

# Feature generation for multiscale time series forecasting

A. Motrenko, R. Neychev, R. Isachenko, M. Popova, V. Strijov

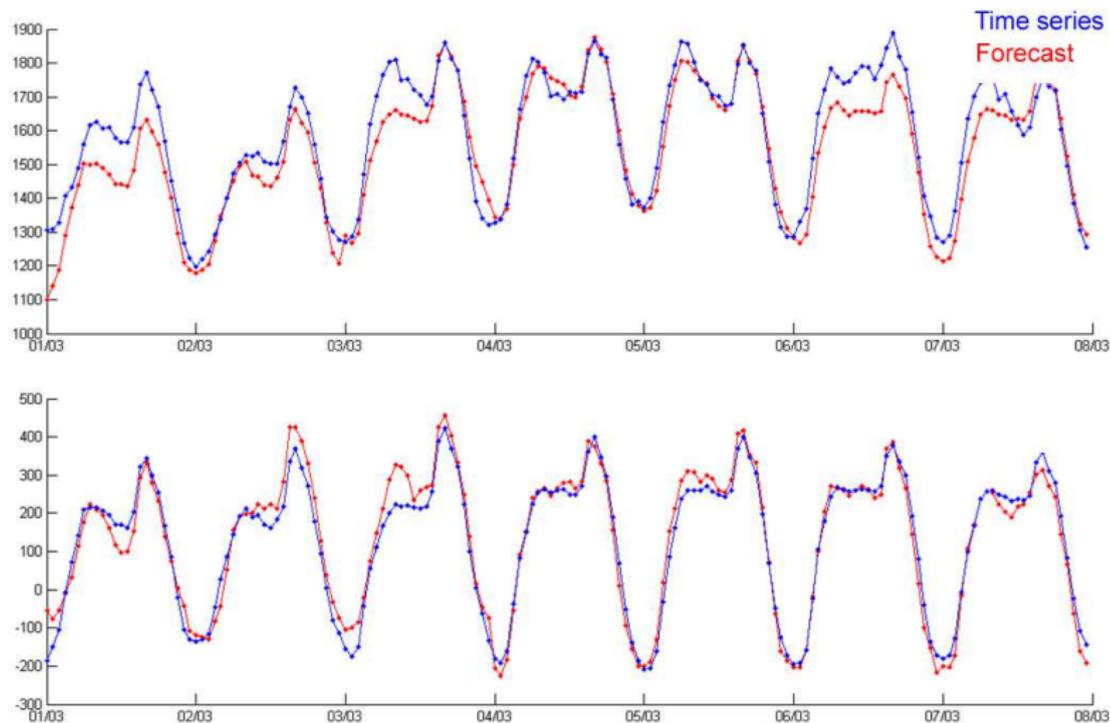
Moscow Institute of Physics and Technology

EURO 2016

- ▶ Problem statement
- ▶ Testing procedure
- ▶ Feature generation
- ▶ Mixture models
- ▶ Computational experiments

# Problem statement

# Energy consumption one-week forecast for each hour



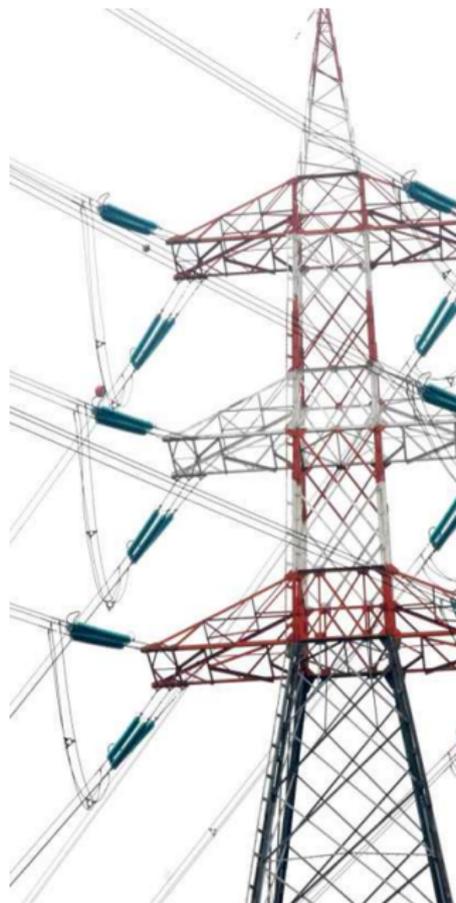
# The periodic components of the multivariate time series

## The time series:

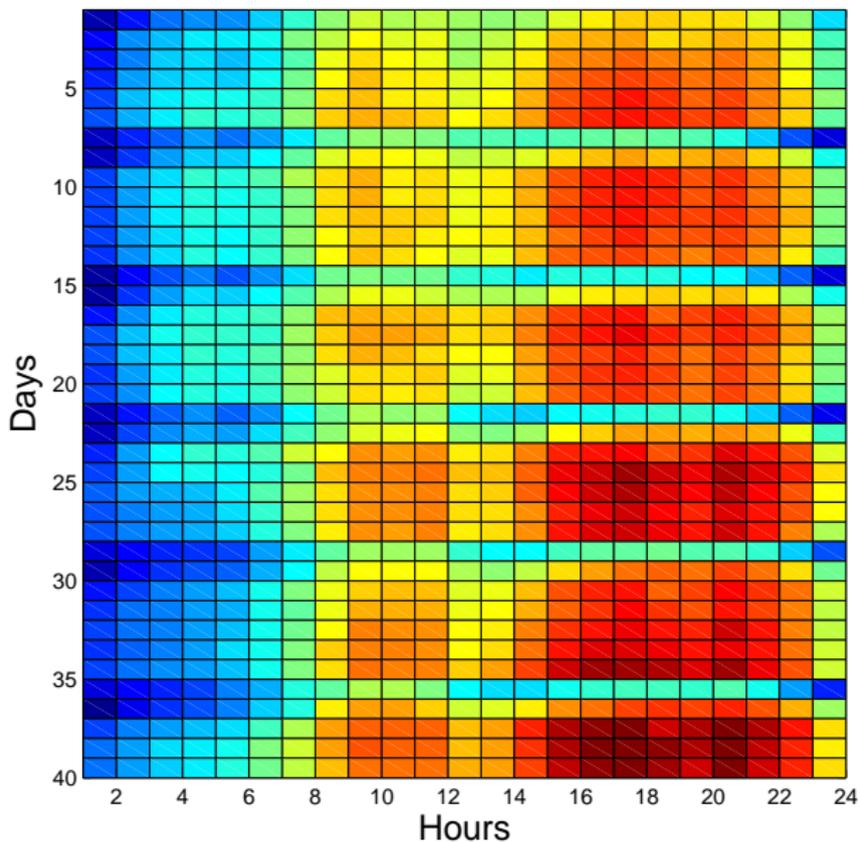
- ▶ energy price,
- ▶ consumption,
- ▶ daytime,
- ▶ temperature,
- ▶ humidity,
- ▶ wind force,
- ▶ holiday schedule.

## Periods:

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



# The autoregressive matrix, five week-ends



# The autoregressive matrix and the linear model

$$\mathbf{X}^*_{(m+1) \times (n+1)} = \left( \begin{array}{c|ccc} S_T & S_{T-1} & \dots & S_{T-\kappa+1} \\ \hline S_{(m-1)\kappa} & S_{(m-1)\kappa-1} & \dots & S_{(m-2)\kappa+1} \\ \dots & \dots & \dots & \dots \\ S_{n\kappa} & S_{n\kappa-1} & \dots & S_{n(\kappa-1)+1} \\ \dots & \dots & \dots & \dots \\ S_\kappa & S_{\kappa-1} & \dots & S_1 \end{array} \right) .$$

In a nutshell,

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

In terms of linear regression:

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

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

# Model generation

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

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

$$\left( \begin{array}{ccc|ccc} g_1 \circ S_{T-1} & \dots & g_r \circ S_{T-1} & \dots & g_1 \circ S_{T-\kappa+1} & \dots & g_r \circ S_{T-\kappa+1} \\ g_1 \circ S_{(m-1)\kappa-1} & \dots & g_r \circ S_{(m-1)\kappa-1} & \dots & g_1 \circ S_{(m-2)\kappa+1} & \dots & g_r \circ S_{(m-2)\kappa+1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ g_1 \circ S_{n\kappa-1} & \dots & g_r \circ S_{n\kappa-1} & \dots & g_1 \circ S_{n(\kappa-1)+1} & \dots & g_r \circ S_{n(\kappa-1)+1} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ g_1 \circ S_{\kappa-1} & \dots & g_r \circ S_{\kappa-1} & \dots & g_1 \circ S_1 & \dots & g_r \circ S_1 \end{array} \right)$$

