Feature generation for multiscale time series forecasting

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

Moscow Institute of Physics and Technology

2016

Outline

- Data preprocessing
- Design matrix
- Testing procedure
- Feature generation
- Feature selection

Problem statement

Multiscale data

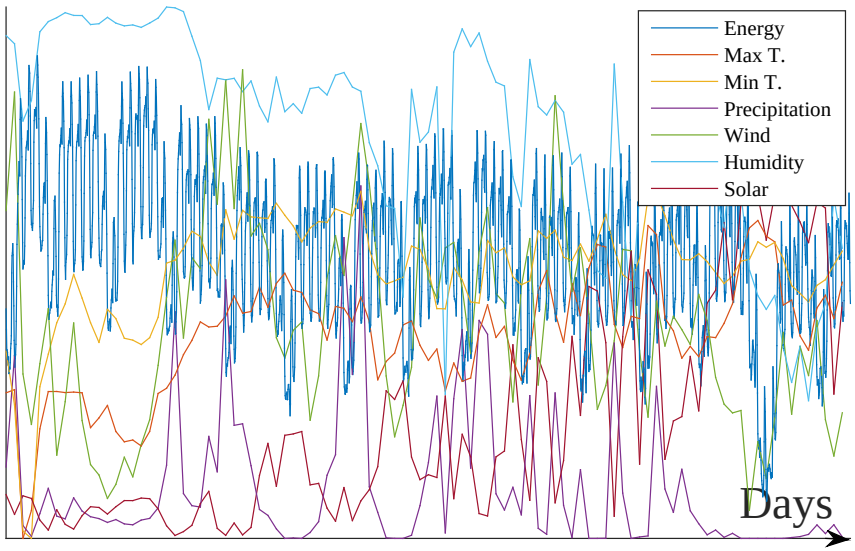
Consider a large set of time series $\mathfrak{D} = \{\mathbf{s}^{(q)} | \; q = 1 \ldots, Q\}$.

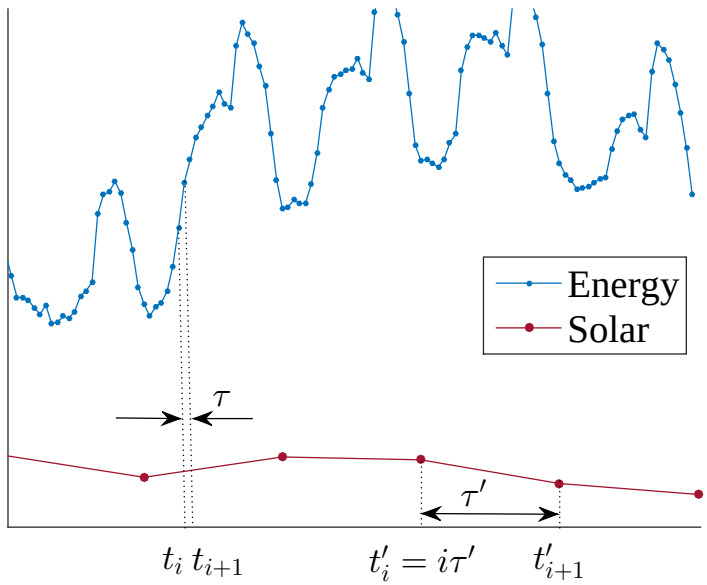
Each real-valued time series $\mathbf{s}$

