$\Delta \mathbf{v}_y = (\mathbf{J}_y^{\mathsf{T}} \mathbf{J}_y)^{-1} \mathbf{J}_y^{\mathsf{T}} \mathbf{e}. \tag{4}$$

3.1 Stacked Nonlinear Autoencoders

A stacked autoencoder is a neural network consisting of multiple layers of sparse autoencoders in which the outputs of each layer is wired to the inputs of the successive layer.

Let $W_{ij}^{(l)}$ denote the parameter (or weight) associated with the connection between unit j in layer l, and unit i in layer $l+1$. Also, $b_i^{(l)}$ is the bias associated with unit i in layer $l+1$, $a_i^{(l)}$ is the activation of unit i in layer l. For $l = 1$, we also use $a_i^{(1)} = x_i$ to denote the i-th input.

Define autoencoder $\mathbf{g}(\mathbf{x})$ as a superposition

$$\mathbf{a}^{(2)} = f(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}) - \text{encoder},$$
$$\mathbf{g}(\mathbf{x}) = f(\mathbf{W}^{(2)}\mathbf{a}^{(2)} + \mathbf{b}^{(2)}) - \text{decoder},$$

where $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)}$ are autoencoder parameter. We consider nonlinear autoencoders where f is a function from table **??**.

Also let $\mathbf{z}^{(l)}$ denote the total weighted sum of inputs to unit i in layer l, including the bias term (e.g., $\mathbf{z}^{(2)} = \mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}$), so that $\mathbf{a}^{(l)} = f(\mathbf{z}^{(l)})$.

To optimize the autocoder parameters $\mathbf{\Theta} = (\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)})$, we need to select the initial approximation for the parameters of each block separately, and then adjust the parameters of the whole model as a whole by the backpropagation method.

The autoencoder tries to learn a function $\mathbf{g}_{\mathbf{\Theta}}(\mathbf{x}) \approx \mathbf{x}$. In other words, it is trying to learn an approximation to the identity function, so as to output $\hat{\mathbf{x}}$ that is similar to $\mathbf{x}$

$$\|\mathbf{g}(\mathbf{x}|\mathbf{\Theta}) - \mathbf{x}\|_2^2 \to \min_{\mathbf{\Theta}}.$$

Let $\mathbf{\Theta}^{(k)} = (\mathbf{W}^{(k,1)}, \mathbf{W}^{(k,2)}, \mathbf{b}^{(k,1)}, \mathbf{b}^{(k,2)})$ denote the parameters $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)}$ for k-th autoencoder. Then the encoding step for the stacked autoencoder is given by running the encoding step of each layer in forward order:

$$\mathbf{a}^{(l)} = f(\mathbf{z}^{(l)}),$$
$$\mathbf{z}^{(l+1)} = \mathbf{W}^{(l,1)}\mathbf{a}^{(l)} + \mathbf{b}^{(l,1)}.$$

The decoding step is given by running the decoding stack of each autoencoder in reverse order:

$$\mathbf{a}^{(n+l)} = f(\mathbf{z}^{(n+l)}),$$
$$\mathbf{z}^{(n+l+1)} = \mathbf{W}^{(n-l,2)}\mathbf{a}^{(n+l)} + \mathbf{b}^{(n-l,2)}.$$

Thus, the stacked autoencoder result is

$$\mathbf{f}_{\mathbf{\Theta}}(\mathbf{x}) = \mathbf{a}_{\mathbf{\Theta}^{(n)}}(\dots\mathbf{a}_{\mathbf{\Theta}^{(1)}}(\mathbf{x})\dots),$$

where $\mathbf{\Theta} = (\mathbf{\Theta}^{(1)},\dots,\mathbf{\Theta}^{(n)})$ are stacked autoencoder parameters. The result of k-th encoder is $\mathbf{x}^{(k)} = \mathbf{a}_{\mathbf{\Theta}^{(k)}}(\mathbf{x}^{(k-1)}), \mathbf{x}^{(1)} = \mathbf{x}$.