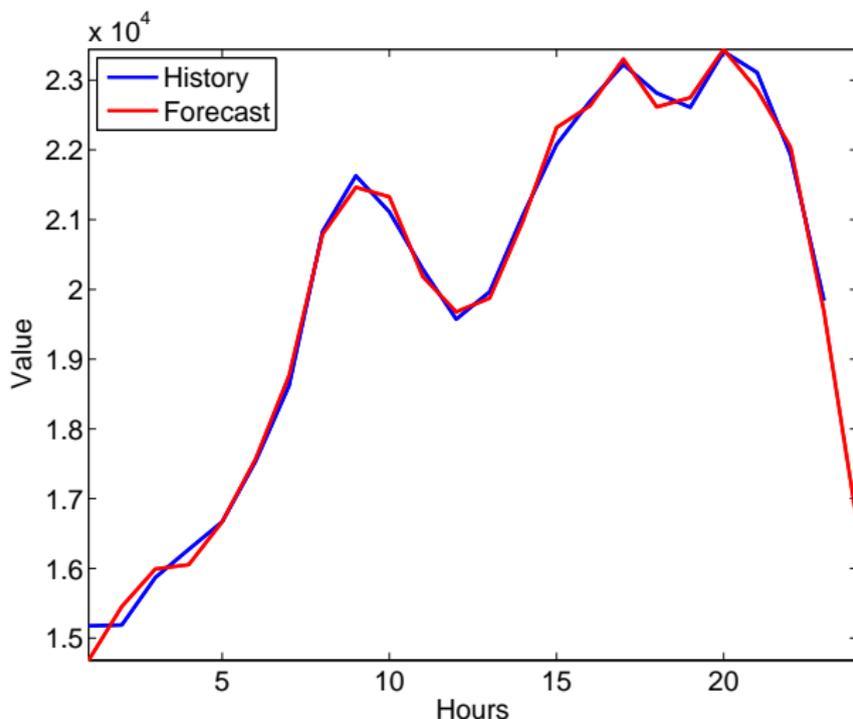
Kolmogorov-Gabor polynomial as a variant for model generation

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

where the coefficients

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

# 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  $\mathbf{X}^*$  is  
 $(m+1) \times (n+1)$

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

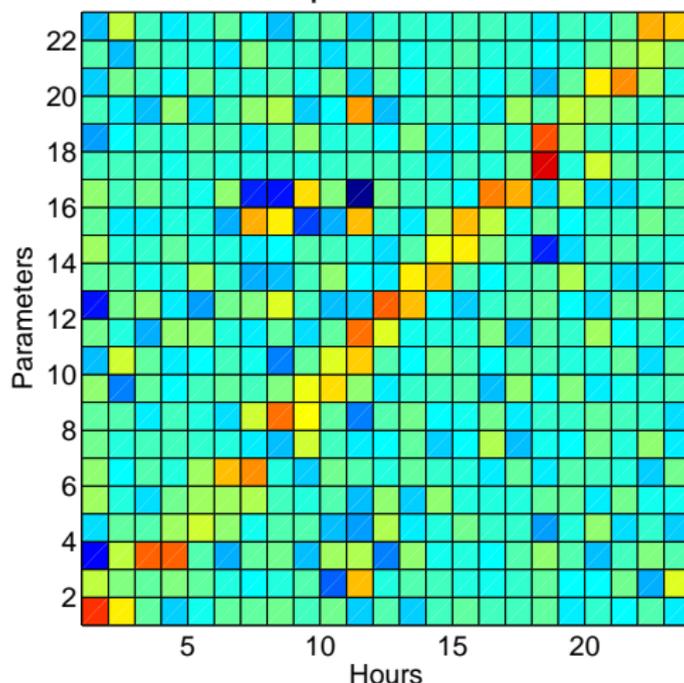
- ▶ for 6 time series the matrix  $\mathbf{X}$  is  $156 \times 1008$ ,
- ▶ for 4 primitive functions it is  $156 \times 4032$ ,

$$m \ll n.$$

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

# How many parameters must be used to forecast?

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



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

# Ways to model selection: brief list of algorithms

## Exhaustive search and modifications

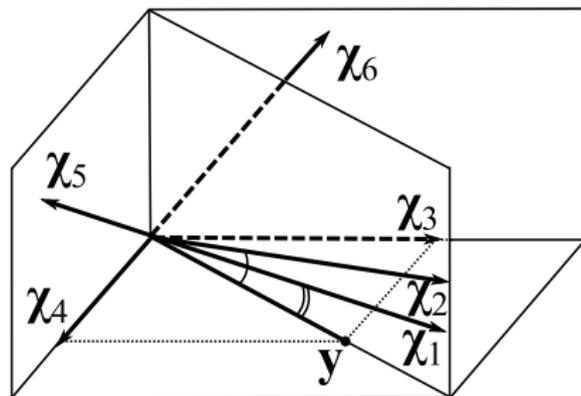
1. Exhaustive search of  $2^P$  models
2. Genetic algorithms
3. Add/Del (append/delete a feature),  $P(P - 1)/2$  models
4. Add-del or stepwise regression,  $\sim P^2$  models

## Parameter space analysis

1. Least angle regression, Lasso, Stagewise, Elastic net
2. Optimal brain damage/surgery

# Selection of a stable set of features of restricted size

The sample contains multicollinear  $\chi_1, \chi_2$  and noisy  $\chi_5, \chi_6$  features, columns of the design matrix  $\mathbf{X}$ . We want to select two features from six.

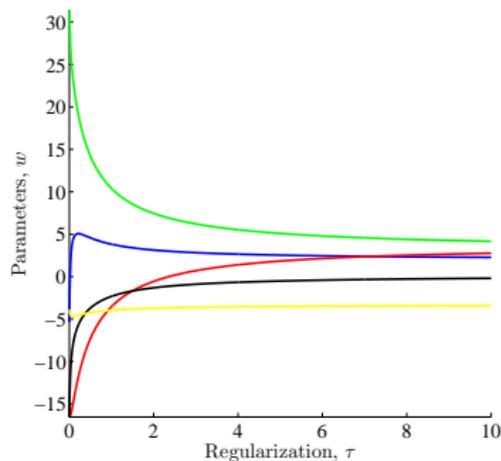


## Stability and accuracy for a fixed complexity

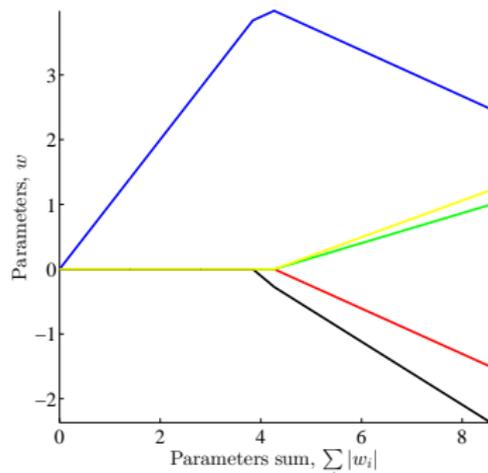
The solution:  $\chi_3, \chi_4$  is an orthogonal set of features minimizing the error function.

# Model parameter values with regularization

Vector-function  $\mathbf{f} = \mathbf{f}(\mathbf{w}, \mathbf{X}) = [f(\mathbf{w}, \mathbf{x}_1), \dots, f(\mathbf{w}, \mathbf{x}_m)]^T \in \mathbb{Y}^m$ .



$$S(\mathbf{w}) = \|\mathbf{f}(\mathbf{w}, \mathbf{X}) - \mathbf{y}\|^2 + \gamma^2 \|\mathbf{w}\|^2$$



$$S(\mathbf{w}) = \|\mathbf{f}(\mathbf{w}, \mathbf{X}) - \mathbf{y}\|^2, \\ T(\mathbf{w}) \leq \tau$$

# Multiscale data

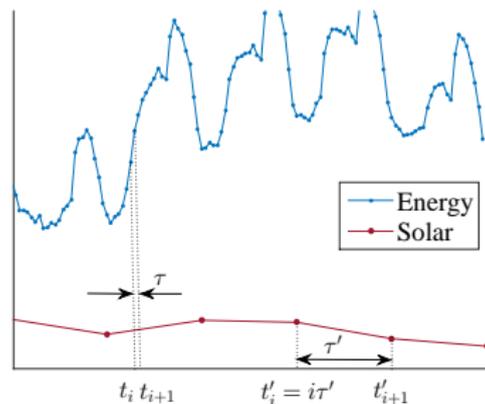
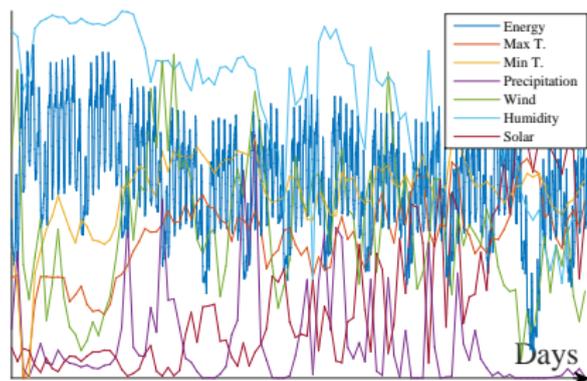
Consider a large set of time series  $\mathcal{D} = \{\mathbf{s}^{(q)} \mid q = 1 \dots, Q\}$ .