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

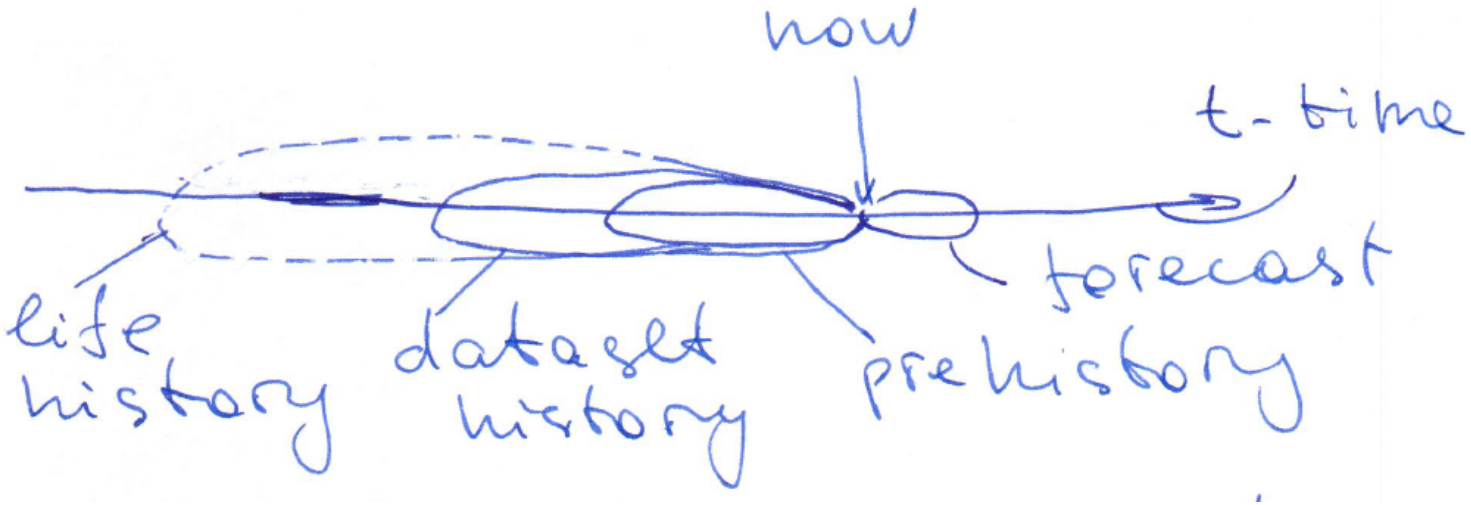
is a sequence of observations of some real-valued signal $s(t)$.

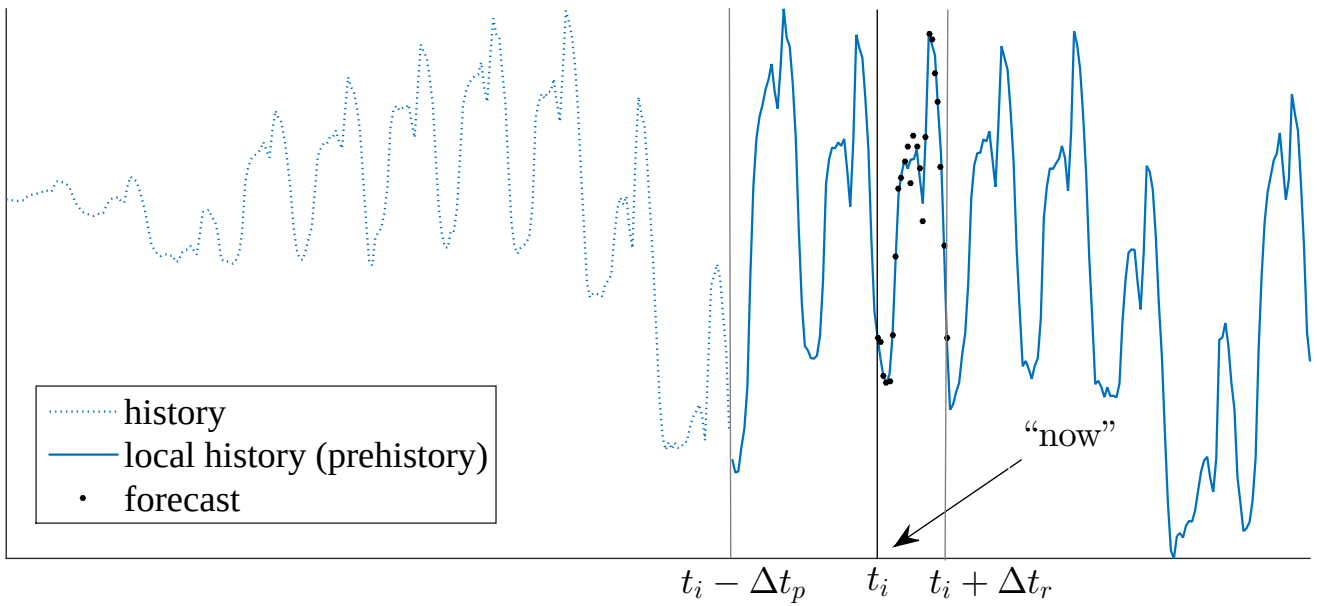
Each time series $\mathbf{s}^{(q)}$ has its own sampling rate $\tau^{(q)}$.