The purpose is to minimize the following system

$$S(\mathbf{\Theta}^{(k)}, \mathbf{x}^{(k)}) = \|\mathbf{g}(\mathbf{x}^{(k)}|\mathbf{\Theta}^{(k)}) - \mathbf{a}^{(k)}\|_2^2 \to \min_{\mathbf{\Theta}}, \quad k \in \{1,\dots,n\}.$$

Thus,

$$\hat{\mathbf{\Theta}}^{(k)} = \underset{\mathbf{\Theta}}{\operatorname{argmin}} \frac{1}{2|\mathcal{L}|}\sum_{\mathbf{x}\in\mathcal{L}} S(\mathbf{\Theta}^{(k)}, \mathbf{x}^{(k)}), \quad k \in \{1,\dots,n\}.$$

3.2 Nonlinear PLS (nlPLS)

Extend the linear PLS model to its nonlinear form by replacing the linear inner relation (2) between the score vectors $\mathbf{t}$ and $\mathbf{u}$ by a nonlinear model

$$\mathbf{u} = g_x(\mathbf{t}) + \mathbf{z} = g_x(\mathbf{X}, \mathbf{v}_x) + \mathbf{z}, \tag{5}$$

where g represents a continuous nonlinear function, $\mathbf{h}$ denotes a vector of residuals and $\mathbf{w}$ is a vector of weights. The fuction $g_x(\mathbf{X})$ is a parametric matrix $\mathbf{G}_x(\mathbf{X}) = \mathbf{G}_x(\mathbf{X}, \mathbf{v}_x)$ where $\mathbf{v}_x$ is a solution of optimization PLS problem (1).

The second-order Taylor expansion of (5) has the form

$$\mathbf{u} = \mathbf{u}_0 + \left.\frac{\partial g_x}{\partial \mathbf{v}_x}\right|_0 \Delta\mathbf{v}_x,$$

where $\mathbf{u}_0 = g_x(\mathbf{t})$ is the value of g_x at the known value of $\mathbf{t}$. The following steps to compute $\Delta\mathbf{v}$ were suggested. The approach considers the difference $\mathbf{u} - \mathbf{u}_0 = \left.\frac{\partial g_x}{\partial \mathbf{v}}\right|_0 \Delta\mathbf{v}$ at the first step. Next, this leads to the following definition of a mismatch $\mathbf{e}$

$$\mathbf{e} = \mathbf{u} - \mathbf{u}_0 = \left.\frac{\partial g_x}{\partial \mathbf{v}_x}\right|_0 \Delta\mathbf{v}_x = \mathbf{J}_x\Delta\mathbf{v}_x,$$

where $\mathbf{J}_x$ consists of the partial derivatives $\left\{\left.\frac{\partial g}{\partial v_i}\right|_0\right\}_{i=1}^{N}$. From this mismatch relation $\Delta\mathbf{v}_x$ can be directly computed by regressing the mismatch $\mathbf{e}$ on $\mathbf{J}_v$

$$\Delta\mathbf{v}_x = (\mathbf{J}_x^{\mathsf{T}}\mathbf{J}_v)^{-1}\mathbf{J}_x^{\mathsf{T}}\mathbf{e}. \tag{6}$$

3.3 Curvelinear PLS (clPLS)

Extend the linear PLS model to its nonlinear form by replacing the linear inner relation (2) between the score vectors $\mathbf{t}$ and $\mathbf{u}$ by a nonlinear model

$$\mathbf{u} = g_x(\mathbf{t})\mathbf{w} + \mathbf{h} = g_x(\mathbf{X})\mathbf{w} + \mathbf{h}. \tag{7}$$

The function $g_x(\mathbf{X}) = \mathbf{G}_x$ is a nonparametric matrix and $\mathbf{w}$ is a vector of weights.

The following step to compute $\Delta\mathbf{w}$ was suggested:

$$\Delta\mathbf{w} = (\mathbf{G}_x^{\mathsf{T}}\mathbf{G}_x)^{-1}\mathbf{G}_x^{\mathsf{T}}\mathbf{u}. \tag{8}$$

3.4 nclPLS

Extend the nonlinear PLS model to its curvelinear form by replacing the linear inner relation (5) between the score vectors $g(\mathbf{t})$ and $\mathbf{u}$ by a nonlinear model

$$\mathbf{u} = g_x(\mathbf{t})\mathbf{w} + \mathbf{z} = g_x(\mathbf{X}, \mathbf{v}_x)\mathbf{w} + \mathbf{z}. \tag{9}$$

