Mathematical methods of forecasting

Intelligent systems, Phystech

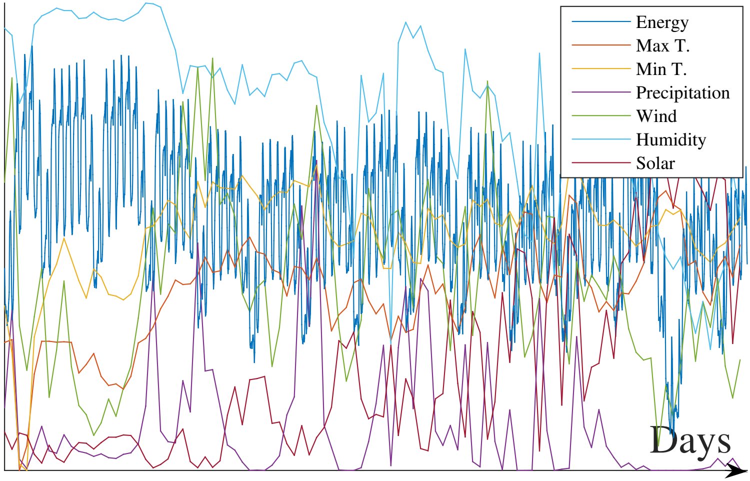
2022

Internet of things

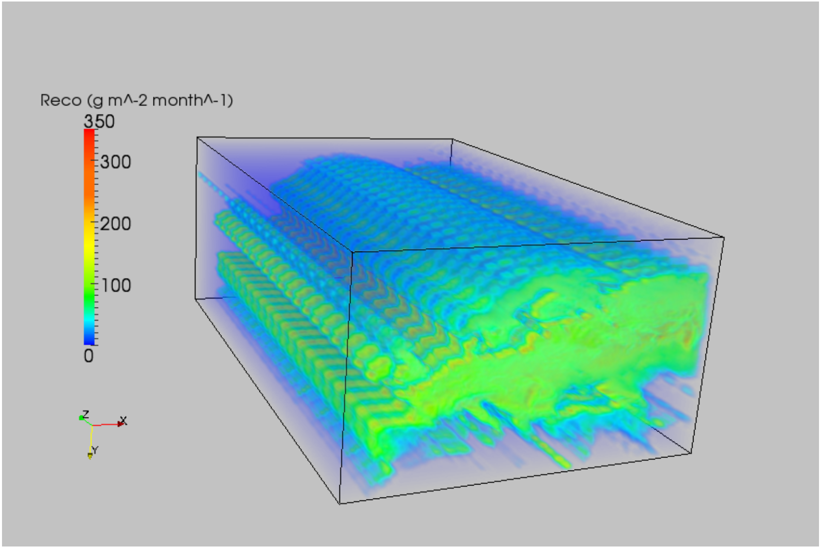
IoT is the networking of devices (portables, vehicles, buildings) embedded with sensors and software.

- Environment and energy monitoring
- Medical and health monitoring
- Consumer support, sales monitoring
- Urban management and manufactiring

Example of a multiscale dataset

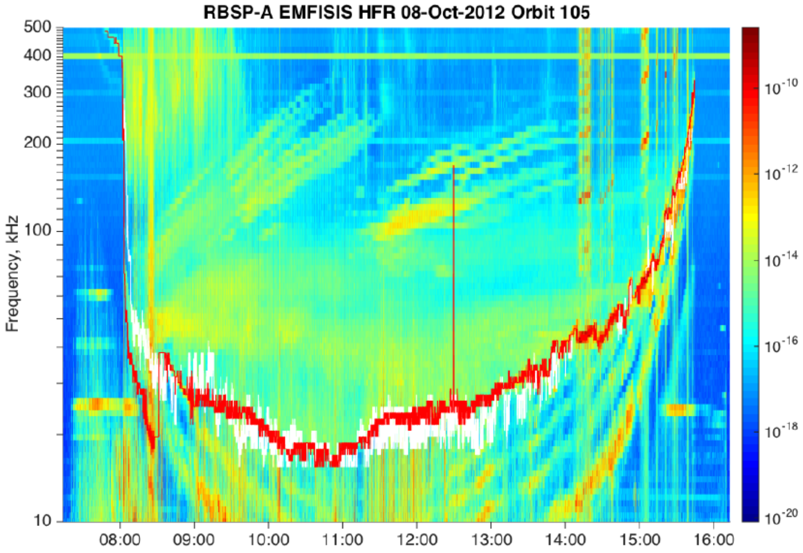


Example of a spatio temporal dataset



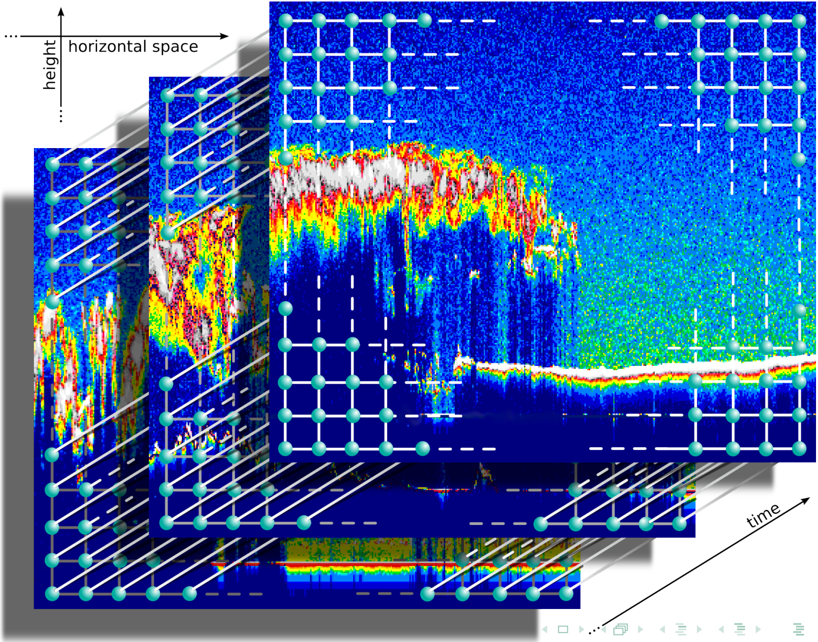
Monthly Ecosystem Respiration by M. Reichstein, GHG-Europe.

Example of a spatio temporal dataset

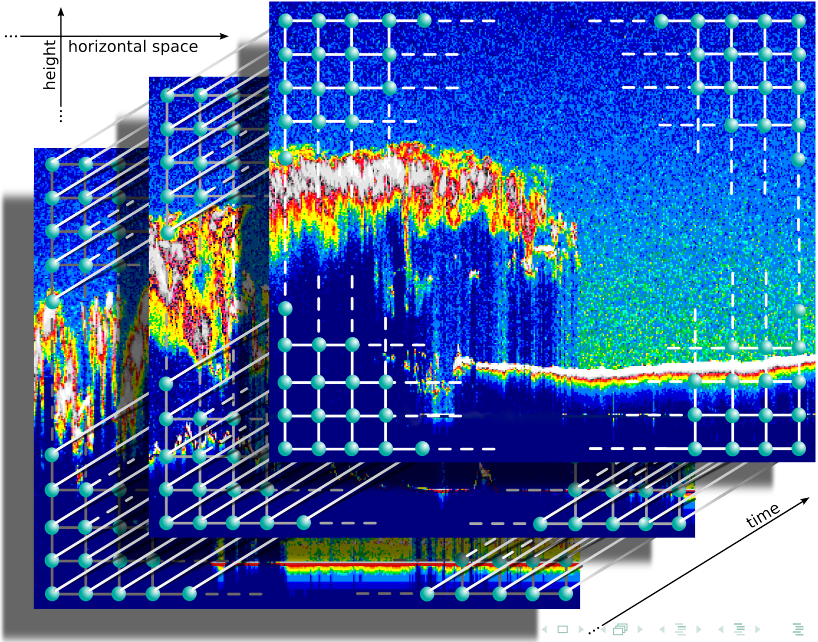


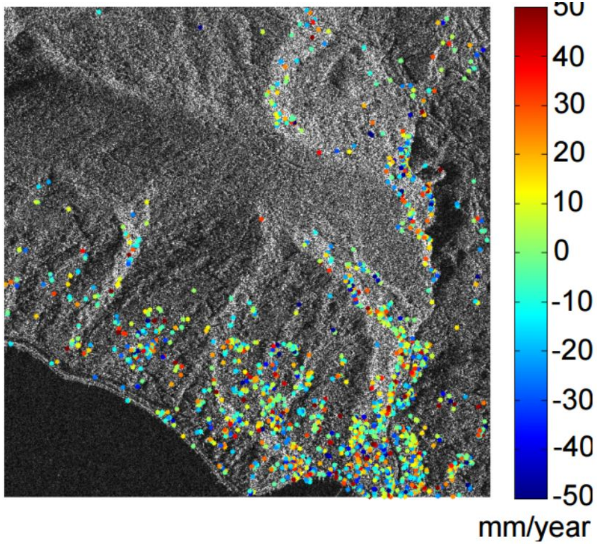
Electric field measurement, the Van Allen probes by I. Zhelavskaya, Skoltech.