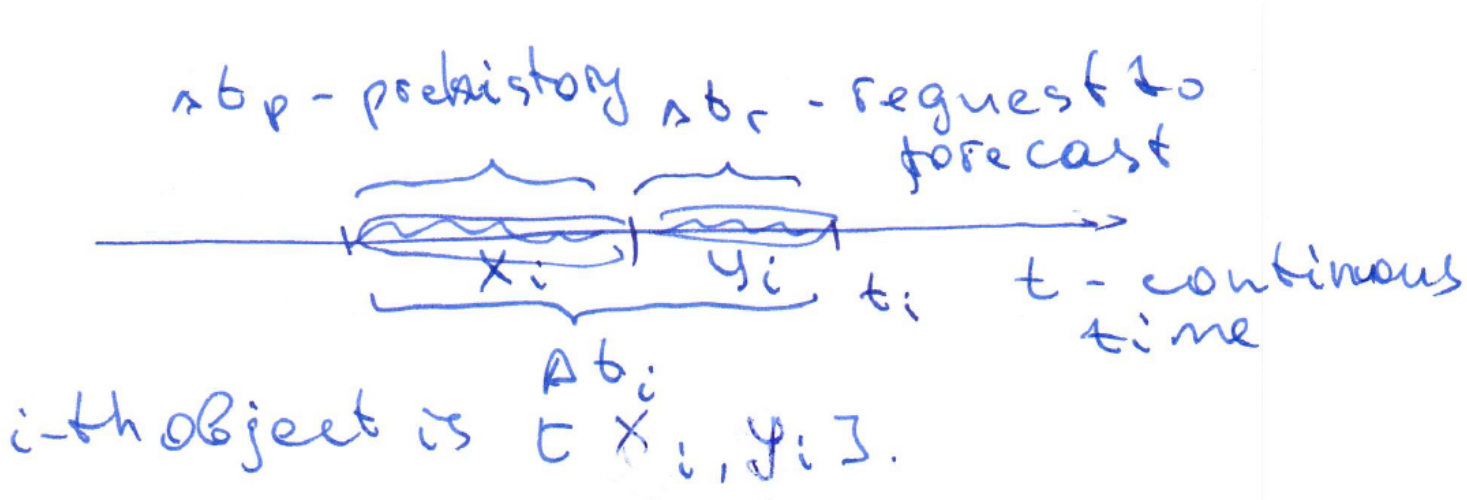


Time series forecasting



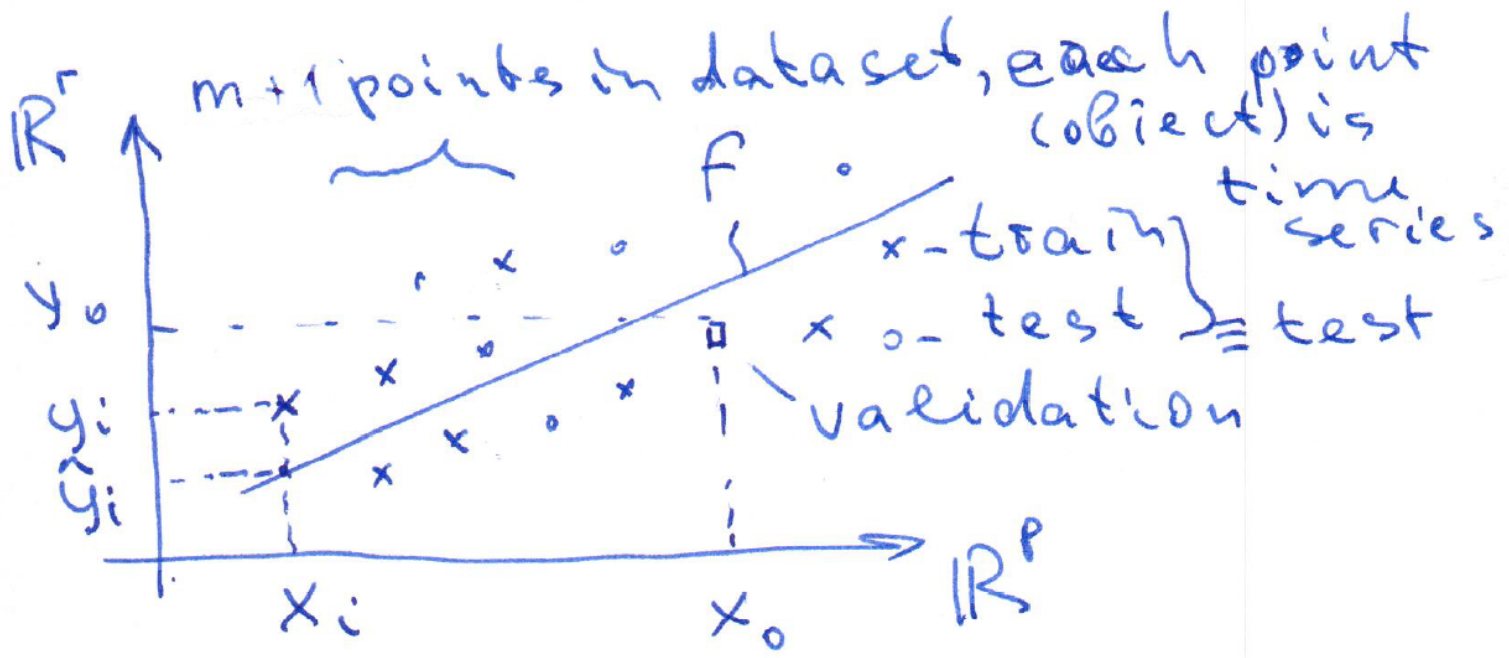


Design matrix



$$[\mathbf{x}_i|\mathbf{y}_i] = \underbrace{s(t_i - \Delta t_\mathrm{r} - \Delta t_\mathrm{p}), \ldots,}_{\mathbf{x}_i} \underbrace{s(t_i - \Delta t_\mathrm{r}), \ldots, s(t_i))}_{\mathbf{y}_i}.$$

Design matrix



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

Regression problem

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

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

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

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

Let ε denote residual vector

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

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

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