The previous steps in nlPLS (6) and clPLS (8) are proposed to be used in the nclPLS method. Finnaly, the whole nclPLS method starts from random initialization of $\mathbf{u}$ and the following steps are repeated until convergence

```
1.  $\mathbf{w} = \mathbf{X}^{\mathsf{T}}\mathbf{u}/(\mathbf{u}^{\mathsf{T}}\mathbf{u})$
2.  $\|\mathbf{w}\| \to 1$
3.  $\mathbf{t} = \mathbf{X}\mathbf{w}$
4.  fit $g_x(\cdot)$ using $\mathbf{u}, \mathbf{t}$
5.  $\mathbf{u}_0 = g_x(\mathbf{t})$
6.  $\mathbf{c} = \mathbf{Y}^{\mathsf{T}}\mathbf{u}_0/(\mathbf{u}_0^{\mathsf{T}}\mathbf{u}_0)$
7.  $\|\mathbf{c}\| \to 1$
8.  $\mathbf{e} = \mathbf{u} - \mathbf{u}_0$
9.  compute $\mathbf{J}_x$
10. $\Delta\mathbf{v}_x = (\mathbf{J}_x^{\mathsf{T}}\mathbf{J}_x)^{-1}\mathbf{J}_x^{\mathsf{T}}\mathbf{e}$
11. $\mathbf{u} = \mathbf{u}_0 + \mathbf{J}_x\Delta\mathbf{v}$
12. $\Delta\mathbf{w} = (\mathbf{G}_x^{\mathsf{T}}\mathbf{G}_x)^{-1}\mathbf{G}_x^{\mathsf{T}}\mathbf{u}$
13. $\mathbf{w} = \mathbf{w} + \Delta\mathbf{w}$
14. go to step 2.
```

4. Experiment

В рамках вычислительного эксперимента строится прогноз временных рядов. В ходе эксперимента сравниваются методы PLSR, нелинейных автоэнкодеров и нелинейной коррекции. Сравнение проводится на реальных данных объемов потребления электроэнергии в Польше и на реальных данных ECoG сигналов.

4.1 "Energy-Weather" Dataset

The computational experiments demonstrated in this section are based on the Energy-Weather data set (Available: http://gdudek.el.pcz.pl/varia/stlf-data). The dataset consists of the Polish electricity load time series and weather time series in Warsaw (Longtitude: 21.25, Latitude: 52.30, Elevation: 94). Energy time series contain hourly records (total of 52512 observations), while weather time series were measured daily and contain 2188 observations. The multiscale time series correspond to the period of 1999 to 2004. The results observed on this data set are illustrative of the proposed framework since the data set contains the time series that are both multiscale and have various nature.

Here are results of the baseline method: for time series the next forecasted value is predicted with the most recent observed value.

4.2 "ECoG" Dataset

The monkey was tracking food rewards with the hand contralateral to the implant side. ECoG data and motion data were recorded simultaneously during the task ($r$eference) (Available: http://neurotycho.org/social-competition-task). There was no eye tracking. ECoG and motion data were sampled at 1KHz and 120Hz, respectively, with time stamps synchronized.

Data format:

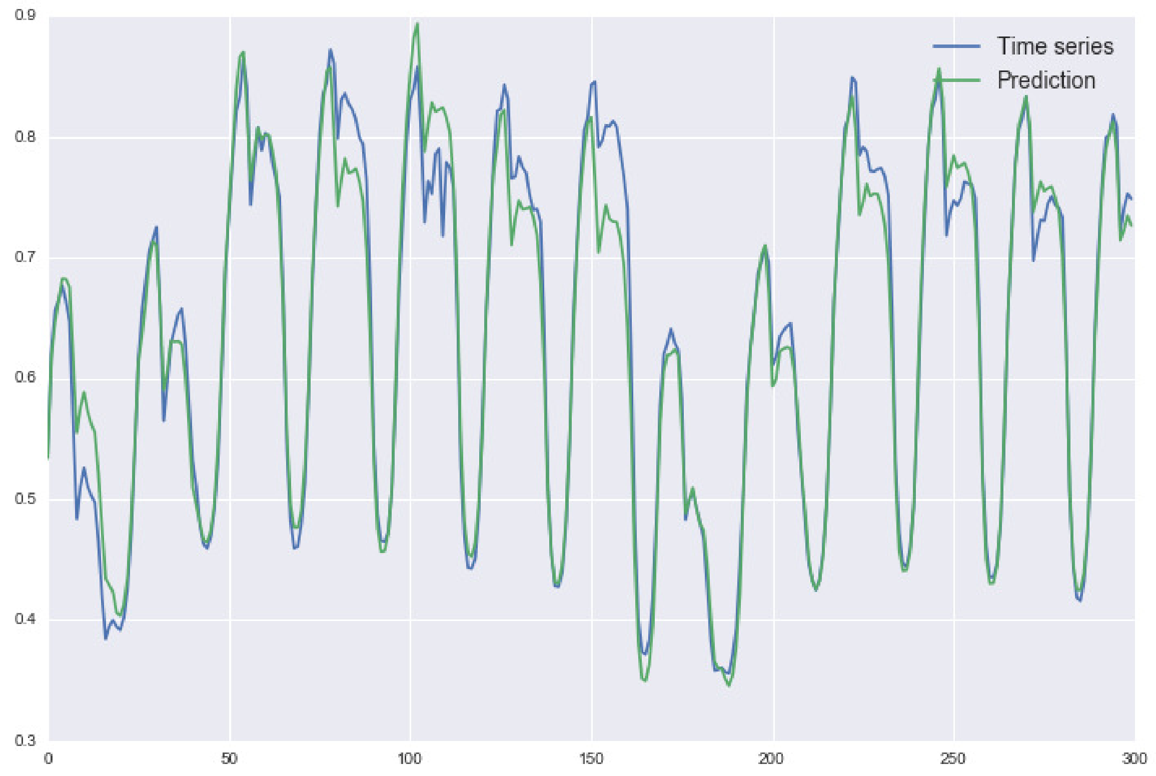


Рис. 1. PLS Regression, “Energy-Weather” Dataset

— ECoG_chN.mat
  ECoGData_chN: ECoG signal (μV) recorded from electrode N (1-64), sampled at 1kHZ. The Location of electrode is documented in “B.png”.
— ECoG_time.mat
  ECoGTime: ECoGTime is a one row-vector contains Time-stamps with the same length as ECoGData_chN.
— Motion.mat
  MotionData: MotionData is a time $\times$ marker matrix containing $3D$ position of marker (index 1, LSHO: left shoulder, index 2, LELB: left elbow, index 3, LWRI: left wrist, index 4, RSHO: right shoulder, index 5, RELB: right elbow, index 6, RWRI: right wrist).
  MotionTime: MotionTime contains the corresponding time-stamps.