Example of a spatio temporal dataset

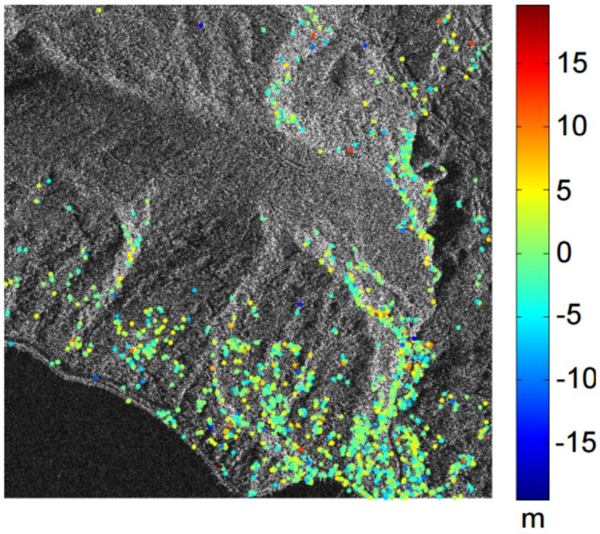


Example of a spatio temporal dataset





Скорость движения вдоль направления наблюдения



Погрешность в высоте

Рудаков К.В., Адуенко А.А., Рейер И.А., Василейский А.С., Карелов А.И., Стрижов В.В. Алгоритмы выделения и совмещения устойчивых отражателей на спутниковых снимках // Компьютерная оптика, 2015.

The periodic components of the multivariate time series

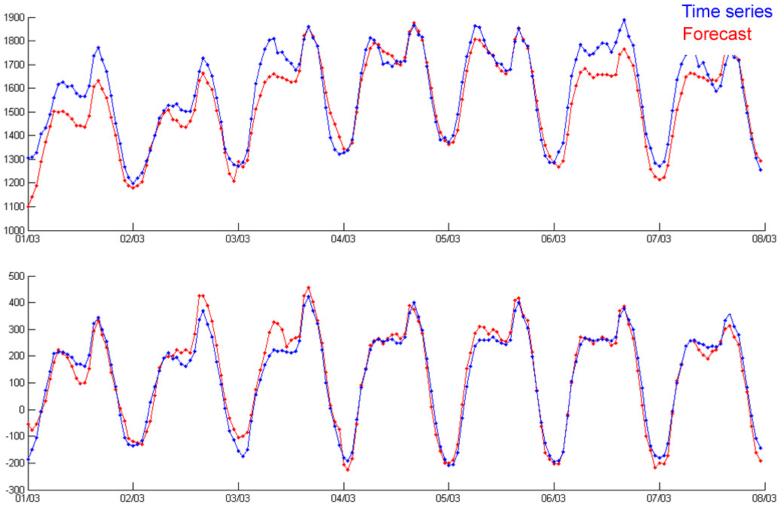
The time series:

- energy price,
- consumption,
- daytime,
- temperature,
- humidity,
- wind force,
- holiday schedule.

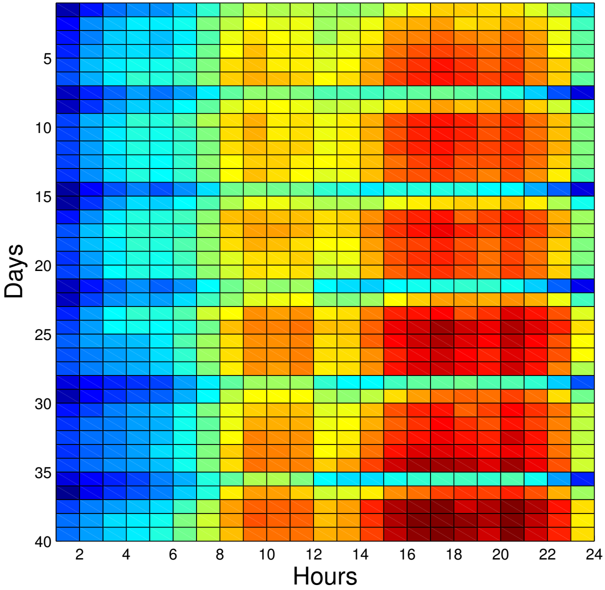
Periods:

- one year seasons (temperature, daytime),
- one week,
- one day (working day, week-end),
- a holiday,
- aperiodic events.

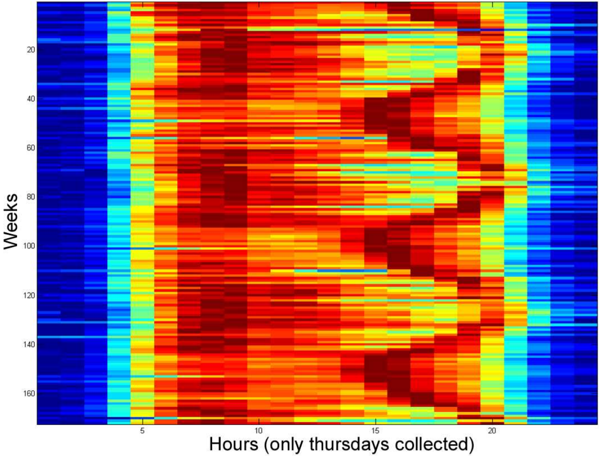
Energy consumption one-week forecast for each hour



The autoregressive matrix, five week-ends



Daily similarity



The autoregressive matrix and the linear model

$$\underset{(m+1)\times(n+1)}{\mathbf{X}^*} = \left(\begin{array}{l|lll} s_T & s_{T-1} & \dots & s_{T-\kappa+1} \\ \hline & & & \\ s_{(m-1)\kappa} & s_{(m-1)\kappa-1} & \dots & s_{(m-2)\kappa+1} \\ \dots & \dots & \dots & \dots \\ s_{n\kappa} & s_{n\kappa-1} & \dots & s_{n(\kappa-1)+1} \\ \dots & \dots & \dots & \dots \\ s_{\kappa} & s_{\kappa-1} & \dots & s_1 \end{array}\right).$$

In a nutshell,

$$\mathbf{X}^* = \left[\begin{array}{c|c} \underset{1\times 1}{s_T} & \underset{1\times n}{\mathbf{x}_{m+1}} \\ \hline \underset{m\times 1}{\mathbf{y}} & \underset{m\times n}{\mathbf{X}} \end{array}\right].$$

In terms of linear regression:

$$\mathbf{y} = \mathbf{X}\mathbf{w},$$

$$y_{m+1} = s_T = \mathbf{w}^{\mathrm{T}}\mathbf{x}_{m+1}^{\mathrm{T}}.$$

Model generation

Introduce a set of the primitive functions $\mathfrak{G} = \{g_1, \ldots, g_r\}$, for example $g_1 = 1$, $g_2 = \sqrt{x}$, $g_3 = x$, $g_4 = x\sqrt{x}$, etc.