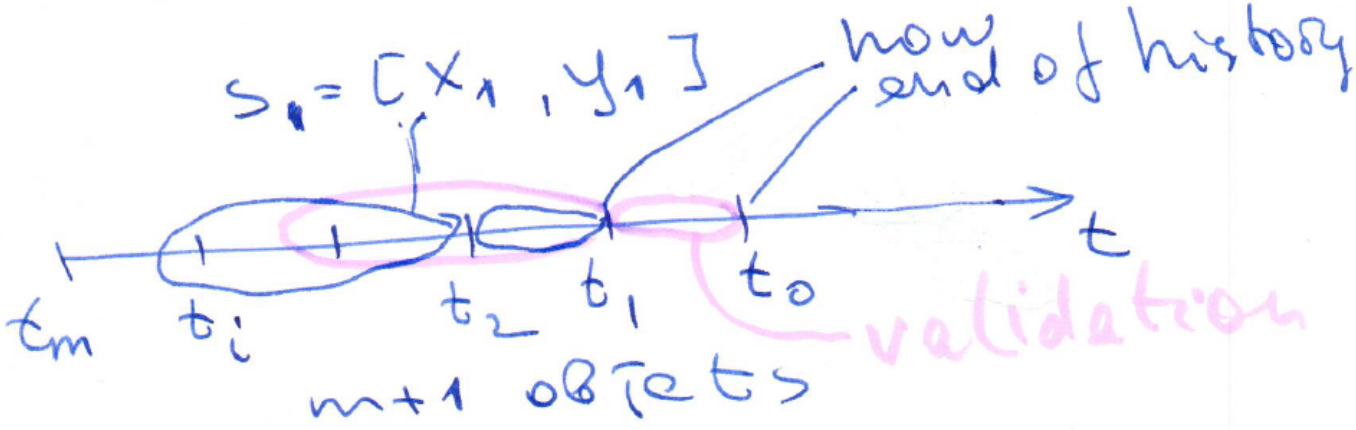
- relative-error metrics (to residues ε^* of a benchmark method):

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

- and scale-free error metrics:

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

Rolling validation



Rolling validation

1) construct the validation vector $\mathbf{x}^*_{\text{val},k}$ for time series of the length Δt_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

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

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 ϕ requires:

- the original features $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_Q\}$,
- 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 $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_Q, \phi_1, \ldots, \phi_N\}$.