Each real-valued time series  $\mathbf{s}$

$$\mathbf{s} = [s_1, \dots, s_i, \dots, s_T], \quad s_i = s(t_i), \quad 0 \leq t_i \leq t_{\max}$$

is a sequence of observations of some real-valued signal  $s(t)$ .

Each time series  $\mathbf{s}^{(q)}$  has its own sampling rate  $\tau^{(q)}$ .



# Resampling time series

# Resampling time series

Suppose that the observations  $s_i = s(t_i)$  of the signal  $s(t)$  are sampled unevenly:

$$G = \{t_1, \dots, t_T\}, \quad t_i \neq i \cdot \frac{t_T - t_1}{T - 1}$$

To obtain evenly spaced observations:

- 1) select a new sampling rate  $\tau_{rs}$ ,
- 2) form the new grid

$$G_s = \{t_1, \dots, T_{rs}\}, \quad t_i = t_1 + (i - 1) \cdot \tau_{rs}$$

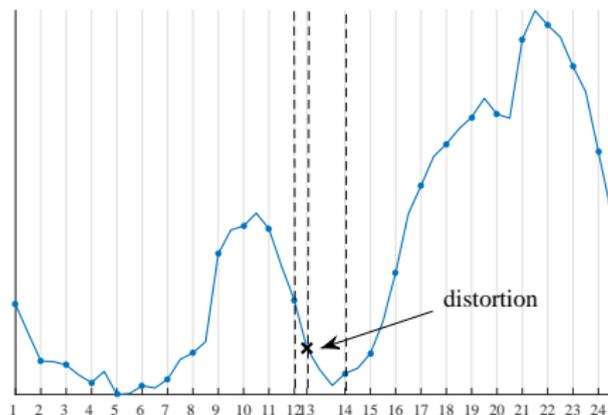
- 3) and approximate unobserved evenly-spaced values  $\hat{s}_i = s(t_i)$ ,  $t_i \in G_s$  using the sampled observations  $s_j = s(t_j)$ ,  $t_j \in G$ .

# Resampling: special case

1. The initial sampling rate is approximately even, but distortions are possible:

$$t_i = i \cdot \tau + \delta_i, |\delta_i| < \frac{\tau}{2}.$$

In this case the number  $T_s$  of resampled observations equals the initial number of observations  $T$ .

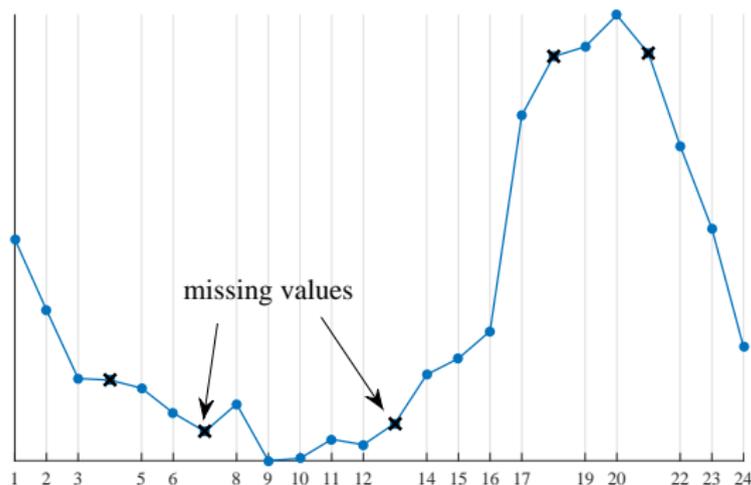


# Resampling: special case

2. The sampling rate is even, but some values are missing:

$$|t_{i+1} - t_i| = n\tau, n \in \mathbb{N}.$$

Here  $\tau_s = \tau$  and missing values are the only ones that one needs to approximate.

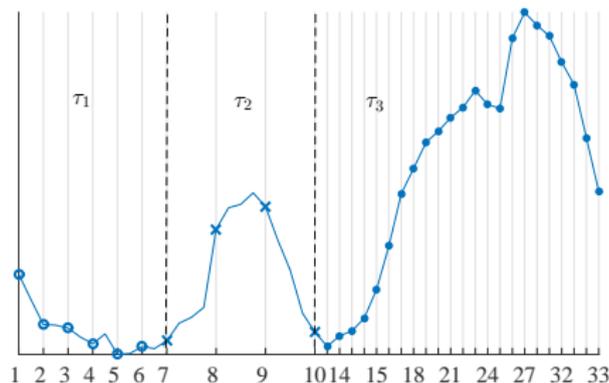


## Resampling: special case

3. Time series  $\mathbf{s}$  comprises a finite number of intervals  $\mathbf{s}_k$ , each sampled from  $s(t)$  at fixed sampling rate:

$$\mathbf{s} = \left[ s(\tau_1), \dots, s(T_1\tau_1), s(T_1\tau_1 + \tau_2), \dots, s\left(\sum_k T_k\tau_k\right) \right],$$

where  $\sum_k T_k = T$ . Here we select the maximum sampling rate  $f_s = \max_k \frac{1}{T_k}$  and upsample the rest time series, using piecewise constant approximation.



# Resampling details

Suppose that the signal  $s(t)$  is bandlimited with frequency  $f_b$ .

**Nyquist–Shannon sampling condition:** it is sufficient to sample the signal  $s(t)$  with frequency

$$\frac{1}{\tau_{rs}} = f_s > 2f_b$$

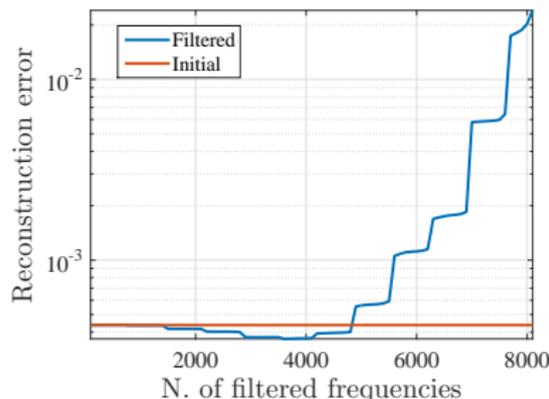
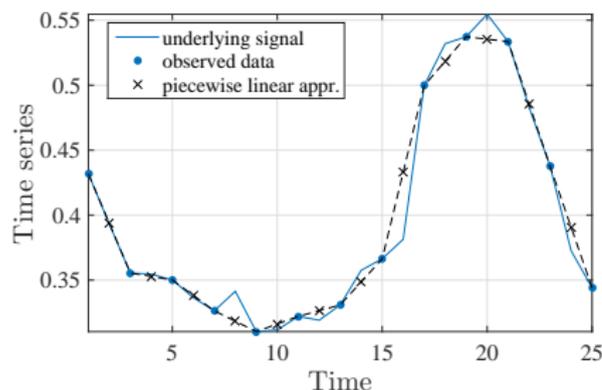
to be able to fully reconstruct the signal from its discretely sampled observations  $s(t_j) = s(i\tau_s)$ .

Discrete signals are never bandlimited  $\Rightarrow$  the time series have to be low-pass filtered to satisfy the Nyquist condition.

# Upsampling procedure

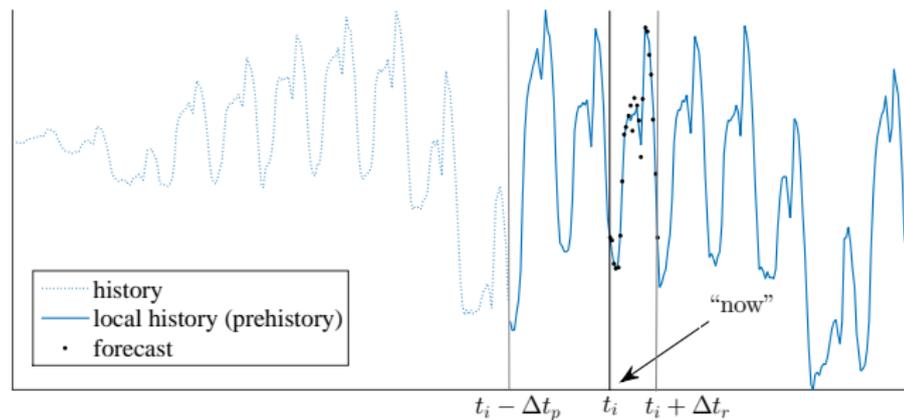
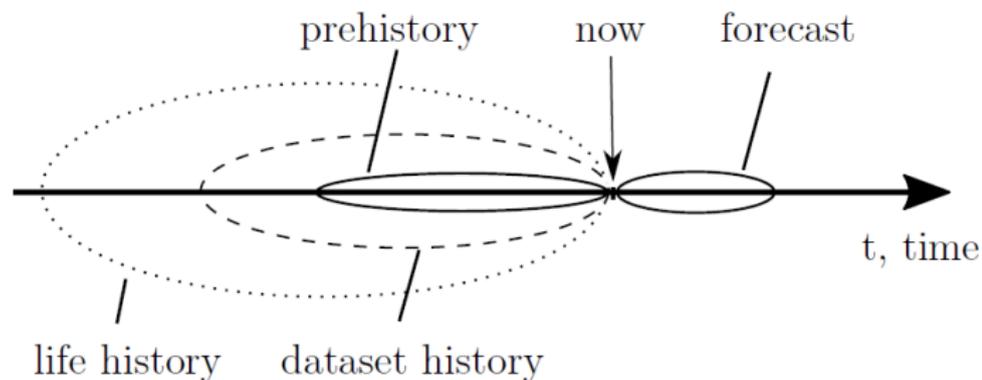
Let  $G_s$  be the desired grid,  $G \subseteq G_s$ . To obtain  $s(G_s)$ :

1. Approximate  $s(G_s \setminus G)$  using piecewise linear approximation.
2. Find  $s$ 's FFT coefficients  $a_j$ ,  $b_j$  for  $j = 1, \dots, 2^{\lfloor \log_2 T \rfloor}$ .
3. Set  $a_j = 0$ ,  $b_j = 0$ , for  $j > 2^{\lfloor \log_2 T \rfloor - 1}$ .
4. Reconstruct the time series, using inverse FFT.

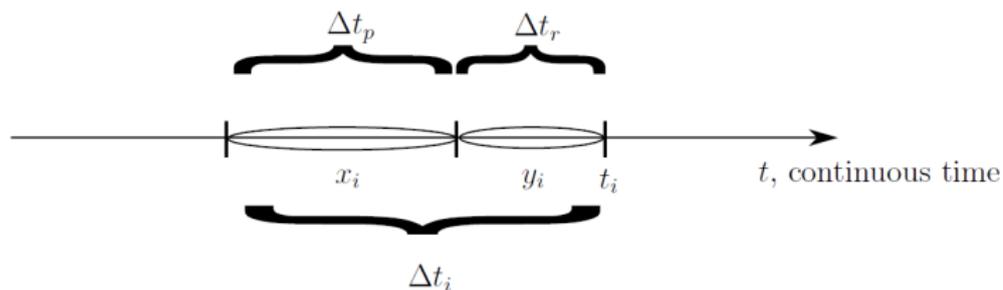


# Testing procedure

# Time series forecasting

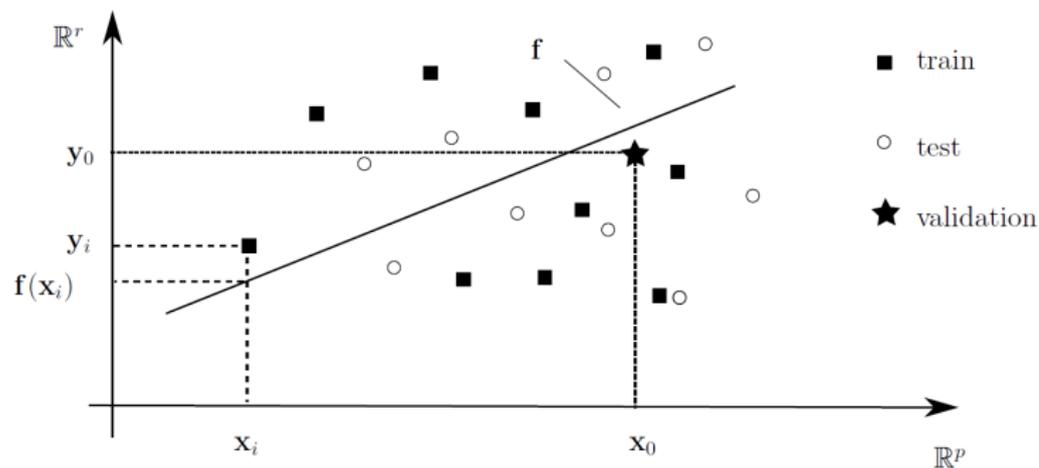


# Design matrix



$$[\mathbf{x}_i | \mathbf{y}_i] = \underbrace{s(t_i - \Delta t_r - \Delta t_p), \dots, s(t_i - \Delta t_r)}_{\mathbf{x}_i}, \dots, \underbrace{s(t_i)}_{\mathbf{y}_i}.$$

# Design matrix



$$\mathbf{X}^* = \left[ \begin{array}{c|c} \mathbf{x} & \mathbf{y} \\ \hline \mathbf{X} & \mathbf{Y} \\ \hline \mathbf{x}_1^{(1)} & \dots & \mathbf{x}_1^{(Q)} & \mathbf{y}_1^{(1)} & \dots & \mathbf{y}_1^{(Q)} \end{array} \right] = \left[ \begin{array}{ccc|ccc} \mathbf{x}^{(1)} & \dots & \mathbf{x}^{(Q)} & \mathbf{y}^{(1)} & \dots & \mathbf{y}^{(Q)} \\ \hline \mathbf{x}_m^{(1)} & \dots & \mathbf{x}_m^{(Q)} & \mathbf{y}_m^{(1)} & \dots & \mathbf{y}_m^{(Q)} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_1^{(1)} & \dots & \mathbf{x}_1^{(Q)} & \mathbf{y}_1^{(1)} & \dots & \mathbf{y}_1^{(Q)} \end{array} \right].$$

# Regression problem

Now we are able to state the regression problem as follows:

$$\hat{\mathbf{y}} = f(\mathbf{x}, \hat{\mathbf{w}}), \quad \hat{\mathbf{w}} = \arg \min_{\hat{\mathbf{w}}} S(\mathbf{w} | \mathbf{f}(\mathbf{w}, \mathbf{x}), \mathbf{y}). \quad (1)$$

Here the error function  $S(\mathbf{w} | \mathbf{f}(\mathbf{w}, \mathbf{x}), \mathbf{y})$  averages forecasting errors of  $[\mathbf{x}_i | \mathbf{y}_i]$  over all segments  $i = 1, \dots, m$  in the test set:

$$S(\mathbf{w} | \mathbf{f}(\mathbf{w}, \mathbf{x}), \mathbf{y}) = \frac{r}{m} \sum_{i=1}^m l(\mathbf{y}_i, f(\mathbf{x}_i, \mathbf{w})).$$

Let  $\boldsymbol{\varepsilon}$  denote residual vector

$$\boldsymbol{\varepsilon} = [\varepsilon_1, \dots, \varepsilon_r] = \mathbf{y} - \hat{\mathbf{y}}$$

for the forecast  $\hat{\mathbf{y}} = \mathbf{f}(\mathbf{w}, \mathbf{x})$  of  $\mathbf{y}$ .

# Types of forecasting errors

- ▶ scale-dependent metrics: mean absolute error

$$MAE = \frac{1}{r} \sum_{j=1}^r |\varepsilon_j|,$$

- ▶ percentage-error metrics: (symmetric) mean absolute percent error

$$MAPE = \frac{1}{r} \sum_{j=1}^r \frac{|\varepsilon_j|}{|y_j|}, \quad sMAPE = \frac{1}{r} \sum_{j=1}^r \frac{2|\varepsilon_j|}{|\hat{y}_j + y_j|},$$

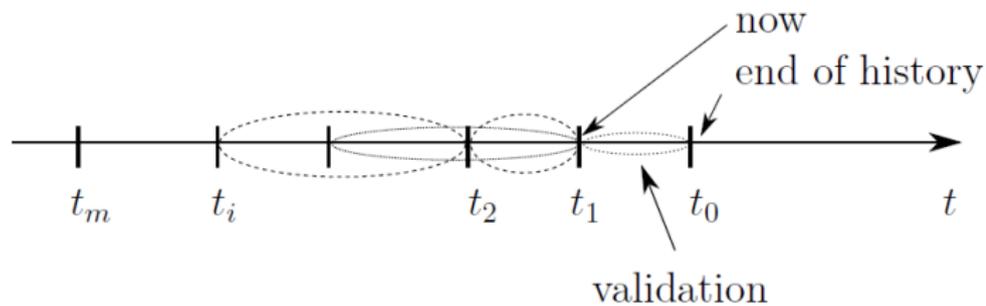
- ▶ relative-error metrics (to residues  $\varepsilon^*$  of a benchmark method):

$$MRAE = \frac{1}{r} \sum_{j=1}^r \frac{|\varepsilon_j|}{\varepsilon_j^*},$$

- ▶ and scale-free error metrics:

$$MASE = \frac{n-1}{r} \frac{\sum_{i=1}^r |\varepsilon_i|}{\sum_{j=2}^n |x_j - x_{j-1}|}.$$

# Rolling validation



# Rolling validation

- 1) construct the validation vector  $\mathbf{x}_{\text{val},k}^*$  for time series of the length  $\Delta t_r$  as the first row of the design matrix  $\mathbf{Z}$ ,
- 2) construct the rest rows of the design matrix  $\mathbf{Z}$  for the time after  $t_k$  and present it as

$$\mathbf{Z} = \left[ \begin{array}{c|c} \dots & \dots \\ \hline \mathbf{x}_{\text{val},k} & \mathbf{y}_{\text{val},k} \\ \mathbf{1} \times n & \mathbf{1} \times r \\ \hline \mathbf{X}_{\text{train},k} & \mathbf{Y}_{\text{train},k} \\ m_{\text{min}} \times n & m_{\text{min}} \times r \\ \hline \dots & \dots \end{array} \right], \uparrow_k$$

- 3) optimize model parameters  $\mathbf{w}$  using  $\mathbf{X}_{\text{train},k}$ ,  $\mathbf{Y}_{\text{train},k}$ ,
- 4) compute residues  $\varepsilon_k = \mathbf{y}_{\text{val},k} - \mathbf{f}(\mathbf{x}_{\text{val},k}, \mathbf{w})$  and MAPE,
- 5) increment  $k$  and repeat.

# Feature generation

# Generating extra features

To augment feature description, consider the following types of features:

- 1) the local history of all time series themselves,
- 2) transformations (non-parametric and parametric) of local history,
- 3) parameters of the local models,
- 4) distances to the centroids of local clusters.

# Functional transforms

The procedure of generating new features  $\phi$  requires:

- ▶ the original features  $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_Q\}$ ,
- ▶ the set of primitive functions  $G = \{g(\mathbf{b}, \mathbf{x})\}$ ,

$$g : \mathbf{x} \mapsto \phi;$$

- ▶ the generation rules:  $\mathcal{G} \supset G$ , where the superposition  $g_k \circ g_l \in \mathcal{G}$  w.r.t. numbers and types of the input and output arguments;
- ▶ the simplification rules:  $g_u$  is not in  $\mathcal{G}$ , if there exist a rule

$$r : g_u \mapsto g_v \in \mathcal{G}.$$

The result is

the set of the features  $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_Q, \phi_1, \dots, \phi_N\}$ .

# Examples of nonparametric transformation functions

► Univariate

Formula	Output dimension
$\sqrt{x}$	1
$x\sqrt{x}$	1
$\arctan x$	1
$\ln x$	1
$x \ln x$	1

► Bivariate

Plus	$x_1 + x_2$
Minus	$x_1 - x_2$
Product	$x_1 \cdot x_2$
Division	$\frac{x_1}{x_2}$
	$x_1 \sqrt{x_2}$
	$x_1 \ln x_2$

# Nonparametric transformations: sample statistics

Nonparametric transformations include basic data statistics:

- ▶ Sum or average value of each row  $\mathbf{x}_i$ ,  $i = 1, \dots, m$ :

$$\phi_i = \sum_{j=1}^n x_{ij}, \text{ or } \phi'_i = \frac{1}{n} \sum_{j=1}^n x_{ij}.$$

- ▶ Min and max values:  $\phi_i = \min_j x_{ij}$ ,  $\phi'_i = \max_j x_{ij}$ .
- ▶ Standard deviation:

$$\phi_i = \frac{1}{n-1} \sqrt{\sum_{j=1}^n (x_{ij} - \text{mean}(\mathbf{x}_i))^2}.$$

- ▶ Data quantiles:  $\phi_i = [X_1, \dots, X_K]$ , where

$$\sum_{j=1}^n [X_{k-1} < x_{ij} \leq X_k] = \frac{1}{K}, \text{ for } k = 1, \dots, K.$$

# Nonparametric transformations: Haar's transform

Applying Haar's transform produces multiscale representations of the same data.

Assume that  $n = 2^K$  and init  $\phi_{i,j}^{(0)} = \phi'_{i,j}{}^{(0)} = x_{ij}$  for  $j = 1, \dots, n$ .

To obtain coarse-graining and fine-graining of the input feature vector  $\mathbf{x}_i$ , for  $k = 1, \dots, K$  repeat:

- ▶ data averaging step

$$\phi_{i,j}^{(k)} = \frac{\phi_{i,2j-1}^{(k-1)} + \phi_{i,2j}^{(k-1)}}{2}, \quad j = 1, \dots, \frac{n}{2^k},$$

- ▶ and data differencing step

$$\phi'_{i,j}{}^{(k)} = \frac{\phi_{i,2j}^{(k-1)} - \phi_{i,2j-1}^{(k-1)}}{2}, \quad j = 1, \dots, \frac{n}{2^k}.$$

The resulting multiscale feature vectors are  $\phi_i = [\phi_i^{(1)}, \dots, \phi_i^{(K)}]$  and  $\phi'_i = [\phi'_i{}^{(1)}, \dots, \phi'_i{}^{(K)}]$ .

# Parametric transformations

Optimization of the transformation function parameters  $\mathbf{b}$  is iterative:

1. Fix the vector  $\hat{\mathbf{b}}$ , collected over all the primitive functions  $\{g\}$ , which generate features  $\phi$ :

$$\hat{\mathbf{w}} = \arg \min S(\mathbf{w} | \mathbf{f}(\mathbf{w}, \mathbf{x}), \mathbf{y}), \quad \text{where } \phi(\hat{\mathbf{b}}, \mathbf{s}) \subseteq \mathbf{x}.$$

2. Optimize transformation parameters  $\hat{\mathbf{b}}$  given model parameters  $\hat{\mathbf{w}}$

$$\hat{\mathbf{b}} = \arg \min S(\mathbf{b} | \mathbf{f}(\hat{\mathbf{w}}, \mathbf{x}), \mathbf{y}).$$

Repeat these steps until vectors  $\hat{\mathbf{w}}$ ,  $\hat{\mathbf{b}}$  converge.

# Examples of parametric transformation functions

Function name	Formula	Output dim.	Num. of args	Num. of pars
Add constant	$x + w$	1	1	1
Quadratic	$w_2x^2 + w_1x + w_0$	1	1	3
Cubic	$w_3x^3 + w_2x^2 + w_1x + w_0$	1	1	4
Logarithmic sigmoid	$1/(w_0 + \exp(-w_1x))$	1	1	2
Exponent	$\exp x$	1	1	0
Normal	$\frac{1}{w_1\sqrt{2\pi}} \exp\left(\frac{(x-w_2)^2}{2w_1^2}\right)$	1	1	2
Multiply by constant	$x \cdot w$	1	1	1
Monomial	$w_1x^{w_2}$	1	1	2
Weibull-2	$w_1w_2x^{w_2-1} \exp -w_1x^{w_2}$	1	1	2
Weibull-3	$w_1w_2x^{w_2-1} \exp -w_1(x - w_3)^{w_2}$	1	1	3
...	...	...	...	...

# Monotone functions

## ► By grow rate

Function name	Formula	Constraints
Linear	$w_1x + w_0$	
Exponential rate	$\exp(w_1x + w_0)$	$w_1 > 0$
Polynomial rate	$\exp(w_1 \ln x + w_0)$	$w_1 > 1$
Sublinear polynomial rate	$\exp(w_1 \ln x + w_0)$	$0 < w_1 < 1$
Logarithmic rate	$w_1 \ln x + w_0$	$w_1 > 0$
Slow convergence	$w_0 + w_1/x$	$w_1 \neq 0$
Fast convergence	$w_0 + w_1 \cdot \exp(-x)$	$w_1 \neq 0$

## ► Other

Soft ReLu	$\ln(1 + e^x)$	
Sigmoid	$1/(w_0 + \exp(-w_1x))$	$w_1 > 0$
Softmax	$1/(1 + \exp(-x))$	
Hiberbolic tangent	$\tanh(x)$	
softsign	$\frac{ x }{1+ x }$	

# Parameters of the local models

Other options:

- ▶ Parameters of SSA approximation of the time series  $\mathbf{x}^{(q)}$ .
- ▶ Parameters of the FFT of each  $\mathbf{x}^{(q)}$ .
- ▶ Parameters of polynomial/spline approximation of each  $\mathbf{x}^{(q)}$ .

# Parameters of the local models: SSA

For the time series  $\mathbf{s}$  construct the Hankel matrix with a period  $k$  and shift  $p$ , so that for  $\mathbf{s} = [s_1, \dots, s_T]$  the matrix

$$\mathbf{H}^* = \left[ \begin{array}{c|cc} s_T & \dots & s_{T-k+1} \\ \vdots & \ddots & \vdots \\ s_{k+p} & \dots & s_{1+p} \\ s_k & \dots & s_1 \end{array} \right], \text{ where } 1 \geq p \geq k.$$

Reconstruct the regression to the first column of the matrix  $\mathbf{H}^* = [\mathbf{h}, \mathbf{H}]$  and denote its least square parameters as the feature vector

$$\phi(\mathbf{s}) = \arg \min \|\mathbf{h} - \mathbf{H}\phi\|_2^2.$$

For the original feature vector  $\mathbf{x} = [\mathbf{x}^{(1)}, \mathbf{x}^{(Q)}]$  use the parameters  $\phi(\mathbf{x}^{(q)})$ ,  $q = 1, \dots, Q$  as the features.

# Metric features: distances to the centroids of local clusters

Apply kernel trick to the time series.

1. For given local feature vector  $\mathbf{x}_i^{(q)}$ ,  $q = 1, \dots, Q$  compute  $k$ -means centroids  $\mathbf{c}_p^{(m)}$ ,  $p = 1, \dots, P$ .
2. With the selected  $k$ -means distance function  $\rho$  construct the feature vector

$$\phi_i^{(q)} = [\rho(\mathbf{c}_1^{(q)}, \mathbf{x}_i^{(q)}), \dots, \rho(\mathbf{c}_P^{(q)}, \mathbf{x}_i^{(q)})] \in \mathbb{R}_+^P.$$

The procedure may be applied to each  $\mathbf{x}^{(q)}$  or directly to the  $\mathbf{x} = [\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(Q)}]$ , resulting in only  $P$  additional features instead of  $Q \cdot P$

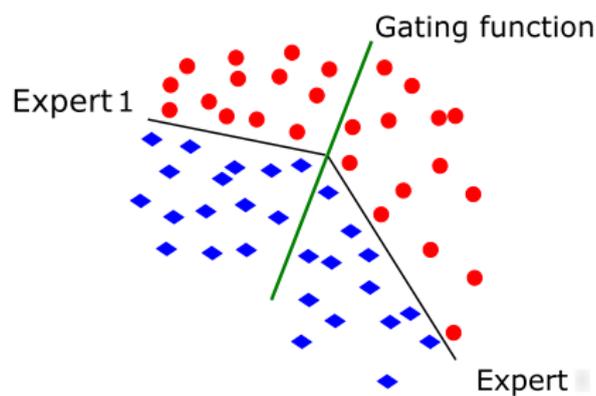
# Mixture of experts

# Gating function

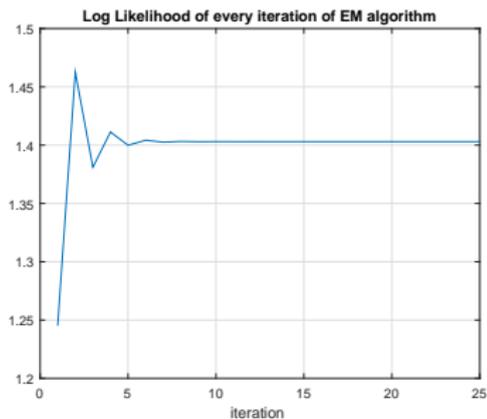
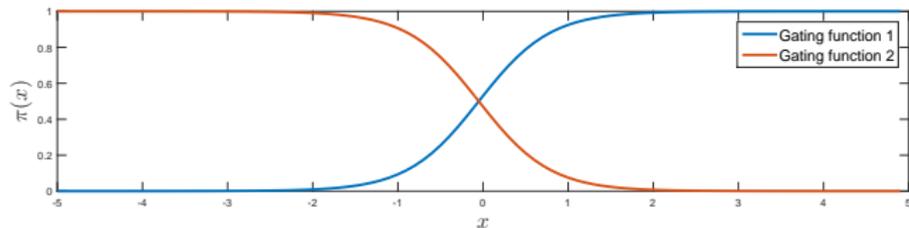
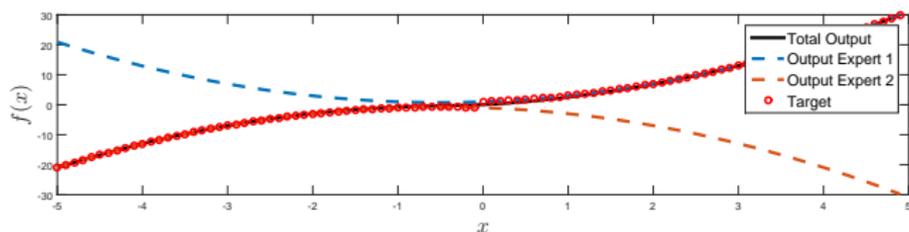
Suppose that each linear model  $f(\mathbf{x}, \mathbf{w}_k)$ ,  $k = 1, \dots, K$  generates a sample  $(\mathbf{x}, y)$  with some probability  $p(k|\mathbf{x})$ .

*Gating function*  $\pi(\mathbf{x}, \mathbf{v})$  maps  $X \mapsto [0, 1]$

$$\pi(\mathbf{x}, \mathbf{v}_k) = \frac{\exp(\mathbf{v}_k^T \mathbf{x})}{\sum_{k'=1}^K \exp(\mathbf{v}_{k'}^T \mathbf{x})}$$



**Mixture of Experts** is a mixture models with weights modeled through gating functions  $\pi_k(\mathbf{x}, \mathbf{v})$ .



Mixture of two experts (2nd degree polynoms), synthetic data.

# Mixture of Experts

Model likelihood  $p(y|\mathbf{x}, \boldsymbol{\theta})$  is given by

$$p(y|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^K p(k|\mathbf{x}, \boldsymbol{\theta})p(y|k, \mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^K \pi(\mathbf{x}, \mathbf{v}_k)\mathcal{N}(y|\mathbf{w}_k^\top \mathbf{x}, \mathbf{B}),$$

where  $\boldsymbol{\theta}$  denotes the concatenated vector of parameters:

$$\boldsymbol{\theta} = [\mathbf{w}_1, \dots, \mathbf{w}_k, \mathbf{v}_1, \dots, \mathbf{v}_k, \beta]^\top,$$

and  $\mathbf{B} = \beta \mathbf{I}_m$  is the covariance matrix for  $\mathbf{y}$ .

# Parameter estimation

To obtain maximum likelihood estimates of  $\hat{\theta}$

$$\hat{\theta} = \arg \max_{\theta} \ln p(\mathbf{y}|\theta),$$

introduce hidden indicator variables  $Z = [\mathbf{z}_1, \dots, \mathbf{z}_m]$ ,  $z_{ik} \in \{0, 1\}$ , such that

$$z_{ik} = 1 \Leftrightarrow y_i \sim \mathcal{N}(\mathbf{w}_k^T \mathbf{x}_i, \beta).$$

Then the loglikelihood function  $p(\mathbf{y}, Z|X, \theta)$  takes the form

$$p(\mathbf{y}|X, Z, \theta) = \sum_{i=1}^m \sum_{k=1}^K z_{ik} (\ln \pi_k + \ln \mathcal{N}(y_i | \mathbf{w}_k^T \mathbf{x}_i, \beta)).$$

**EM-algorithm:** instead of  $p(\mathbf{y}|X, \theta)$  maximize the expected loglikelihood  $E_Z[p(\mathbf{y}, Z|X, \theta)]$  of the observed data.

# EM-algorithm for mixture of experts

**E-step:** Compute expectations, according to Bayes' rule

$$\gamma_{ik}^{r+1} = E(z_{ik}) = p(k|\mathbf{x}_i, \boldsymbol{\theta}^r) = \frac{\pi_k(\mathbf{x}_i, \mathbf{v}_k) \mathcal{N}(\mathbf{y}_i | \mathbf{x}_i^T \mathbf{w}_k^r, \beta^r)}{\sum_{k'=1}^K \pi_{k'}(\mathbf{x}_i, \mathbf{v}_{k'}) \mathcal{N}(\mathbf{y}_i | \mathbf{x}_i^T \mathbf{w}_{k'}^r, \beta^r)}.$$

**M-step:** Recompute parameter estimates:

$$\mathbf{v}_k^{r+1} = \arg \max_{\mathbf{v}} \sum_{i=1}^m \gamma_{ik}^{r+1} \ln \pi_k(\mathbf{x}_i, \mathbf{v}),$$

$$\mathbf{w}_k^{r+1} = \arg \max_{\mathbf{w}_k} \left[ - \sum_{i=1}^m \gamma_{ik}^{r+1} (y_i - \mathbf{w}_k^T \mathbf{x}_i)^2 \right],$$

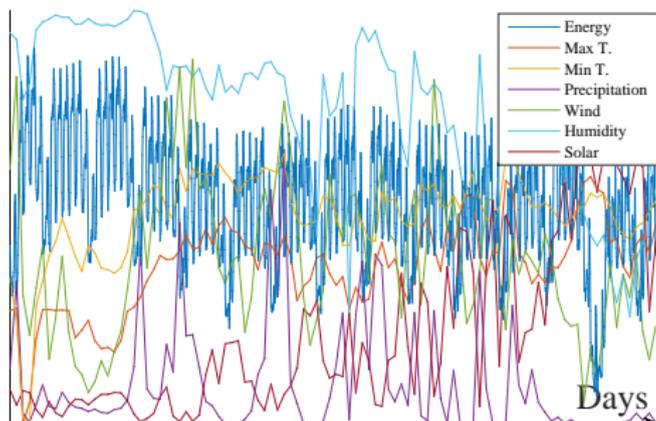
$$\beta_k^{r+1} = \arg \max_{\beta} \left[ n \ln \beta - \frac{1}{\beta} (y_i - \mathbf{x}_i^T \mathbf{w}_k^{r+1})^2 \right].$$

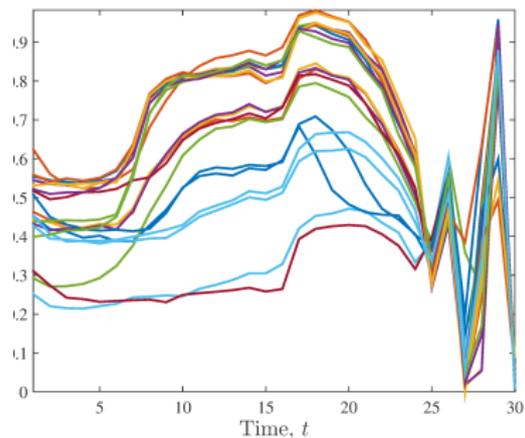
# Computational experiments

- ▶ Demonstrate that the proposed framework yields adequate forecasting quality.
- ▶ Analyze performance of feature generations strategies discussed above.

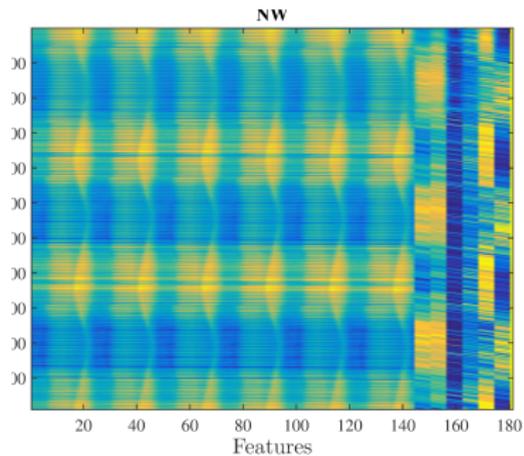
# Data

1. Original Polish electricity load time series, 1999–2004, including:
  - ▶ hourly energy time series (total of 52512 observations),
  - ▶ six daily weather time series from Warsaw (2188 observations): Max Temperature, Min Temperature, Precipitation, Wind, Relative Humidity, Solar.
- 2–5. Data sets with artificial inserted missing values, 1, 3, 5 and 10% missing.
6. Data set with artificially varied sampling rate.





Target variables.



The design matrix.

## Models:

- ▶ Baseline method:  $\hat{s}_i = s_{i-1}$ .
- ▶ Multivariate linear regression (MLR) with  $l_2$ -regularization. Regularization coefficient: 2
- ▶ SVR with multiple output. Kernel type: RBF,  $p_1: 2$ ,  $p_2: 0$ ,  $\gamma: 0.5$ ,  $\lambda: 4$ .
- ▶ Feed-forward ANN with single hidden layer, size: 25
- ▶ Random forest (RF). Number of trees: 25 , number of variables for each decision split: 48.

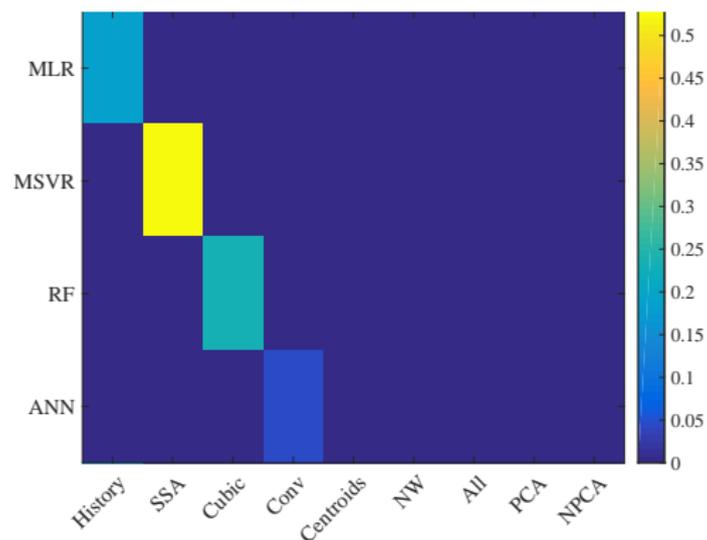
## Feature combinations:

- ▶ History: the standard regression-based forecast with no additional features.
- ▶ SSA, Cubic, Conv, Centroids, NW: history + a particular feature.
- ▶ All: all of the above, with no feature selection.
- ▶ PCA and NPCA: all generation strategies with feature selection.

# Forecasting errors, SMAPE

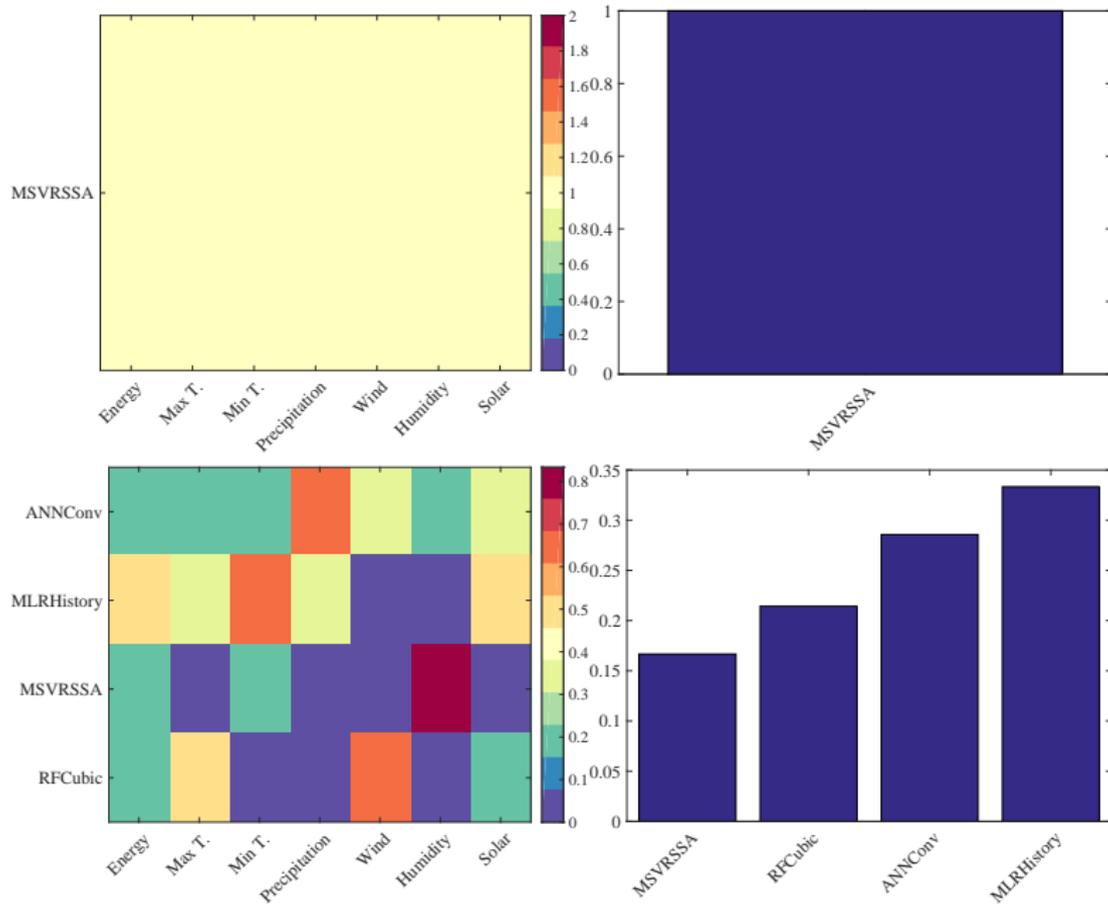
Data	Energy	Max T.	Min T.	Precip.	Wind	Humid.	Solar
Test							
orig	0.111	0.127	0.111	1.222	0.396	0.201	0.495
0.01	0.230	0.185	0.129	1.028	0.397	0.254	0.577
0.03	0.231	0.191	0.137	1.026	0.396	0.253	0.591
0.05	0.230	0.200	0.141	1.017	0.390	0.250	0.592
0.1	0.247	0.198	0.151	1.192	0.381	0.225	0.562
varying	0.124	0.139	0.102	1.232	0.395	0.219	0.489
Train							
orig	0.031	0.073	0.057	0.848	0.111	0.051	0.267
0.01	0.034	0.055	0.040	0.595	0.111	0.055	0.253
0.03	0.034	0.057	0.042	0.595	0.110	0.055	0.249
0.05	0.034	0.060	0.043	0.592	0.109	0.054	0.246
0.1	0.031	0.081	0.063	0.743	0.102	0.051	0.272
varying	0.027	0.057	0.044	0.888	0.112	0.055	0.272

# Feature analysis

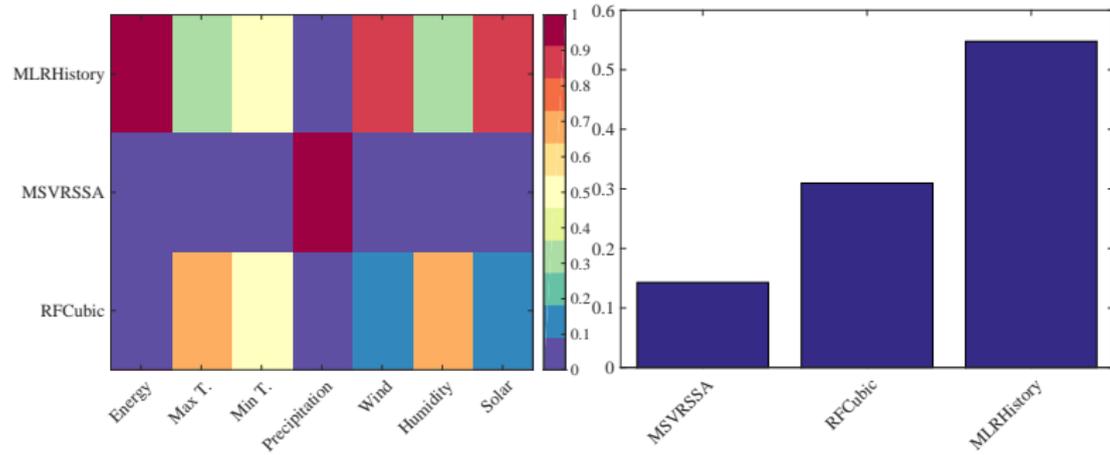
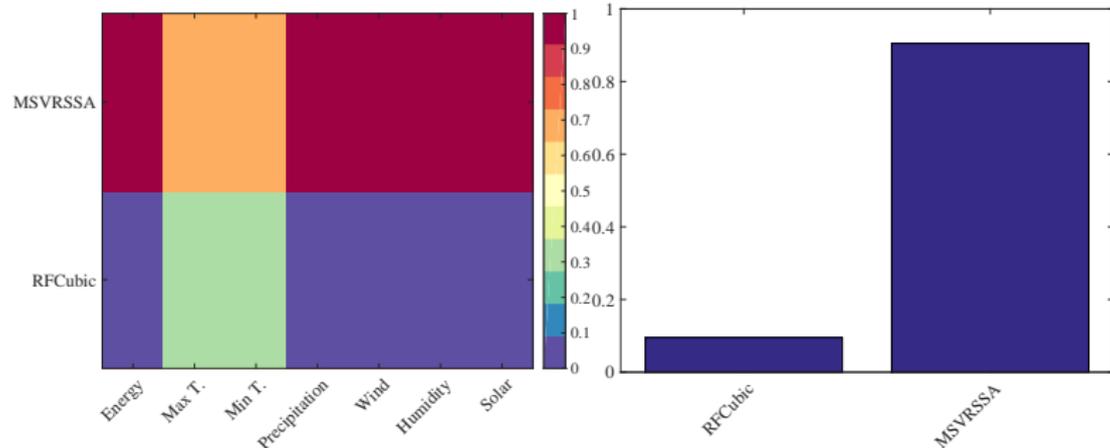


Ratio of times each combination of model and feature performed best for at least one of the time series (7) or error functions (6), all (6) data sets ( $6 \times 7 \times 6 = 252$  cases).

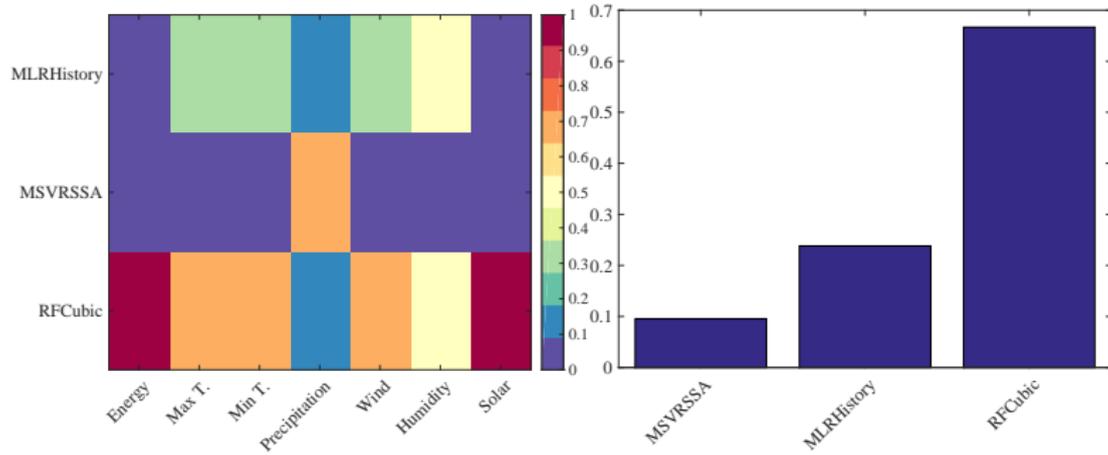
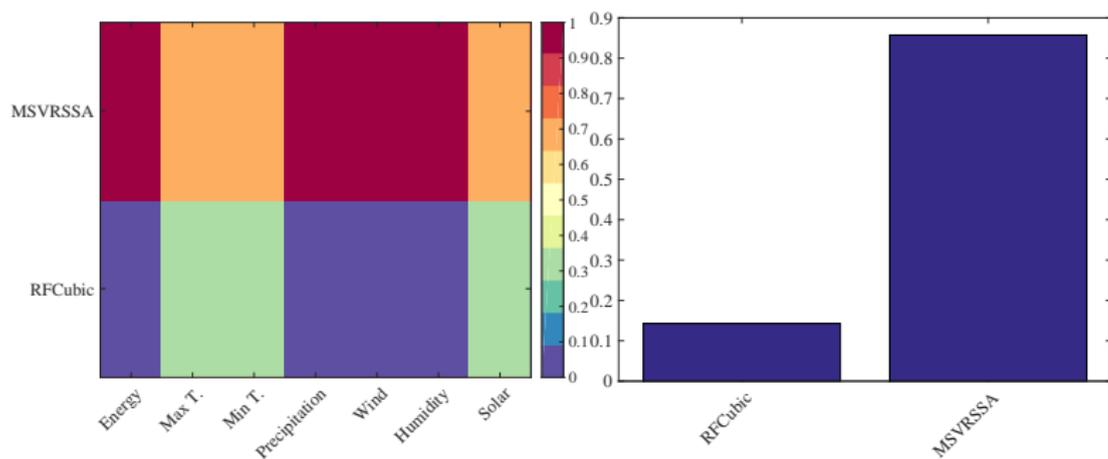
# Best models, train/test residues



# Best models, standard deviations (train/test residues)

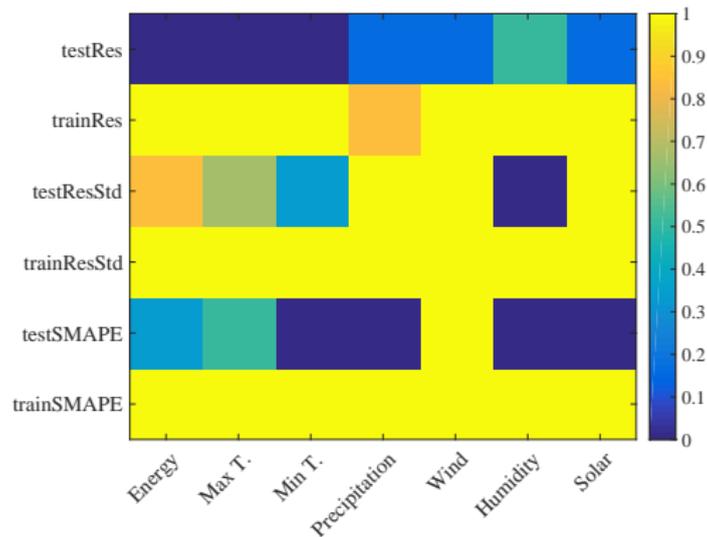


# Best models, train/test SMAPE



# Validation of multiple forecast approach

Additional loss functions: mean residues (test/train), standard deviation of residues (test/train) → 6 loss functions.



Ratio of datasets, where best forecasts outperformed baseline according to a particular error function.

# Conclusion

- ▶ The framework for multiscale time series forecast employs “multiple forecast” approach: all time series are forecasted simultaneously.
- ▶ For better discrimination between experts apply feature generation strategies.
- ▶ Even for such naive approach the results are still better than those of the baseline method.

Stresstest procedure for feature selection algorithms.

*A.M. Katrutsa, V.V. Strijov / Chemometrics and Intelligent Laboratory Systems 2015.*

Feature generation for multiscale time series forecasting. *A. Motrenko, R. Neychev, R. Isachenko, M. Popova, V. Strijov / Submitted to ICDM 2016.*