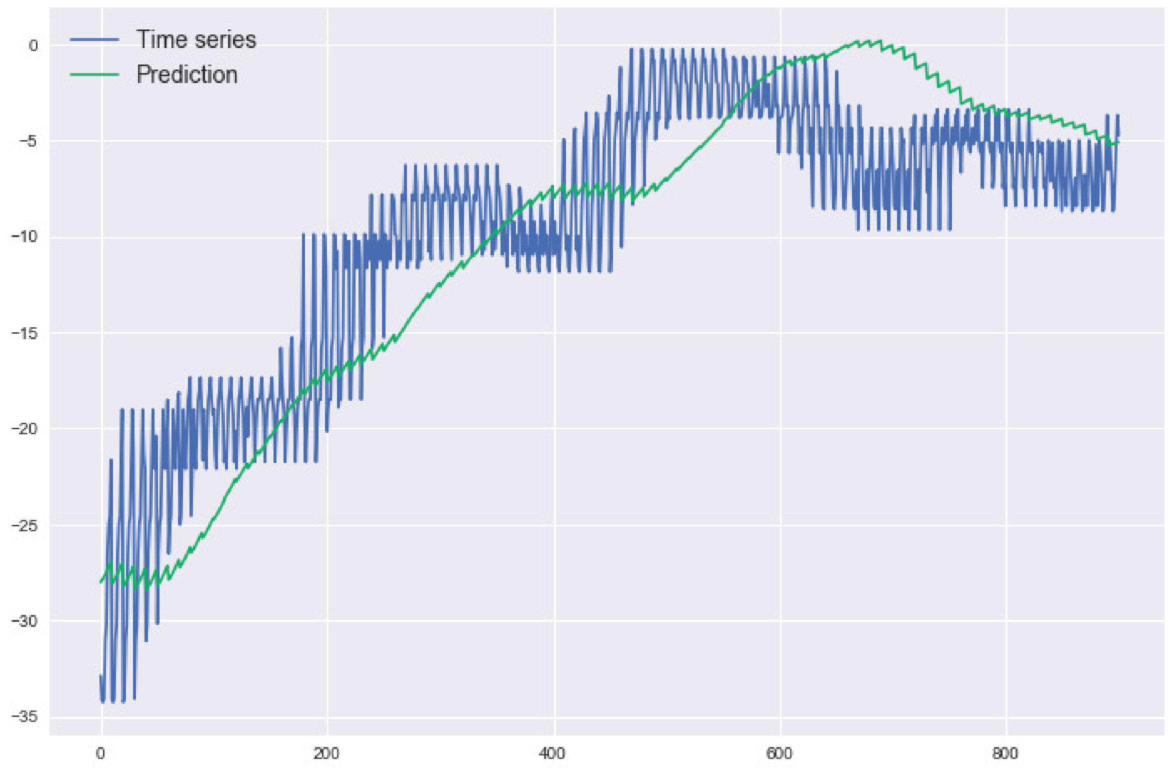


Рис. 2. PLS Regression, “ECoG” Dataset, 1 electrode

References

[1] Adolph, K. E. *Learning to learn.*

[2] Evgeniou, A. A. T., and M. Pontil. 2007. Multi-task feature learning. *Advances in Neural Information Processing Systems* 19(January): 41.

[3] Varnek, A., and I. Baskin. 2012. Machine learning methods for property prediction in chemoinformatics: Quo Vadis? *Journal of Chemical Information and Modeling* 52(6):1413–1437. doi: 10.1021/ci200409x.

[4] Caruana, R., and V. R. de Sa. 2003. Benefitting from the Variables that Variable Selection Discards. *Journal of Machine Learning Research* 3(7-8):1245–1264. doi: 10.1162/153244303322753652.

[5] Abdi, H. 2003. Partial Least Squares (PLS) Regression. *Encyclopedia for research methods for the social sciences* 792–795. doi: http://dx.doi.org/10.4135/9781412950589.n690.

[6] Geladi, P. 1988. Notes on the history and nature of partial least squares (PLS) modelling. *Journal of Chemometrics* 2(January):231–246. doi: http://doi.wiley.com/10.1002/cem.1180020403.

[7] Höskuldsson, A. 1988. PLS regression. *Journal of Chemometrics* 2(August 1987):581–591. doi: 10.1002/cem.1180020306.

[8] Lehky, S. R., R. Kiani, H. Esteky, and K. Tanaka. 2014. Dimensionality of object representations in monkey inferotemporal cortex. *Neural computation* 1872(10):1840–1872. doi: 10.1162/NECO.

[9] Lu, W.-C., N.-Y. Chen, G.-Z. Li, and J. Yang. 2004. Multitask Learning Using Partial Least Squares Method. *Proceedings of the Seventh International Conference on Information Fusion; International Society of Information Fusion* 1:79–84.

[10] Frank, I. E. 1990. A nonlinear PLS model. *Chemometrics and Intelligent Laboratory Systems* 8(2):109–119. doi: 10.1016/0169-7439(90)80128-S.

[11] Mcavovt, J., and C. Process. 1992. USING 16(4):379–391.

[12] Yan, X. F., D. Z. Chen, and S. X. Hu. 2003. Chaos-genetic algorithms for optimizing the operating conditions based on RBF-PLS model. *Computers and Chemical Engineering* 27(10):1393–1404. doi: 10.1016/S0098-1354(03)00074-7.

[13] Zhou, Y. P., J. H. Jiang, W. Q. Lin, L. Xu, H. L. Wu et al. 2007. Artificial neural network-based transformation for nonlinear partial least-square regression with application to QSAR studies. *Talanta* 71(2):848–853. doi: 10.1016/j.talanta.2006.05.058.

[14] Xuefeng, Y. 2010. Hybrid artificial neural network based on BP-PLSR and its application in development of soft sensors. *Chemometrics and Intelligent Laboratory Systems* 103(2):152–159. doi: 10.1016/j.chemolab.2010.07.002.

[15] Bulut, E., and E. Egrioglu. 2014. A New Partial Least Square Method Based on Elman Neural Network 4(4):154–158. doi: 10.5923/j.ajis.20140404.05.