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 $\mathbf{x}_i$, $i = 1, \ldots, m$:

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

- 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: $\phi_i = [X_1, \ldots, X_K]$, where

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

Nonparametric transformations: Haar's transform

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

Assume that $n = 2^K$ and init $\phi_{i,j}^{(0)} = \phi_{i,j}^{\prime(0)} = x_{ij}$ for $j = 1, \ldots, n$.
To obtain coarse-graining and fine-graining of the input feature vector $\mathbf{x}_i$, for $k = 1, \ldots, K$ repeat:

- data averaging step

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

- and data differencing step

$$\phi_{i,j}^{\prime(k)} = \frac{\phi_{i,2j}^{\prime(k-1)} - \phi_{i,2j-1}^{\prime(k-1)}}{2}, \quad j = 1, \ldots, \frac{n}{2^k}.$$

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

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 ϕ:

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

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

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

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	$w_2 x^2 + w_1 x + w_0$	1	1	3
Cubic	$w_3 x^3 + w_2 x^2 + w_1 x + w_0$	1	1	4
Logarithmic sigmoid	$1/(w_0 + \exp(-w_1 x))$	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	$w_1 x^{w_2}$	1	1	2
Weibull-2	$w_1 w_2 x^{w_2-1} \exp -w_1 x^{w_2}$	1	1	2
Weibull-3	$w_1 w_2 x^{w_2-1} \exp -w_1 (x - w_3)^{w_2}$	1	1	3
...	...	...	...	...

Monotone functions

- **By grow rate**

Function name	Formula	Constraints
Linear	$w_1 x + w_0$	
Exponential rate	$\exp(w_1 x + w_0)$	$w_1 > 0$
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	$1/(w_0 + \exp(-w_1 x))$	$w_1 > 0$
Softmax	$1/(1 + \exp(-x))$	
Hiberbolic tangent	$\tanh(x)$	
softsign	$\frac{\lvert x \rvert}{1+\lvert x \rvert}$	

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 $\mathbf{s} = [s_1, \ldots, s_T]$ the matrix

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

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 orignal feature vector $\mathbf{x} = [\mathbf{x}^{(1)}, \mathbf{x}^{(Q)}]$ use the parameters $\phi(\mathbf{x}^{(q)})$, $q = 1, \ldots, Q$ 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 $\mathbf{x}_i^{(q)}$, $q = 1, \ldots, Q$ compute k-means centroids $\mathbf{c}_p^{(m)}$, $p = 1, \ldots, P$.
2. With the selected k-means distance function ρ construct the feature vector

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

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

Feature selection

Mixture models

Linear mixture models

Assume the target variable $\mathbf{y}$ is generated by one of K linear models $f_k(\mathbf{x}, \mathbf{w}_k)$. Let the distribution of the target variable $\mathbf{y}$ be a mixture of normal distributions

$$p(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{y}|\mathbf{w}_k^{\mathsf{T}}\mathbf{x}, \beta) =$$

$$\sum_{k=1}^{K} \frac{1}{(2\pi\beta_k)^{n/2}} \exp\left((-\frac{1}{2\beta_k}(\mathbf{y} - \mathbf{w}_k^{\mathsf{T}}X)^{\top}(\mathbf{y} - \mathbf{w}_k^{\mathsf{T}}X) \right).$$

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

$$\boldsymbol{\theta} = [\mathbf{w}_1, \ldots, \mathbf{w}_k, \boldsymbol{\pi}, \beta]^{\mathsf{T}},$$

where $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_k]$ are weights of the models, and $\mathbf{B} = \beta \mathbf{I}_m$ is the covariance matrix for $\mathbf{y}$.

Parameter estimation

To find maximum likelihood estimates of $\hat{\boldsymbol{\theta}}$

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

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

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

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

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

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

Expectation step

Expectation-Maximization algorithm maximizes $\mathrm{E}_Z[p(\mathbf{y}, Z|X, \boldsymbol{\theta})]$, updating parameters estimates $\boldsymbol{\theta}^{(r)}$ in two iterative steps.