The generated set of features $\mathbf{X} =$

$$\left(\begin{array}{lcl|c|lcl}
g_1 \circ s_{T-1} & \ldots & g_r \circ s_{T-1} & \ldots & g_1 \circ s_{T-\kappa+1} & \ldots & g_r \circ s_{T-\kappa+1} \\
\hline
g_1 \circ s_{(m-1)\kappa-1} & \ldots & g_r \circ s_{(m-1)\kappa-1} & \ldots & g_1 \circ s_{(m-2)\kappa+1} & \ldots & g_r \circ s_{(m-2)\kappa+1} \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
g_1 \circ s_{n\kappa-1} & \ldots & g_r \circ s_{n\kappa-1} & \ldots & g_1 \circ s_{n(\kappa-1)+1} & \ldots & g_r \circ s_{n(\kappa-1)+1} \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
g_1 \circ s_{\kappa-1} & \ldots & g_r \circ s_{\kappa-1} & \ldots & g_1 \circ s_1 & \ldots & g_r \circ s_1
\end{array}\right).$$

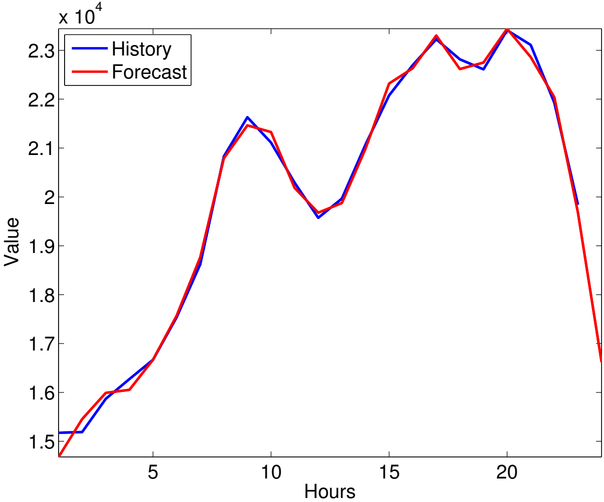
Kolmogorov-Gabor polynomial as a variant for model generation

$$y = w_0 + \sum_{i=1}^{UV} w_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} x_i x_j + \cdots + \sum_{i=1}^{n} \ldots \sum_{z=1}^{n} w_{i\ldots z} x_i \ldots x_z,$$

where the coefficients

$$\mathbf{w} = (w_0, w_i, w_{ij}, \ldots, w_{i\ldots z})_{i,j,\ldots,z=1,\ldots,n}.$$

The one-day forecast (an example)



The function $y = f(\mathbf{x}, \mathbf{w})$ could be a linear model, neural network, deep NN, SVN, ...

Ill-conditioned matrix, or curse of dimensionality

Assume we have hourly data on price/consumption for three years. Then the matrix $\underset{(m+1)\times(n+1)}{\mathbf{X}^*}$ is

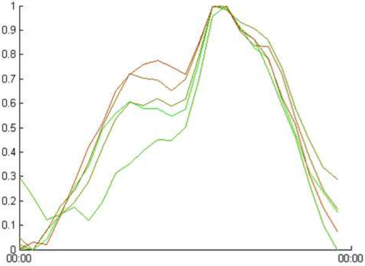
$$156 \times 168, \text{ in details: } 52\text{w} \cdot 3\text{y} \times 24\text{h} \cdot 7\text{d};$$

- for 6 time series the matrix $\mathbf{X}$ is 156×1008,
- for 4 primitive functions it is 156×4032,

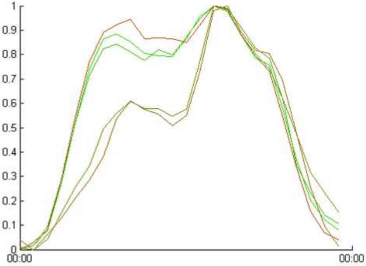
$$m << n.$$

The autoregressive matrix could be considered as *ill-conditioned* and *multi-correlated*. The model selection procedure is required.

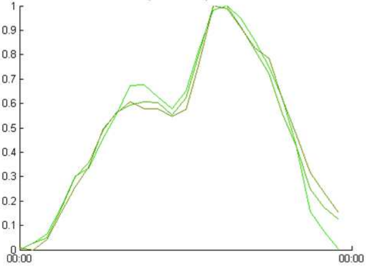
Similarity of daily consumption



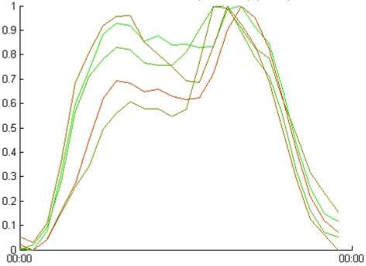
Five days of one week



One day of five consequent weeks

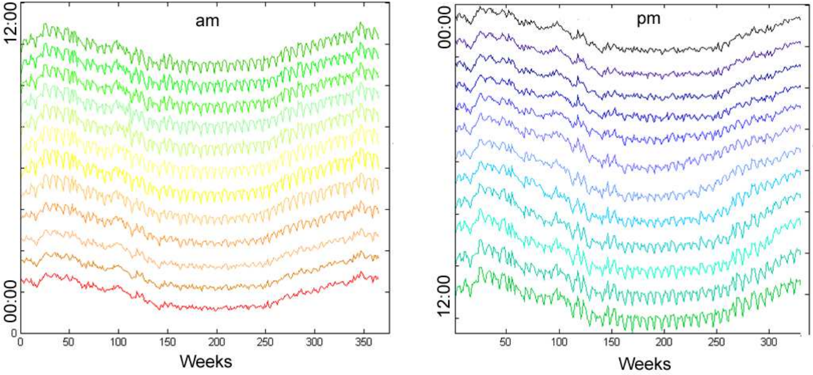


The same weekday and month of three years

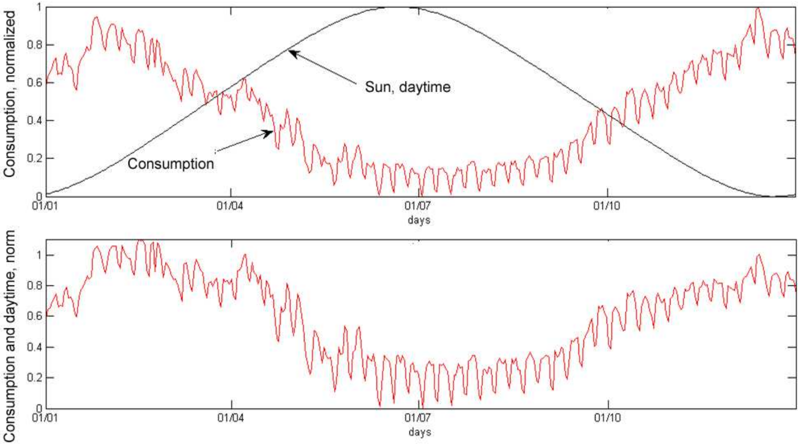


The same weekday of five months

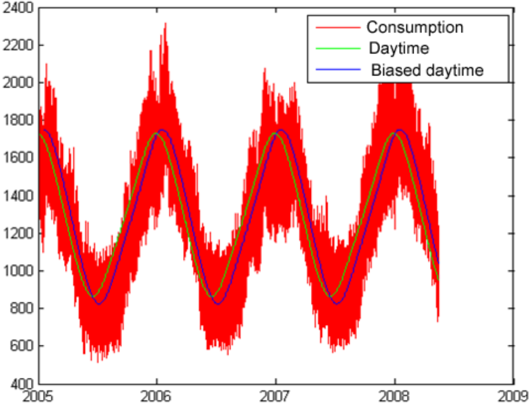
One-hour line, day-by-day during a year: autoregressive analysis



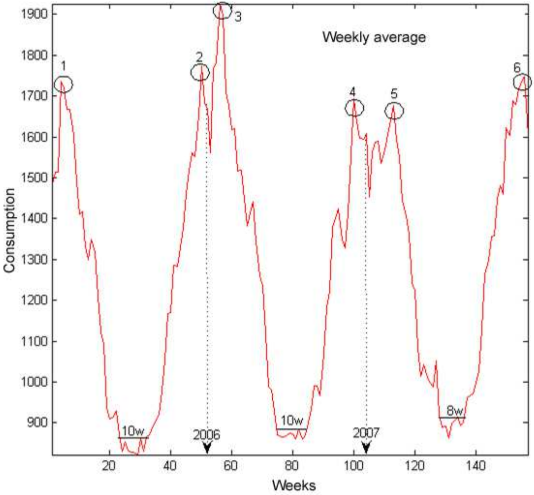
Sunrise bias: one-year daytime and consumption



Biased and original daytime to fit consumption over years

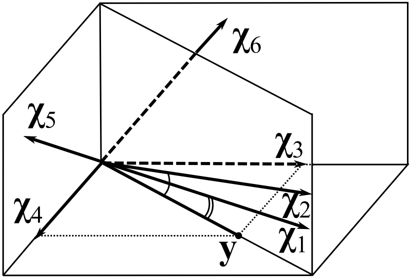


Structure of energy consumption



Selection of a stable set of features of restricted size

The sample contains multicollinear χ_1, χ_2 and noisy χ_5, χ_6 features, columns of the design matrix $\mathbf{X}$. We want to select two features from six.

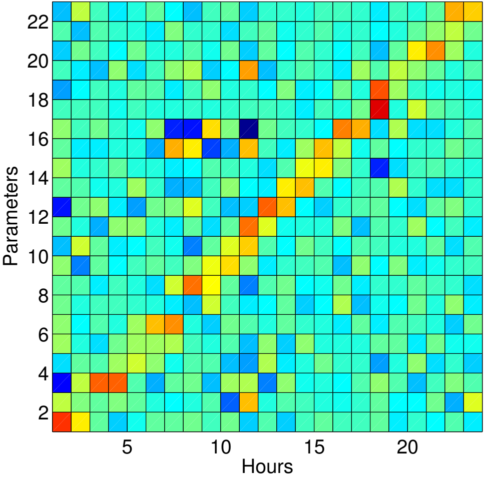


Stability and accuracy for a fixed complexity

The solution: $\chi_3, \chi_4$is an orthogonal set of features minimizing the error function.

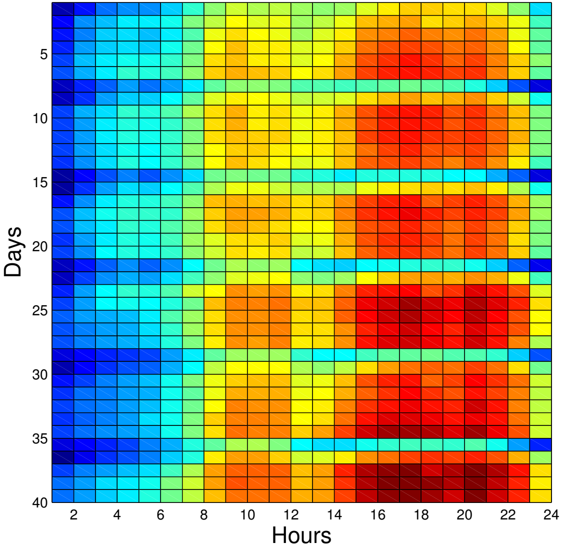
How many parameters must be used to forecast?

The color shows the value of a parameter for each hour.



Estimate parameters $\mathbf{w}(\tau) = (\mathbf{X}^\mathsf{T}\mathbf{X})^{-1}\mathbf{X}^\mathsf{T}\mathbf{y}$, then calculate the sample $s(\tau) = \mathbf{w}^\mathsf{T}(\tau)\mathbf{x}_{m+1}$ for each τ of the next $(m+1$-th$)$ period.

Матрица авторегрессии



$$\mathbf{X}^* = \left[\begin{array}{c|c} \underset{1\times 1}{s_T} & \underset{1\times n}{\mathbf{x}_{m+1}} \\ \hline \underset{m\times 1}{\mathbf{y}} & \underset{m\times n}{\mathbf{X}} \end{array}\right],$$

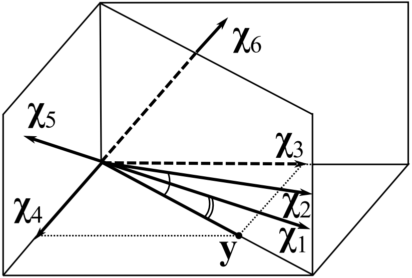
$$\hat{\mathbf{y}} = \mathbf{f}(\mathbf{X}, \mathbf{w}) = \mathbf{X}\mathbf{w},$$

$$y_{m+1} = s_T = \mathbf{w}^{\mathsf{T}}\mathbf{x}_{m+1}^{\mathsf{T}}.$$

Нейчев Р.Г., Катруца А.М., Стрижов В. Выбор оптимального набора признаков из мультикоррелирующего множества в задаче прогнозирования // Заводская лаборатория. Диагностика материалов, 2016.

Selection of a stable set of features of restricted size

The sample contains multicollinear χ_1, χ_2 and noisy χ_5, χ_6 features, columns of the design matrix $\mathbf{X}$. We want to select two features from six.



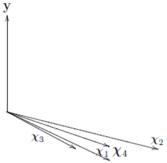
Stability and accuracy for a fixed complexity

The solution: $\chi_3, \chi_4$is an orthogonal set of features minimizing the error function.

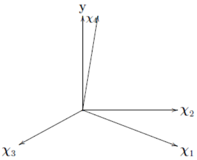
Algorithms: Add/Del, GMDH, Stepwise, Ridge, Lasso, Stagewise, FOS, LARS, Genetics, ...

Гипотезы и аксиоматика не противоречат измеряемым данным.

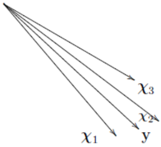
Multicollinear features to forecast: possible configurations



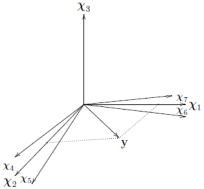
Inadequate and correlated



Adequate and random



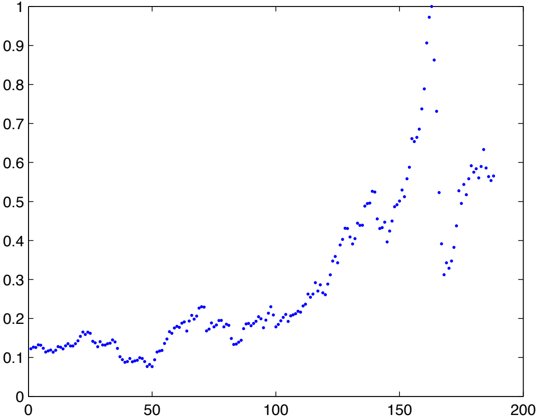
Adequate and redundant



Adequate and correlated

Katrutsa A.M., Strijov V.V. Stresstest procedure for feature selection algorithms // Chemometrics and Intelligent Laboratory Systems, 2015, 142 : 172-183.

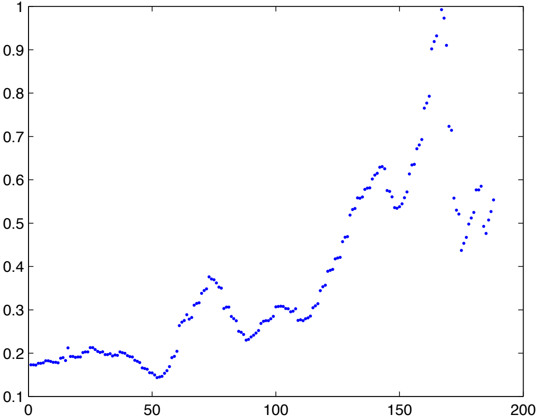
Time series with a bubble, example 1



The World Bank. Global Economic Monitor 2010.

http://data.worldbank.org/data-catalog/global-economic-monitor.

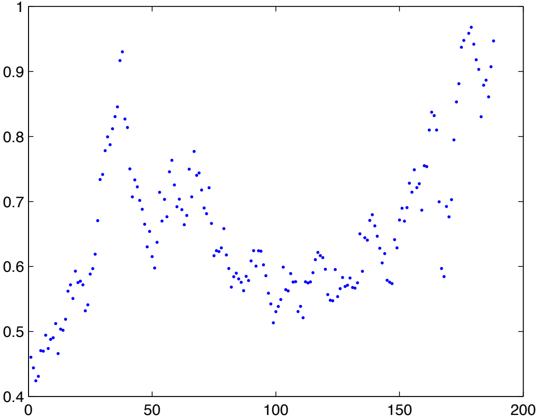
Time series with a bubble, example 2



The World Bank. Global Economic Monitor 2010.

http://data.worldbank.org/data-catalog/global-economic-monitor.

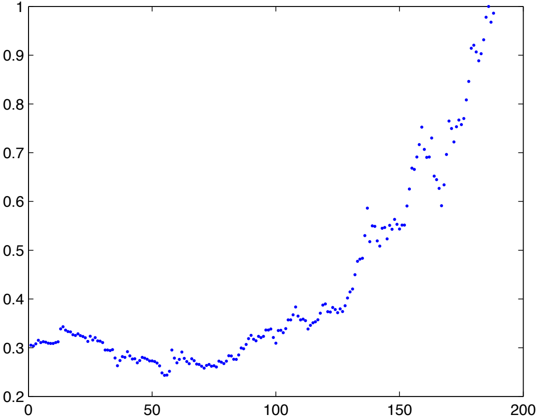
Time series with no bubble, example 3



The World Bank. Global Economic Monitor 2010.

http://data.worldbank.org/data-catalog/global-economic-monitor.

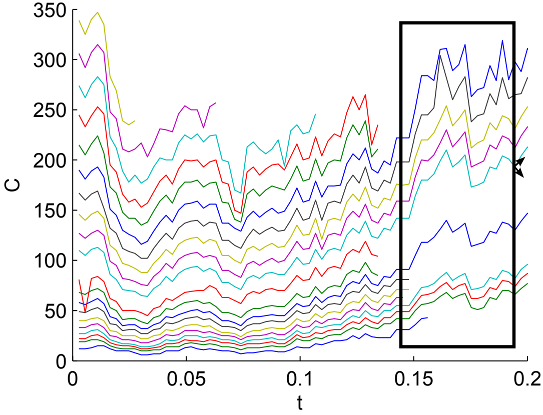
Time series with no bubble, example 4



The World Bank. Global Economic Monitor 2010.

http://data.worldbank.org/data-catalog/global-economic-monitor.

Time series trends/events forecasting, example



Event forecasting; One must forecast $a_{1,T+1} \in \mathcal{M}$.

There are N time series of length T (denote the element $a_{n,t} \in \mathcal{M}$) as the matrix

$$A = \begin{matrix} a_{1,1} & a_{1,2} & \cdots & a_{1,T-1} & a_{1,T} & a_{1,T+1} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,T-1} & a_{2,T} & a_{2,T+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a_{N,1} & a_{N,2} & \cdots & a_{N,T-1} & a_{N,T} & a_{N,T+1} \end{matrix}$$

Denote Δ the time-lag and for the time series $[a_{1,t}], t \in \{\Delta + 1, \dots, T\}$ form the matrix

$$A_t = \begin{matrix} a_{1,t-\Delta} & a_{1,t-\Delta+1} & \cdots & a_{1,t-2} & a_{1,t-1} \\ a_{2,t-\Delta} & a_{2,t-\Delta+1} & \cdots & a_{2,t-2} & a_{2,t-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{N,t-\Delta} & a_{N,t-\Delta+1} & \cdots & a_{N,t-2} & a_{N,t-1} \end{matrix}$$

and vectorize it to obtain the sample $\mathbf{x}_t$

$$\mathbf{x}_t = [a_{1,t-\Delta}, a_{2,t-\Delta}, \dots, a_{N,t-\Delta}, a_{1,t-\Delta+1}, \dots, a_{N,t-1}]^T.$$

Set $y_t \equiv a_{1,t}$.

Event forecasting is a classification problem

Introduce the data set $D = (X, \mathbf{y})$, where

$$X = \begin{bmatrix} \mathbf{x}_1^\mathsf{T} \\ \mathbf{x}_2^\mathsf{T} \\ \vdots \\ \mathbf{x}_{T-\Delta}^\mathsf{T} \end{bmatrix} \quad \text{and} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{T-\Delta} \end{bmatrix}.$$

Treat this classification problem as the logistic regression

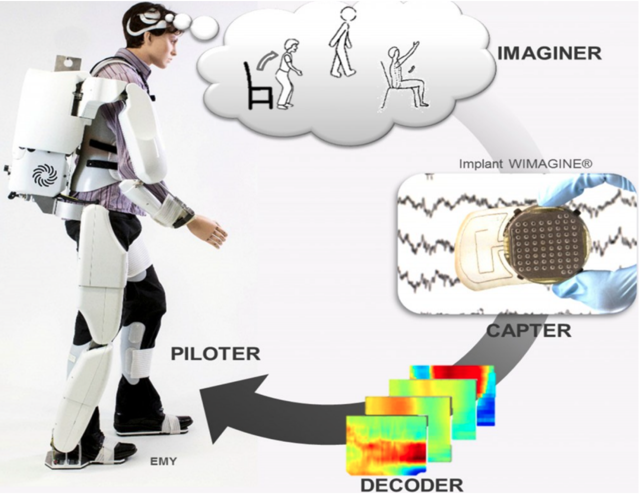
$$\mathbf{f}(\mathbf{w}, X) = \frac{1}{1 + \exp(-X\mathbf{w})} \to \mathbf{y}$$

or as another classification problem.
The error function

$$S(\mathbf{w}) = \sum_{i \in \mathcal{I}} y_i \ln f(\mathbf{w}, \mathbf{x}_i) + (1 - y_i) \ln (1 - f(\mathbf{w}, \mathbf{x}_i)).$$

The Brain-Computer interface project

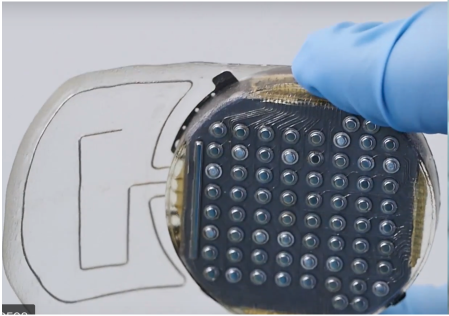
aims to develop compensating systems that will help people with a severe motor control disability recover mobility.

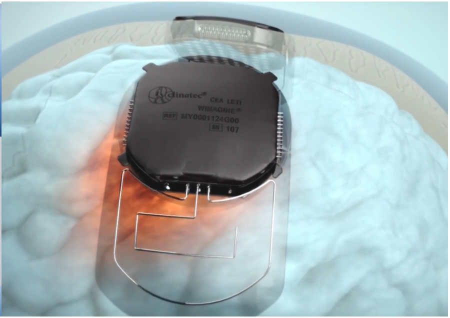


www.clinatec.fr

Implant WIMAGINE to measure ECoG

detects the electrical activity in the motor cortex with “minimally-invasive implantation in the cranium and, over the long term, to measure electrocorticograms thanks to an array of electrodes in contact with the dura mater.”

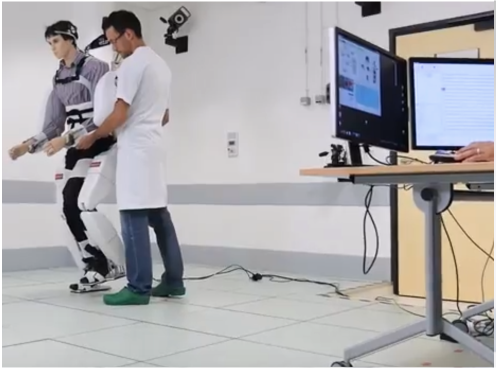


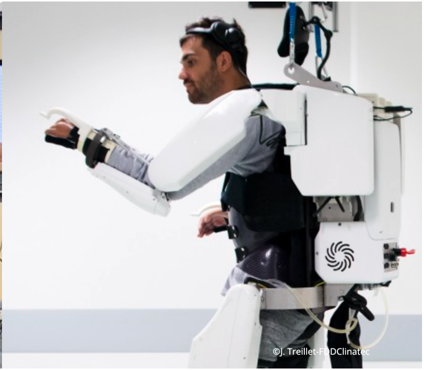


Eliseyev, A., and Aksenova, T. Stable and artifact-resistant decoding of 3D hand trajectories from ECoG signals using the generalized additive model // J. Neural Eng. 2014.

Clinatec, Centre de recherche Edmond J. Safra

"The subject placed inside the exoskeleton can drive it by imagining movements as if they were making the movement themself."

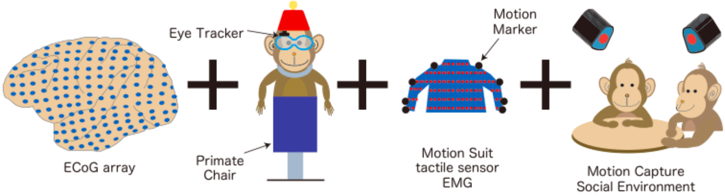


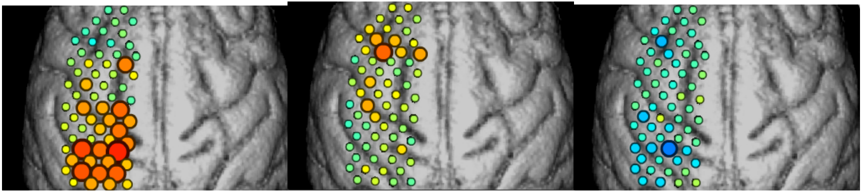


www.clinatec.fr

Анализ пространственно-временных данных

Прогнозирование движений конечностей по электрокортикограммам (ECoG) с учетом движений глаз



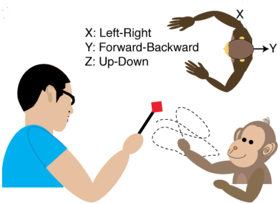


http://neurotycho.org/

Макарчук Г. И., Задаянчук А. И., Стрижов В. В. Преобразования временных рядов для декодирования движения руки с помощью ECoG сигналов (electrocorticographic signals) у обезьян // Машинное обучение и анализ данных, 2016.

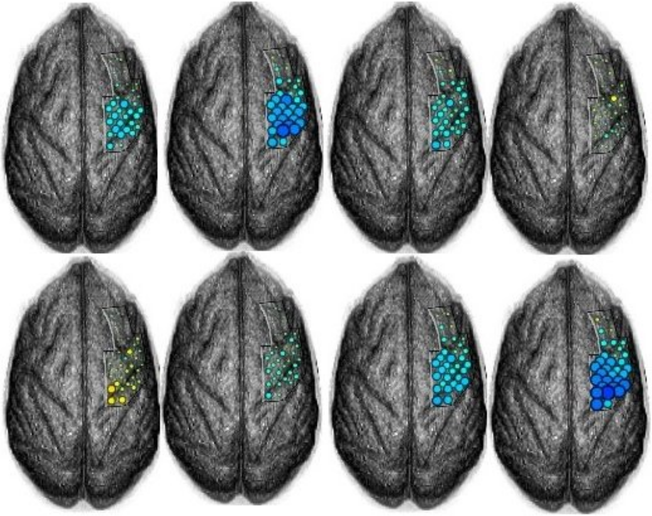
Neurotycho data, foodtracking task

A monkey is tracking food rewards with the hand contralateral to the implant side. The experimenter demonstrated foods at random locations at a distance of 20 cm for the monkey at random time intervals 3-4 times per minute, and the monkey grasped the foods



- Subdural (32 electrodes): 2 monkeys, 3 and 5 records, taken within 7 months.
- Each record measures about 1000 seconds with ECoG and motion data (wrists, elbows and shouders) sampled at 1KHz and 120Hz, respectively.

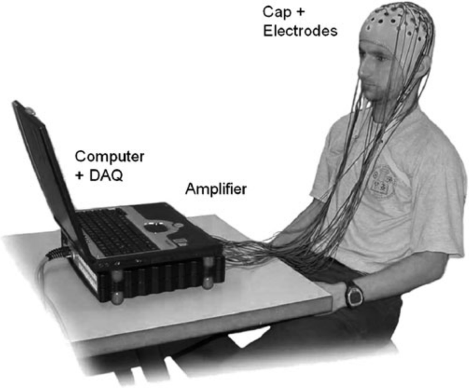
Демонстрация изменения физической активности



Neurotycho, RIKEN Brain Science Institute, Saitama, Japan

Electroencephalogram-based BCI

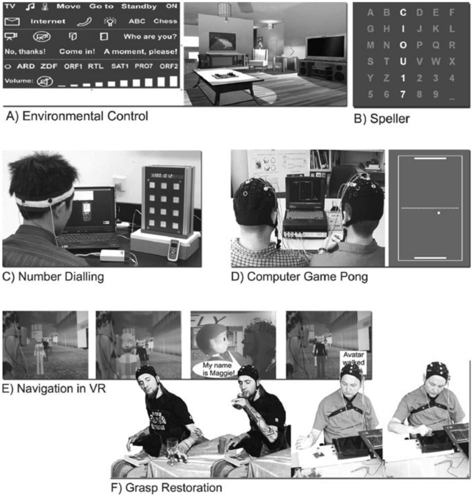
“consists of an electrode cap with electrodes, cables that transmit the signals from the electrodes to the biosignal amplifier, a device that converts the brain signals from analog to digital format. . . ”



Graimann B. et al. (eds.) Brain–Computer Interfaces, a gentle introduction. Springer, 2010.

Examples of BCI applications

P300 synchronous, blinking stimulus, imagination



A) Environmental Control

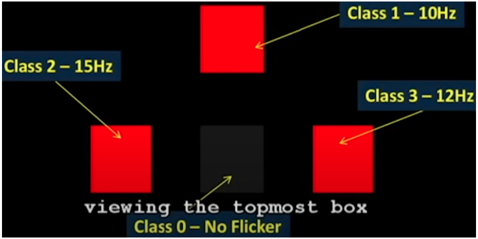
B) Speller

C) Number Dialling

D) Computer Game Pong

E) Navigation in VR

F) Grasp Restoration



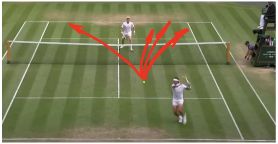
Graimann B. et al. (eds.) Brain–Computer Interfaces, a gentle introduction. Springer, 2010.

Cattan G., Andreev A., Visinoni E. Recommendations for integrating a P300-based brain–computer interface in virtual reality environments for gaming: An Update // Computers, 2020

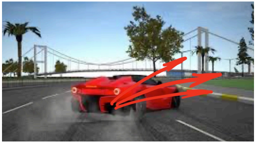
Control problem for game[2] activities

$$\begin{tikzcd}
\text{Control} \arrow[d, dash] & & \text{Stimulus} \arrow[r, dash] & z \arrow[d, "\text{Exogenous}"] \\
u = C(\varepsilon) \arrow[rrr, "\text{Control}", "\text{Action}"'] & & & \dot{y} = f(x, y, u, z) \arrow[d, "\text{Measurement}"] \\
\hat{f}, \varepsilon \arrow[u, "\text{Decision}"] & & & \hat{x}, \hat{y}, \hat{z} \arrow[lll, "\text{Modelling}"']
\end{tikzcd}$$

Game with opponent



Game with nature



Proportional integral derivative PID controller

$$u(t) = C_{\mathrm{p}}\varepsilon(t) + C_{\mathrm{i}} \int_0^t \varepsilon(\tau)\, d\tau + C_{\mathrm{d}} \frac{d\varepsilon(t)}{dt}$$

with reinforcement[1] learning $\varepsilon(t) = f(t) - y(t)$.

[1] over space and time to say nothing of a player

[2] repeating actions with goal

Multi-modal signals of brain-computer interfaces

Stimulus

Video 30Hz × 1280 × 1024

Sound 48KHz × 2

Environment ∼ 10Hz

Control

ECoG 1000Hz × 64

EEG 1000Hz × 64

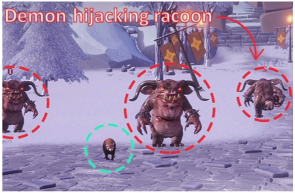
fMRI 5Hz × 1024^3

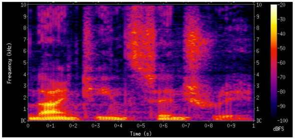
Action

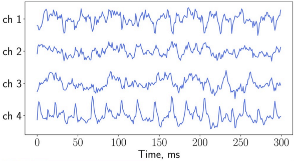
Eye track 30Hz × 3

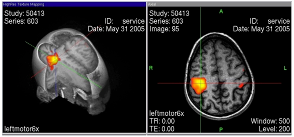
IMU Limb 200Hz × 6

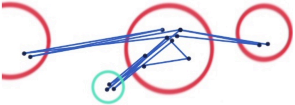
MEG 100Hz × 12

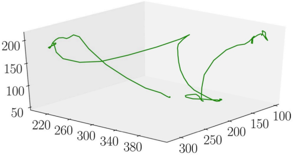


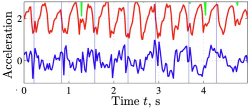










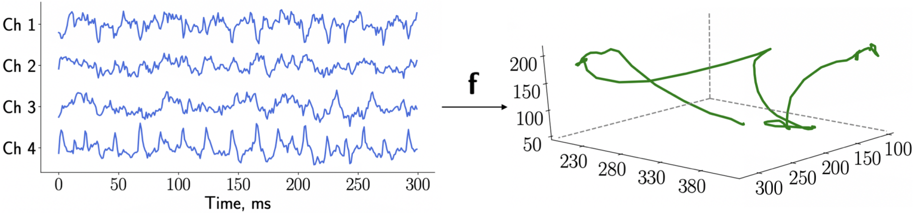


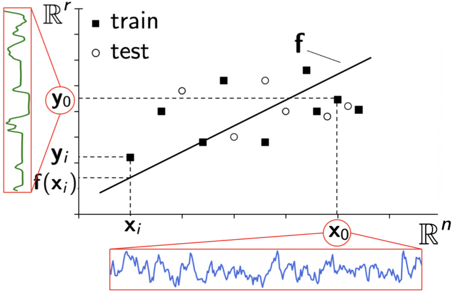
Types of Machine Learning problems to model BCI

Inputs	Classification	Regression	Decoding	Clustering
Video			x^{new}	
Sound				✓
ECoG	✓ P300		x	
EEG	✓		x?	
fMRI				✓
IMU	✓	✓	y, y^{new}	✓
MEG	✓			

Why decoding can not be replaced for serial classification?

Ensemble of models for brain computer interface





$$\begin{tikzcd}
\mathbf{x} \in \mathbb{R}^n \arrow[rr, "\mathbf{f}"] \arrow[dr, bend right, "\mathbf{W}"'] & & \mathbf{y} \in \mathbb{R}^r \arrow[dl, bend left, "\mathbf{C}"] \\
& \mathbf{t}, \mathbf{u} \in \mathbb{R}^\ell \arrow[ul, bend right, "\mathbf{P}"'] \arrow[ur, bend left, "\mathbf{Q}"] &
\end{tikzcd}$$

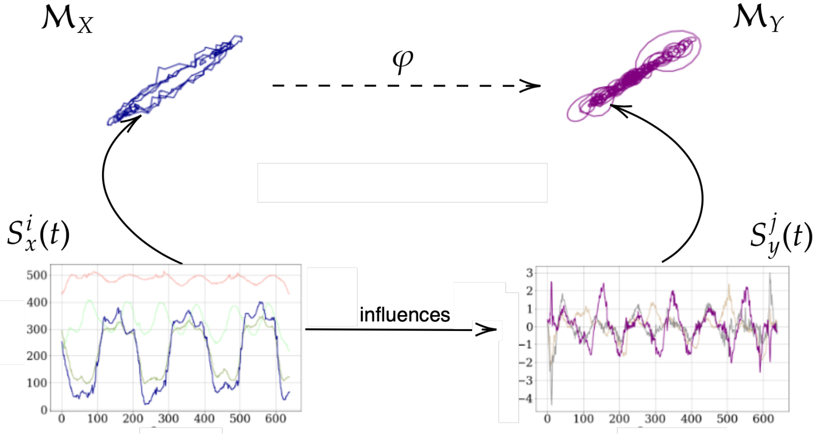
$$\mathbf{x} = \mathbf{P}\mathbf{t} + \mathbf{e}_\mathbf{x}$$

$$\mathbf{y} = \mathbf{Q}\mathbf{u} + \mathbf{e}_\mathbf{y}$$

$$\text{cov}(\mathbf{t}, \mathbf{u}) \to \max_{\mathbf{P}, \mathbf{Q}}$$

Isachenko R.V., Strijov V.V. Quadratic programming feature selection for multicorrelated signal decoding with partial least squares // Expert Systems with Applications. Volume 207, 30 November 2022.

Time series and phase space[3]

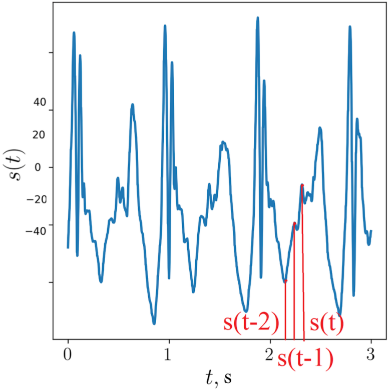


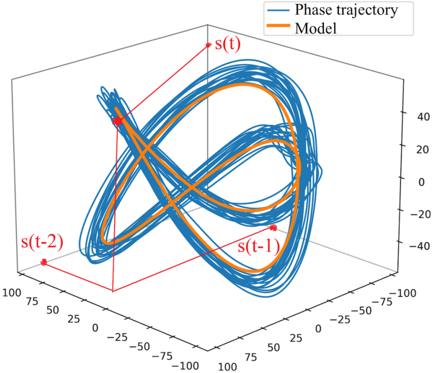
Transform from time domain to frequency domain is essential trick.

[3] Thanks to Ed. Vladimirov

Isachenko R.V., Strijov V.V. Quadratic Programming Optimization with Feature Selection for Non-linear Models // Lobachevskii Journal of Mathematics, 2018, 39(9) : 1179-1187.

Phase trajectory of the accelerometer time series





dim(s) ≈ 1000

dim(x) = 4

Before the next week

Check what the heck is

1. Covariance and contravariance of vectors
2. Covector, one-form
3. Low-rank approximation
4. PCA, SVD

For fun

What is the difference between
kernel function and covariance function (and kernel of a linear map)

See also

ICLR 2021 Keynote - "Geometric Deep Learning: The Erlangen Programme of ML" - M. Bronstein
https://www.youtube.com/watch?v=w6Pw4MOzMuo

Titles:

Geometric deep learning
Functional data analysis
Physics-informed machine learning

Michael Bronstein made this necessary, important, but obvious topic fashionable.

MIPT is intrinsically ready to fulfill this topic.

Testbench for decoding models

August 16, 2022

Test examples. The most common BCI decoding problem is the Neurotycho experiment "Chimpanzee catches banana" to forecast limb positions by ECoG. The second is the problem of labour moitoring, human motion recognition "Smoking workers", to forecast hand acceleration by recorded video. The third one is the Open iEEG-fMRI data "Participants watching film". This set connects several data sources: videofilm, audiovisual labels, iEEG, fMRI. This three decoding problems are unified in table 1. To start with, the "Basic" data set serves to test a trivial model, which forecasts IMU of one hand motion by IMU of the other one.

Table 1: Three decoding problem for BCI data. All data are the time series with synchronised timeline. Some time series have spatial information.

Project	Source	Target	Suppl.	Field source	Field target
Chimpanzee catcing banana	ECoG	xyz		2d cortex layout	3d hand motion
Smoking workers	video	acc (xyz)	pose	2d image space	3d hand motion
Participants watching film	iEEG	fMRI	video, audio, labels	2d cortex layout	3d brain space
ditto	video, audio, labels	iEEG, fMRI		2d image space	2d cortex layout, 3d brain space
Basic	acc(xyz)	acc'(xyz)		3d motion space	3d motion space

Forecasting model. The forecasting model maps the source time series to target time series minimising the forecasting error. Since dimensionality of the source series is supposed to exceed dimensionality of the target, the dimensionality reduction models are needed. This model consists of three parts. The first one is the source reconstruction model. It reconstructs dependencies in the source space and maps the source space to the low-dimensional latent space. The second one of the target reconstruction model. It maps the target space to the latent space. The third one is the alignment model. It aligns the time series in the latent space. The forecasting model is the superposition, the consequent application of source reconstruction, latent alignment, and target reconstruction models.

Phase spaces and trajectories. The spatial time series are represented as points in the phase space. The timeline forms this series as a phase trajectory in this space. This operation increases dimensionality of the data significantly. There are other ways to form a phase trajectory, like a Fourier transform to a spectrogram or a scalogram. To run machine learning methods, which create the desired forecasting model, define an object of sample set. It is a tensor over a timestamp. An object usually carries its local time history. This locality defines the dimensionality of the object. The object could carry its spatial layout.

Reconstruction model. The reconstruction model reduces dimensionality to map an object to the latent space.

Index reduction It is a part of the reconstruction model. The number of ways in the latent space is given.

Reconstruction error. It estimates quality of either source or target model, so it does not require synchronization of time stamps. This error delivers the initial value of the model parameters.

Alignment model. The alignment model maps reduced source and target object within the same space. By default this model just scales objects from both spaced with some time-independent factor. Time alignment to match sampling rated or to keep motion phase of signals is under consideration (but difficult to realise).

Alignment error. There is no error for a single object, since the original alignment error is covariance of two random variables. This covariance is estimated by either of synchronised pairs from both spaces, or by distance between distributions of unsynchronized objects.

The model superposition. The parameters of the forecasting model joins the parameters of reconstruction and the alignment models. These parameters are fine-tuned by the forecasting error.

Synchronised and unsynchronised time stamps. The target and the source time series usually have different sample rates. Table 2 shows various procedures of learning and testing modes.

Table 2: Objects of the source and target phase trajectories could have synchronised or non synchronised timestamps.

Model	Procedure	Synchronization
Phase trajectory	generation	no
Reconstruction	train and test	no
Alignment	train and test	depends
Forecasting	train	no if possible
ditto	test, validation	yes

The computation experiment schedule. The phase trajectories are generated before the computational experiment starts. These trajectories form the initial data set. It cannot be changed during the computational experiment. First train the source and target reconstruction model. This delivers the initial parameters of these two model. Then estimate the parameters of the alignment model. After these three vectors of parameters are set estimate the error of the forecasting model. Minimize this error optimizing the model parameters. During the optimization various types of the error functions could be used. After the parameters converge, fix them and estimate the test error.

Time-scale synchronization procedure. Denote by $\mathbf{x}_t$ an object at the time t, the index $\mathbf{x}_i$ fixes the synchronised time $i = i(t)$, so it gives the pair $(\mathbf{x}_i, \mathbf{y}_i)$. Denote by $\mathbf{X}$ the set of objects for a given time segment, or a time sample set $\mathbf{X}_{\mathcal{T}} = [\mathbf{x}_t^\mathsf{T}, t \in \mathcal{T}]$.

List of functions. Below both the mathematical and the programming notations have the local visibility for simplicity.

$$t \in \mathcal{T} = \mathcal{T}_{\text{src}} \cup \mathcal{T}_{\text{tgt}} = \sqcup \mathcal{B}, \quad i = i(t)$$

```
Function: timesync
In: time_stamps_src, timestamps_tgt
Out: t_src_reconstruction, t_tgt_reconstruction, t_syn_train, t_syn_test,
     t_syn_validation
```

$$\mathbf{u}_t = \mathbf{f}(\mathbf{x}_t, \mathbf{w}, \boldsymbol{\gamma}), \quad \mathbf{x}'_t = \mathbf{f}'(\mathbf{u}_t, \mathbf{w}', \boldsymbol{\gamma}')$$

```
Function: reconstruct_predict
In: model, x, params, hyparams
Out: u
```

$$\hat{\mathbf{w}}, \hat{\boldsymbol{\gamma}} = \arg\min \mathcal{Q}\big(\mathbf{x}_t, \mathbf{f}'(\mathbf{u}_t, \mathbf{w}'_0, \boldsymbol{\gamma}'_0), \mathbf{w}_0, \boldsymbol{\gamma}_0, t \in \mathcal{T}\big) \mapsto \epsilon$$

```
Function: reconstruct_fit
In: model, X, params, hyparams, model_inv, params_inv, hyparams_inv
Out: fit_err, params, hyparams
```

$$\mathcal{Q}\big(\mathbf{x}_t, \mathbf{x}', \mathbf{w}\big) \mapsto \nabla_{\mathbf{w}}\mathcal{Q}$$

```
Function: reconstruct_err
In: X, X_inv, params
Out: err, grad_par
```

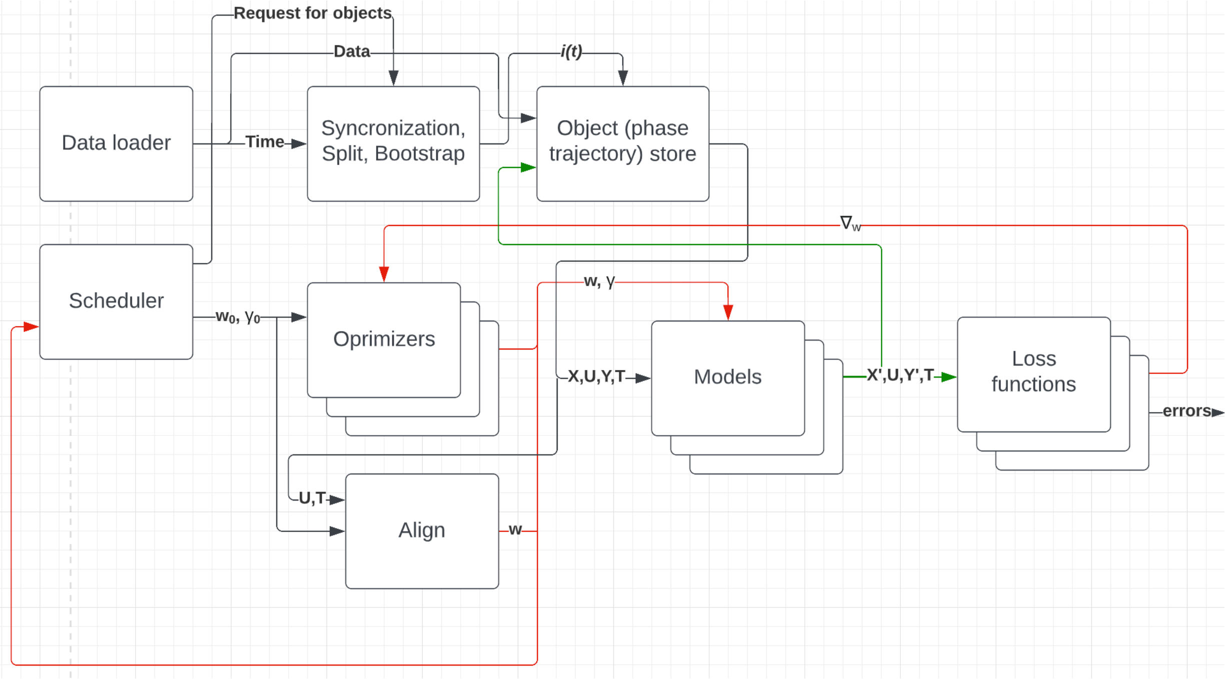
$$\hat{\mathbf{w}} = \mathcal{C}\big(\mathbf{U}, \mathbf{V}, \mathbf{w}_0\big) \mapsto \varsigma$$

```
Function: align_fit
In: U, V, params, hyparams
Out: err, params
```

$$\hat{\mathbf{w}}_{\mathbf{x}}, \hat{\mathbf{w}}_{\mathbf{y}} = \arg\min \mathcal{S}\big(\mathbf{x}_t, \mathbf{f}_{\mathbf{x}}, \mathbf{y}_t, \mathbf{f}'_{\mathbf{y}}, \mathbf{f}_{\mathbf{u}}\{\mathbf{w}_0\}, \{\gamma_0\}, t \in \mathcal{T}\big) \mapsto \varepsilon$$

```
Function: decode_fit
In: X, Y, {params, hyparams}
Out: err, {params}
```

IDEF0 of the testbench. This scheme assumes interchangeability of models and model parameters optimisation algorithms. It requires an automatic differentiation library.


Request for objects
Data
i(t)
Data loader
Time
Syncronization, Split, Bootstrap
Object (phase trajectory) store
∇w
Scheduler
w₀, γ₀
w, γ
Oprimizers
Models
X,U,Y,T
X',U,Y',T
Loss functions
errors
U,T
Align
w