E-step: obtain $\mathrm{E}(Z) = \Gamma = [\gamma_{ik}]$. Using Bayesian rule, obtain:

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

Define expectations of joint loglikelihood $\ln p(\mathbf{y}, Z|X, \boldsymbol{\theta})$ with respect to the posteriors distribution $p(Z|\mathbf{y}, \boldsymbol{\theta})$

$$Q^{(r)}(\boldsymbol{\theta}) = \mathrm{E}_Z(\ln p(\mathbf{y}, Z|\boldsymbol{\theta})) = \sum_{i=1}^{m}\sum_{k=1}^{K} \gamma_{ik}^{(r+1)} \left(\ln \pi_k^{(r)} + \ln \mathcal{N}(y_i|\mathbf{x}_i^{\mathsf{T}}\mathbf{w}_k^{(r)}, \beta^{(r)})\right)$$

Maximization step

Expectation-Maximization algorithm maximizes $\mathrm{E}_Z[p(\mathbf{y}, Z|X, \boldsymbol{\theta})]$, updating parameters estimates $\boldsymbol{\theta}^{(r)}$ in two iterative steps.

M-step: update parameters $\boldsymbol{\theta}$, maximizing $Q^{(r)}(\boldsymbol{\theta})$. Maximize function $Q^{(r)}(\boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$ with $\Gamma^{(r+1)}$ fixed:

$$\pi_k^{(r+1)} = \frac{1}{n} \sum_{i=1}^{m} \gamma_{ik}^{(r+1)}.$$

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

$$\beta_k^{(r)} = \arg\max_{\beta} \sum_{i=1}^{m} \gamma_{ik}^{(r+1)} \left(n \ln \beta - \frac{1}{\beta} (y_i - \mathbf{x}_i^{\mathsf{T}} \mathbf{w}_k^{(r+1)})^2 \right).$$

Mixture of Experts

Suppose that each model $f(\mathbf{x}, \mathbf{w}_k)$ generates a sample $(\mathbf{x}, y)$ with some probability $p(k|\mathbf{x}, \mathbf{w})$. Then the following factorization holds

$$p(y|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^{K} p(y, k|\mathbf{x}, \boldsymbol{\theta}) = \sum_{k=1}^{K} p(k|\mathbf{x}, \boldsymbol{\theta}) p(y|k, \mathbf{x}, \boldsymbol{\theta})$$

for $p(y|\mathbf{x}, \boldsymbol{\theta})$.

Here $p(k|\mathbf{x}, \boldsymbol{\theta})$ correspond to weight parameters π_k in mixture models dependent on the inputs $\mathbf{x}$:

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

Resampling time series

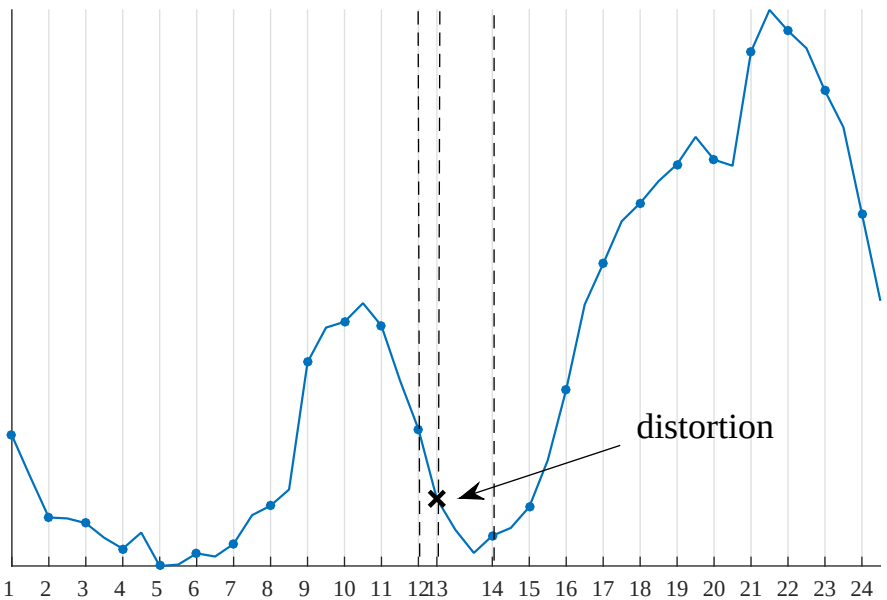
Resampling

The sampling rate might be changing over time, specifically

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.

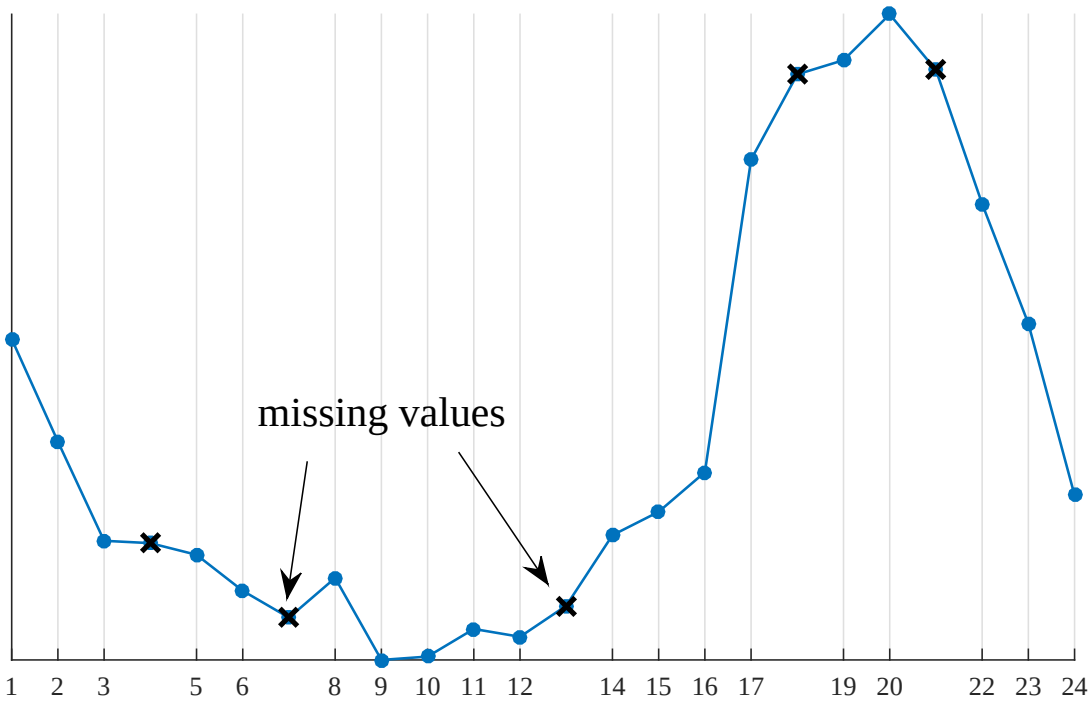


Resampling

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.

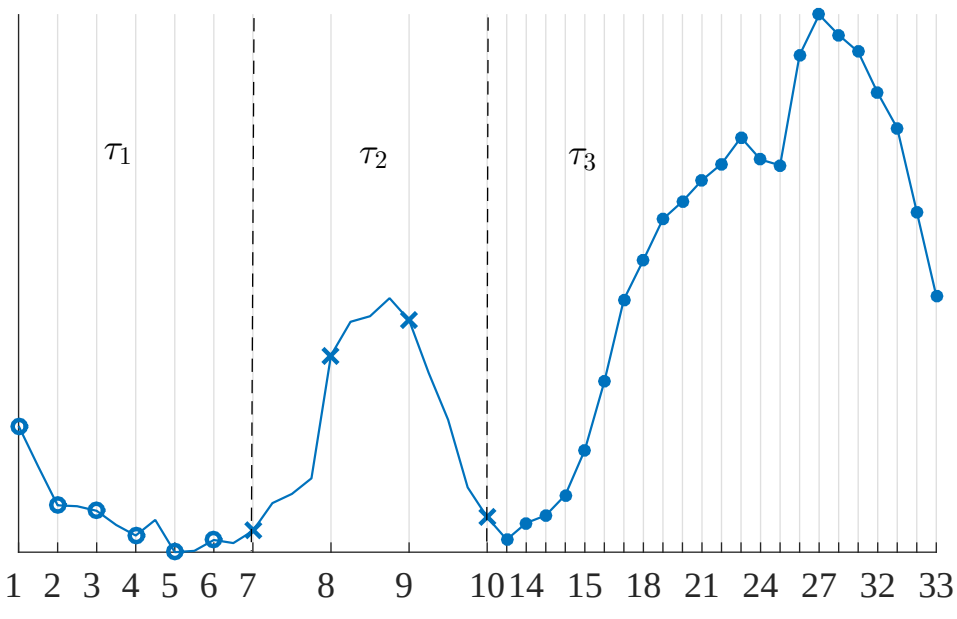


Resampling

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

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

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



Resampling

Piece-wise constant approximation of missing values: