Feature generation for multiscale time series forecasting multimodels

LIG

Technical report (pre-draft)

1 Introduction

The paper investigated behavior of a device, a member of the internet of things. A device is monitored by a set of sensors, which produces large amount of multiscale time series during its lifespan. These time series have various time scales, due to measurements could perform over each millisecond, day, week, etc. The main goal is to forecast the next state of a device.

The investigation assumes the following conditions for a single device unit time series: there are large set of multiscale time series; the sampling rate of a time series is fixed; each time series has its own forecast horizon. Examples of this kind of time series are listed below. (list)

To make an adequate forecasting model hold the following hypothesis: the time history is sufficient long; the time series have auto- and cross-correlation dependencies. The model is static, so there exists a history of optimal size. Each time series could be interpolated by some local model (constant, piece-wise), a that there exist a local approximation model, which could be applied in the case of local data absence.

To forecast time series under the listed conditions autoregressive models constructed in next TODO (list and review state of the art).

2 Problem statement

There give a large set of time series $\mathfrak{D} = \{\mathbf{s}\}$, where each real-valued time series $\mathbf{s}$ has its own sample rate. Assuming each time series could be interpolated introduce a joint sample rate and assign each series $\mathbf{s}$ has minimum one and maximum $T_{\max}$ samples for one time-tick of this rate. Introduce joint time-scale and split it to the following parts, as Fig. shows. The first part is the history itself, the second part is the local history, which forms an object of the object-feature matrix to make a forecast. The third part is a requested forecast time-segment Δt_{r} and the last one is a forecast horizon.

Forecast horizon is a time segment Δt_{h}, which brings an adequate (TODO explain it) forecast $\mathbf{s}(\Delta t_{\mathrm{h}})$ in comparison to the real historical values $\hat{\mathbf{s}}(\Delta t_{\mathrm{h}})$.

Assign to the requested forecast time-segment Δt_{r} a new vector $\mathbf{y}$ of forecasts. This vector contains forecasts to all time series of the set $\{\mathbf{s}\}$. The value of Δt_{r} is given so that each time series $\mathbf{s}$ has minimum one sample. A proportion of $T_{\max}$ samples for time series of high frequency sample rates could be considered, too.

Form an object set at a set of the vectors $\{\mathbf{x}\}$, where each vector $\{\mathbf{x}\}$ collects all the time series over the time Δt_{p}. Here p stands for the local *prehistory*. The vector could (with no necessity) include samples from previous history of any time series as well as any derivatives, which are called generated features. Construct the "object-feature" matrix $\mathbf{X}^*$ for the multiscale autoregressive problem statement as follows. Denote i-th element from the sample set as a row-vector

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

where $s(t)$ is an element of time series $\mathbf{s}$. Denote the other time series from $\mathfrak{D}$ in this segment $(t_i, \dots, t_i - \Delta t_{\mathrm{r}} - \Delta t_{\mathrm{p}})$ as $\mathbf{y}''_i, \mathbf{x}''_i, \dots$ and form the matrix

$$\mathbf{X}^* = \left[\begin{array}{ccc|ccc} \hat{\mathbf{y}}' & \hat{\mathbf{y}}'' & \dots & \mathbf{x}'_0 & \mathbf{x}''_0 & \dots \\ \hline \mathbf{y}'_1 & \mathbf{y}''_1 & \dots & \mathbf{x}'_1 & \mathbf{x}''_1 & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\ \mathbf{y}'_m & \mathbf{y}''_m & \dots & \mathbf{x}'_m & \mathbf{x}''_m & \dots \end{array}\right] = \left[\begin{array}{c|c} \underset{1\times r}{\hat{\mathbf{y}}} & \underset{1\times n}{\mathbf{x}_0} \\ \hline \underset{m\times r}{\mathbf{Y}} & \underset{m\times n}{\mathbf{X}} \end{array}\right].$$

Denote a row from the pair $\mathbf{Y}, \mathbf{X}$ as $\mathbf{y}, \mathbf{x}$ and call these vectors the target and the features. As one must forecast all elements from the target $\mathbf{y}$, only a few elements from the features $\mathbf{x}$

are supposed to be informative in terms of the forecast quality. Denote the index set $\mathcal{J} = \{1, \dots, n\}$ and select the subset of the indexes $\mathcal{A} \in \mathcal{J}$. Introduce the forecasting model

$$\hat{\mathbf{y}}_i = \sum_{k=1}^{K} \pi_{ik} \mathbf{f}_k(\mathbf{w}_{\mathcal{A}_k}, \mathbf{x}_{i\mathcal{A}_k})$$

as some linear combination of K models and call it the *multimodel.* Each model $\mathbf{f}_k$ has its parameters $\mathbf{w}_k$ and selected features $\mathbf{x}_{\mathcal{A}_k}$. The coefficient π_{ik} set a vector $\mathbf{x}_i$ in correspondence to the model $\mathbf{f}_k$, so that

$$\sum_{k=1}^{K} \pi_{ik} = 1 \quad \text{for} \quad i \in \mathcal{I} = \{1, \dots, m\}$$

with two options are to be considered: $\pi \in \{0, 1\}$ and $\pi \in [0, 1]$. Let the forecasting error be

$$S = \sum_{i \in \mathcal{B}_0} \|\hat{\mathbf{y}}_i - \mathbf{y}_i\|_1,$$

where the set of object indexes $\mathcal{I}$ is splatted to the test set $\mathcal{B}_0$ and the train sets,

$$\mathcal{I} = \mathcal{B}_0 \bigsqcup_{k=1}^{K} \mathcal{B}_k.$$

State the forecasting problem as a problem to minimize the error function S given models $\mathbf{f}_1, \dots, \mathbf{f}_K$ by optimizing matrix $\Pi = [\pi_{ik}]$, finite sets $\mathcal{A}_1, \dots, \mathcal{A}_K$ and model parameters $\mathbf{w}_1, \dots, \mathbf{w}_K$ on the sample set with indexes $\mathcal{I} \setminus \mathcal{B}_0$.

3 Special case of the problem

A special case of the problem is an early warning forecasting. There is a special time series $\bar{\mathbf{s}}$ with its element $\bar{s} \in \{0, 1\}$. Here zero is interpreted as a *normal state* of the system and one meas the system goes from normal to the *abnormal* state without return over time t. The problem is to maximize the lapse of the time segment

$$\|\Delta t_{\mathrm{r}}\| \to \max,$$

where the vector

$$[\bar{\mathbf{s}}(\Delta t_{\mathrm{h}}), \bar{\mathbf{s}}(\Delta t_{\mathrm{r}})] = [0, \dots, 0, 0, \dots, 0, 1],$$

which means the system was in the normal state before it changes. Since the quality Q of forecasting time series $\bar{\mathbf{s}}$ depends on $\|\Delta t_{\mathrm{r}}\|$ (the letter time lapse before the warning the higher the forecasting quality [ref]) the minimum level of quality must be set. Let the minimum forecasting quality be

$$Q\left\{\left(\hat{\mathbf{s}}_i(\Delta t_{\mathrm{r}}), \bar{\mathbf{s}}_i(\Delta t_{\mathrm{r}})\right) \mid i \in \mathcal{B}_0\right\} = \mathrm{AUC} = Q_{\mathrm{req}}.$$

4 Time searies resampling

For the time series of sample rate that is changing, unstable, non-rational to the common rate as well as for the time series with missing values the following procedure should be applied.

Let the time t be in continous set $\mathbb{R}^1_+$ and the time series s be piece-wise constant. There are three possibilities to create such time series from a discrete-values one: 1) the constant goes after the sample $s(t)$, 2) before the sample, 3) in the neighborhood of the sample. See red, green and blue lines in the Figure 4.

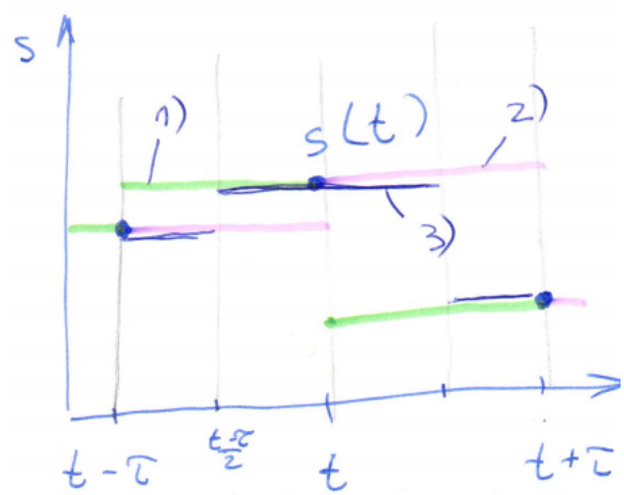

Figure 1: Piece-wise representation of a time series

This assumptions helps introducing a new sampling rate and eliminates the problem of missing values, since the previous (next, current in the terms of Fig. 4) value holds continuously until the following comes. The constant model could be developed into more comples one: a piece-wise, quadratic or cubic spline with its nodes in the time-ticks or over the time-ticks according to the following criterions: 1) NyquistShannon theorem, 2) Fisher-Neyman theorem. The following optimization problem returns the new sampling

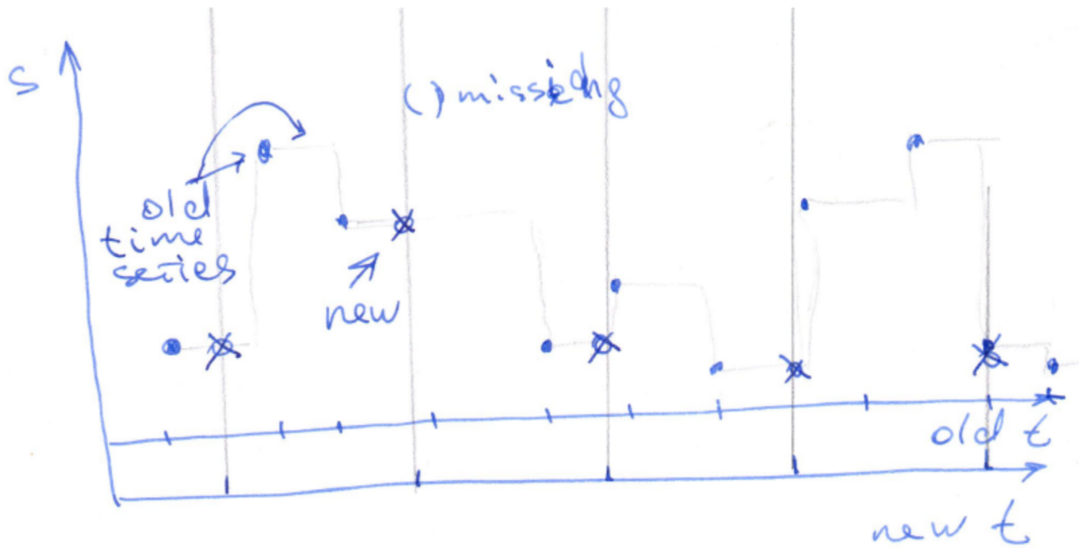

Figure 2: Resample time series of variating sample rate into the fixed one with the optimal period

rate:

todo

This fixed rate is used to obtain a resampled time series with regular time-ticks.

4.1 NyquistShannon resampling criterion

TODO

4.2 Fisher-Neyman resampling criterion

TODO

4.3 Wavelet resampling

In signal processing a common way to change resolution of a signal is to use a combination of upsampling and decimation (FFT wavelet transform and downsampling). Suppose that desired sampling rate u is fixed, that is, we would like to approximate $U = uT_{\max}$ uniformly sampled observations $\tilde{s}(\tilde{t}_i),\ \ i = 1, \ldots U$ of the time series $s(t_i),\ \ i = 1, \ldots, T_{\max}$. Let $G = \{t_1, \ldots, T_{\max}\}$ and $\tilde{G} = \{\tilde{t}_1, \ldots, T_{\max}\}$ denote the current and the desired grids. The first step to rescaling $\mathbf{s} \to \tilde{s}$ is the piecewise approximation $\hat{\mathbf{s}}$ (see Fig. **??** and Fig. **??**) of $\mathbf{s}$ at $\hat{G} = G \cup \tilde{G}$. To increase the smoothness of piecewise approximation $\hat{\mathbf{s}}$ we apply low pass FFT wavelet filtering to $\hat{\mathbf{s}}$ and then downsample the output to $\tilde{G}$.

TimeSeriesRescaling()
Data: $\mathbf{s}$. Parameters: desired grid $\tilde{G} = \{\tilde{t}_1, \dots \tilde{t}_U\}$.
Result: Resamples time series $\tilde{\mathbf{s}}$.
Form the new grid $\hat{G} = G \cup \tilde{G}$;
Upsample time series, using piecewise approximation:
$\hat{\mathbf{s}} \leftarrow$ **Piecewise**$(\mathbf{s}, G, \tilde{G})$;
Zero-pad $\hat{\mathbf{s}}$, so that $|\hat{G}| = 2^N$, where $N = \lceil \log_2(|\hat{G}|) \rceil$;
Apply low pass filtering $\hat{\mathbf{s}}$:
$\hat{\mathbf{s}}_{\text{lf}} =$ **LowPassFFTFiltering**$(\hat{\mathbf{s}})$;
Downsample $\hat{\mathbf{s}}_{\text{lf}}$ to $\tilde{G}$:
$\tilde{\mathbf{s}} = [\hat{s}_{lf}(\tilde{t}_1), \dots,],\ \tilde{t}_i \in \tilde{G}$.

LowPassFFTFiltering()
Data: Time series, $\mathbf{s}$. Parameters: cut-off value $w_{\max}$ for high frequencies.
Result: Filtered time series $\tilde{\mathbf{s}}_{\text{lf}}$.
Find FFT coefficients $a_j,\ b_j$ for $j = 1, \dots, N$ for $\mathbf{s}$;
Set $a_j = 0,\ b_j = 0$, for $w_j > w_{\max}$;
Reconstruct the time series, using inverse FFT.

Algorithm 1: FFT rescaling procedure.

5 Feature generation

List of procedures for constructing the feature and the object sets will be placed here. Discussion point: vector **y remains always unchanged**.

The feature set $\mathcal{J} = \bigcup\limits_k \mathcal{A}_k$ includes

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.

The object set $\mathcal{I} = \bigsqcup\limits_k \mathcal{B}_k$ includes

1) the local history,

2) parametric local models and their residuals (including ones from previous iterations),

3) DTW-shifted local history as a local forecasting procedure,

4) aggregated subsets of time series.

Denote the generated feature vector as $\boldsymbol{\phi}$. This vector consists of concatenated row-vectors $\boldsymbol{\phi} = [\boldsymbol{\phi}', \boldsymbol{\phi}'', \dots]$, which corresponds to time series local histories $\mathbf{s} = [\mathbf{s}', \mathbf{s}'', \dots]$, modified with set of transformations $\mathfrak{G}$. The elements $g : \mathbf{s} \to \boldsymbol{\phi}$ of this set are listed below.

5.1 Transformations of local history

The tables 1, 2, 3, 4, 5 list the time series transformation functions. There are non-parametric and parametric procedures to generate features. For the parametric functions $g = g(\mathbf{b}, s)$ the default values of the parameters $\mathbf{b}$ are assigned empirically.

The parametric procedure request two optimization problem statements of the model parameters $\mathbf{w}$ and the primitive function parameters $\mathbf{b}$. The first one fixes the vector $\hat{\mathbf{b}}$, collected over all the primitive functions $\{g\}$, which generate features $\boldsymbol{\phi}$:

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

The second one optimizes the transformation parameters $\hat{\mathbf{b}}$ given obtained model parameters $\mathbf{w}$

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

This procedure repeats two problems until vectors $\hat{\mathbf{w}}, \hat{\mathbf{b}}$ converge. The initial values of vector $\mathbf{b}$ (are shown in table ??). Due to the various origins of the time series and their transformations the residual vector should be normalized:

$$\boldsymbol{\varepsilon}' = \frac{\hat{\mathbf{y}}' - \mathbf{y}'}{|\mathbf{y}'| \cdot \|\mathbf{y}'\|_2^{\frac{1}{2}}}.$$

It does not change the number elements in the vectors, $|\boldsymbol{\phi}'| = |\mathbf{s}'|$.

5.2 Convolutions, statistics and parameters of local history

The listed feature generation functions convolves time series, so they reduce the dimentionality $|\boldsymbol{\phi}' = \mathbf{g}(\mathbf{s}')| < |\mathbf{s}'|$.

5.3 Parameters of local history forecast

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

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

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

$$\boldsymbol{\phi}' = \arg\min \|\mathbf{h} - \mathbf{H}\boldsymbol{\phi}\|_2^2.$$

For the time series $[\mathbf{s}'_i, \mathbf{s}''_i, \dots]$ use the parameters $[\boldsymbol{\phi}'_i, \boldsymbol{\phi}''_i]$ as the features.

5.4 Distances to centroids of local clusters

This procedure applies the kernel trick to the time series. For given local history time series $\mathbf{s}'_1, \dots, \mathbf{s}'_m$ compute k-means centroids $\mathbf{c}'_1, \dots, \mathbf{c}'_P$. With the selected k-means distance function ρ construct the feature vector

$$\boldsymbol{\phi}'_i = [\rho(\mathbf{c}'_1, \mathbf{s}'_i), \dots, \rho(\mathbf{c}'_P, \mathbf{s}'_i)] \in \mathbb{R}_+^P.$$

This k-means of another clustering procedure may use internal parameters, so that there are no parameters to be included to the feature vector or to the forecasting model.

6 Feature selection

TODO

Table 1: Must-try functions.

Formula	Output dimension	# of arguments	# of parameters
$\sqrt{x}$	1	1	0
$x\sqrt{x}$	1	1	0
$\arctan x$	1	1	0
$\ln x$	1	1	0
$x \ln x$	1	1	0

Table 2: List of elementary functions.

Function name	Formula	Output dimension	# of arguments	# of parameters
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

7 Mixture models

Let $D = (X, \mathbf{y})$ denote the data, where $X = [\mathbf{x}_1^\mathsf{T}, \dots, \mathbf{x}_i, \dots, \mathbf{x}_m^\mathsf{T}]^\mathsf{T}$, denotes the inputs $\mathbf{x}_i \in \mathbb{R}^n$, $\mathbf{y}$ denotes the targets $y_i \in Y$. The task is to estimate y_i, given $\mathbf{x}_i$. Assuming

Table 3: Monotone functions.

By growth rate					
Function name	Formula	Output dimension	# of arguments	# of parameters	Constraints
Linear	$w_1 x + w_0$	1	1	2	
Exponential rate	$\exp(w_1 x + w_0)$	1	1	2	$w_1 > 0$
Polynomial rate	$\exp(w_1 \ln x + w_0)$	1	1	2	$w_1 > 1$
Sublinear polynomial rate	$\exp(w_1 \ln x + w_0)$	1	1	2	$0 < w_1 < 1$
Logarithmic rate	$w_1 \ln x + w_0$	1	1	2	$w_1 > 0$
Slow convergence	$w_0 + w_1/x$	1	1	2	$w_1 \neq 0$
Fast convergence	$w_0 + w_1 \cdot \exp(-x)$	1	1	2	$w_1 \neq 0$
Other					
Soft ReLu	$\ln(1 + e^x)$	1	1	0	
Sigmoid	$1/(w_0 + \exp(-w_1 x))$	1	1	2	$w_1 > 0$
Nonparametric log-sigmoid	$1/(1 + \exp(-x))$	1	1	0	
Hiberbolic tangent	$\tanh(x)$	1	1	0	
softsign	$\frac{\lvert x\rvert}{1+\lvert x\rvert}$	1	1	0	

linear model f with gaussian noise

$$y = f(\mathbf{x}, \mathbf{w}) + \varepsilon, \quad f(\mathbf{x}, \mathbf{w}) = \mathbf{w}^\mathsf{T}\mathbf{x}, \ \varepsilon \sim \mathcal{N}(0, \beta) \Rightarrow y \sim \mathcal{N}(\mathbf{w}^\mathsf{T}\mathbf{x}, \beta),$$

obtain the maximum likelihood estimate

$$\hat{y} = \hat{\mathbf{w}}^\mathsf{T}\mathbf{x}, \quad \hat{\mathbf{w}} = \arg\max_{\mathbf{w}} \frac{1}{2\beta} \sum_{i=1}^{m} (y_i - \mathbf{w}^\mathsf{T}\mathbf{x}_i)^2$$

for the output.

Table 4: Multivariate.

Bivariate				
Plus	$x_1 + x_2$	1	2	0
Minus	$x_1 - x_2$	1	2	0
Product	$x_1 \cdot x_2$	1	2	0
Division	$\frac{x_1}{x_2}$	1	2	0
	$x_1 \sqrt{x_2}$	1	2	0
	$x_1 \ln x_2$	1	2	0
Multivariate				
Sum of products	$\sum_{i,\, j} x_i x_j$	1	$n \geq 2$	0
Sum of products	$\sum_{i,\, j,\, k} x_i x_j x_k$	1	$n \geq 3$	0
Sum of Gaussians	$\sum_{j=1}^{n} a_j \exp(-\frac{(x_j - b_j)^2}{c_j})$	1	n	$3n$
Polynomial	$\sum_{j=0}^{n} a_j x^j$	1	1	n
Rational polynomial	$\frac{\sum_{j=0}^{n} a_j x^j}{x^m + \sum_{j=0}^{m-1} b_j x^j}$	1	1	$n+m+1$

Table 5: Data statistics.

sum	$\sum_i x_i$	1	m	0
mean	$(\sum_i x_i)/m$	1	m	0
min	$\min_i x_i$	1	m	0
max	$\max_i x_i$	1	m	0
std	$\frac{1}{m-1} \sqrt{\sum_i (x_i - \text{mean}(x))^2}$	1	m	0
hist	$\sum_i [X_{j-1} < x_i \leq X_j]$	n	m	$n-1$
conv	$\sum_j x_{i-j} w_j$	1	$m-n+1$	$n \leq m$
FFT coefficients		n	m	1

7.1 EM-algorithm for 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

$$p(\mathbf{y}|\mathbf{x}, \mathbf{s}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\mathbf{y}|\mathbf{w}_k^{\intercal}\mathbf{x}, \beta) = \sum_{k=1}^{K} \frac{1}{(2\pi\beta_k)^{n/2}} \exp\left((-\frac{1}{2\beta_k}(\mathbf{y} - \mathbf{w}_k^{\intercal}X)^{\top}(\mathbf{y} - \mathbf{w}_k^{\intercal}X)\right). \quad (1)$$

Here $\mathbf{s}$ denotes the concatenated vector of parameters:

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

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. The goal is to find parameters vector $\hat{\mathbf{s}}$ which optimizes log-likelihood function for given data set D

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s}} \ln p(\mathbf{y}|\mathbf{s}), \quad \ln p(\mathbf{y}|\mathbf{s}) = \sum_{i=1}^{m} \ln \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{y}|\mathbf{w}_k^\mathsf{T}\mathbf{x}_i, \beta) \right). \tag{2}$$

To obtain maximum likelihood estimates (2) for parameter $\mathbf{s}$ of the model (1), let us 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 $p(\mathbf{y}, Z|X, \mathbf{s})$ takes the form

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

$$= \sum_{i=1}^{m}\sum_{k=1}^{K} z_{ik} \left(\ln \pi_k - \frac{1}{2\beta}(y_i - \mathbf{w}_k^\mathsf{T}\mathbf{x}_i)^2 + \frac{n \ln \beta}{2} + \text{const} \right).$$

Since $p(\mathbf{y}, Z|X, \mathbf{s})$ depends on random variables z_{ik}, instead of $p(\mathbf{y}|X, \mathbf{s})$ maximize the expected loglikelihood of the observed data D:

$$\mathsf{E}_Z[p(\mathbf{y}, Z|X, \mathbf{s})] = \sum_{i=1}^{m}\sum_{k=1}^{K} \gamma_{ik} \left(\ln \pi_k - \frac{1}{2\beta}(y_i - \mathbf{w}_k^\mathsf{T}\mathbf{x}_i)^2 + \frac{n \ln \beta}{2} \right), \quad \gamma_{ik} = \mathsf{E}[z_{ik}|\mathbf{y}, X].$$

Finally, apply Expectation-Maximization algorithm to maximize $\mathsf{E}_Z[p(\mathbf{y}, Z|X, \mathbf{s})]$ updating parameters estimates $\mathbf{s}^{(r)}$ in two iterative steps.

E-step: obtain $\mathsf{E}(Z)$. Let $\Gamma = [\gamma_{ik}]$ be a matrix of posterior probabilities that i-th sample is generated by k-th model. Using Bayesian rule, obtain

$$\gamma_{ik}^{(r+1)} = \mathsf{E}(z_{ik}) = p(k|\mathbf{x}_i, \mathbf{s}^{(r)}) = \frac{\pi_k \mathcal{N}(y_i|\mathbf{x}_i^\mathsf{T}\mathbf{w}_k^{(r)}, \beta^{(r)})}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(y_i|\mathbf{x}_i^\mathsf{T}\mathbf{w}_k^{(r)}, \beta^{(r)}).} \tag{3}$$

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

$$Q^{(r)}(\mathbf{s}) = \mathsf{E}_Z(\ln p(\mathbf{y}, Z|\mathbf{s})) = \sum_{i=1}^{m}\sum_{k=1}^{K}\gamma_{ik}^{(r+1)}\left(\ln \pi_k^{(r)} + \ln \mathcal{N}(y_i|\mathbf{x}_i^{\mathsf{T}}\mathbf{w}_k^{(r)}, \beta^{(r)})\right). \tag{4}$$

M-step: update parameters $\mathbf{s}$, maximizing $Q^{(r)}(\mathbf{s})$. Maximize function $Q^{(r)}(\mathbf{s})$ with respect to $\mathbf{s}$ with $\Gamma^{(r+1)}$ fixed. First, optimize π_k, which is constrained as $\sum_{k=1}^{K}\pi_k = 1$. Using Lagrange multipliers, obtain the following estimation

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

Next, maximize $Q^{(r)}$ with respect to $\mathbf{w}_k$ for k-th model. With π_k fixed maximizing (4) is equivalent to

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

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

7.2 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

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

for $p(y|\mathbf{x}, \mathbf{s})$. Here $p(k|\mathbf{x}, \mathbf{s})$ correspond to weight parameters π_k in (1) dependent on the inputs $\mathbf{x}$. Assuming normal linear models $f(\mathbf{x}, \mathbf{w}_k)$ or, equivalently, normal distributions $p(y|\mathbf{x}, \mathbf{w}_k) = \mathcal{N}(y|\mathbf{w}_k^{\mathsf{T}}, \beta)$, obtain

$$p(\mathbf{y}|\mathbf{x}, \mathbf{s}) = \sum_{k=1}^{K}\pi_k(\mathbf{x}, \mathbf{v}_k)\mathcal{N}(\mathbf{y}|\mathbf{w}_k^{\mathsf{T}}\mathbf{x}, \beta), \tag{5}$$

where

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

The difference between mixture of experts model (5) and mixture model (1) in that model weights π_k depend on inputs $\mathbf{x}$ in mixture of experts. Similarly, EM-procedure for mixture of experts differs from EM-procedure for mixture models in the way γ_{ik} are optimized in M-step.

Data: $(\mathbf{x}_i, y_i)$, $i = 1, \dots, m$. Parameters: number of experts K.
Result: Parameters $\mathbf{s}$ of the model (5).
Initialize $[\mathbf{w}, \beta, \mathbf{v}] \equiv \mathbf{s} = \mathbf{s}^{(0)}$, $r = 0$;
while $\mathbf{s}$ *keeps changing* **do**

E step: compute hidden variables $\gamma_{ik}^{(r+1)}$, the expectation of the indicator variables, using (3);

M step: find new parameter estimates

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

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

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

end

Algorithm 2: EM-algorithm for mixture of experts.

7.3 Distance between two models of N time series

Introduce a distance function $\rho(f_k, f_l)$ between two models. Use the Jensen-Shannon divergence; $\rho_{kl} \in [0, 1]$ is a metric:

$$\rho(p_k \| p_l) = 2^{-1} D_{\mathrm{KL}}(p_k \| p') + 2^{-1} D_{\mathrm{KL}}(p' \| p_l),$$

where $p' = 2^{-1}(p_k + p_l)$ and $p_k \stackrel{\text{def}}{=} (p(\mathbf{w}|D, A, B, f_k)$. The non-symmetric Kullback-Leibler divergence is

$$D_{\mathrm{KL}}(p \| p') = \int\limits_{\mathbf{w} \in \mathbb{W}} p'(\mathbf{w}) \ln \frac{p(\mathbf{w})}{p'(\mathbf{w})} d\mathbf{w}.$$

8 Computational experiment

The goal of the experiment is to compare the following four approaches to the multiscale forecasting: 1) Bayesian mixture model approach, 2) random multumodel, 3) vector random decision forest and 4) vector adaboost. The last two algorithms are modifications of [link]. The modifications are needed to produce the vector of multiscale time series as their

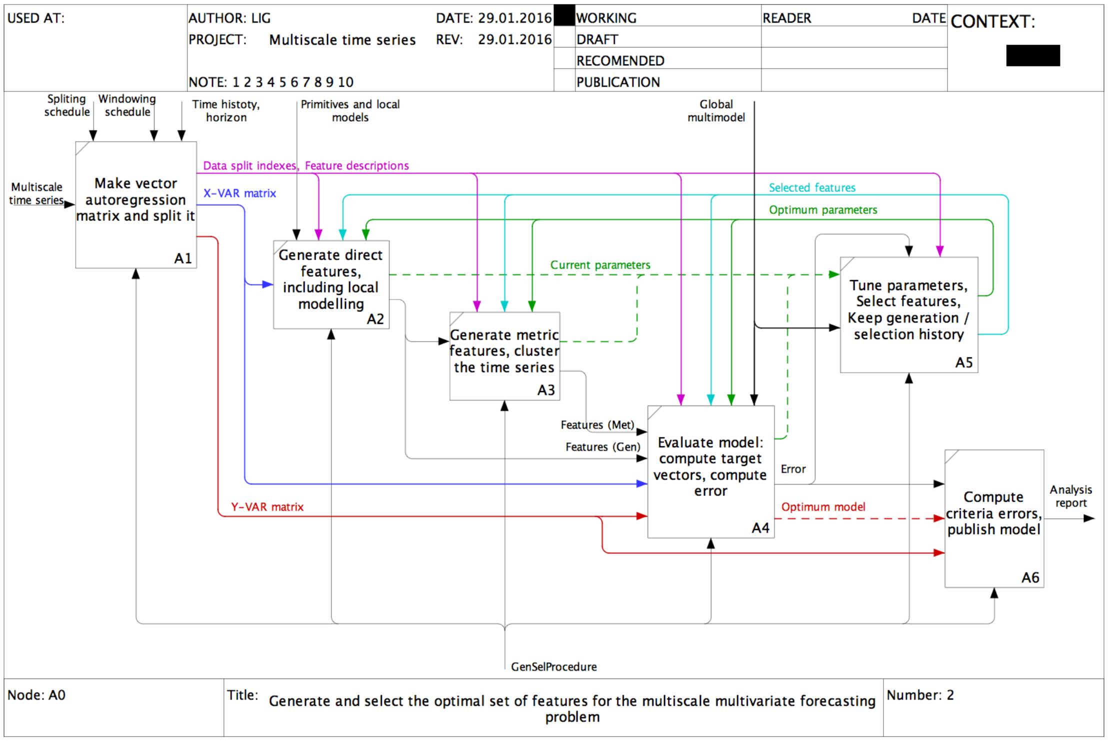

Figure 3: Multiscale forecasting pipeline.

outputs. The experiment is performed on 1) non-modified autoregression data and on 2) data with additionally generated features as it is described in the corresponding section.

9 Appendix: Discrete genetic algorithm for feature selection (will be converted to bootstrap random linear multimodel algorithm)

1. There are set of binary vectors $\{\mathbf{a}_1, \dots, \mathbf{a}_P\}$, $\mathbf{a} \in \{0, 1\}^n$;

2. get two vectors $\mathbf{a}_p, \mathbf{a}_q$, $p, q \in \{1, \dots, P\}$;

3. chose random number $\nu \in \{1, \dots, n-1\}$;

4. split both vectors and change their parts:

$$[a_{p,1}, \dots, a_{p,\nu}, a_{q,\nu+1}, \dots, a_{q,n}] \to \mathbf{a'}_p,$$

$$[a_{q,1},\dots,a_{q,\nu},a_{p,\nu+1},\dots,a_{p,n}] \to \mathbf{a}'_q;$$

5. choose random numbers $\eta_1,\dots,\eta_Q \in \{1,\dots,n\}$;
6. invert positions $\eta_1,\dots,\eta_Q$ of the vectors $\mathbf{a}'_p, \mathbf{a}'_q$;
7. repeat items 2-6 $P/2$ times;
8. evaluate the obtained models.

Repeat R times; here P, Q, R are the parameters of the algorithm and n is the number of the corresponding model features.

10 Appendix: Mixture modelling under random bootstrapped models

Denote the indexes of objects as $\{1,\dots,i,\dots,m\} = \mathcal{I}$, the split $\mathcal{I} = \mathcal{B}_1 \sqcup \dots \sqcup \mathcal{B}_K$ and the indexes of features as $\{1,\dots,j,\dots,n\} = \mathcal{J}$, the active set $\mathcal{A}_k \subseteq \mathcal{J}$.

Let the regression model

$$\mathbf{f} : (\mathbf{w},\mathbf{x}) \mapsto \mathbf{y};$$

with the selected model of optimal structure

$$\mathsf{E}(\mathbf{y}_i|\mathbf{x}) = \mathbf{W}_{\mathcal{A}}\mathbf{x}_i.$$

The multimodel $\mathfrak{f}$ is a set of the models $\mathfrak{f} = \{\mathbf{f}_k \quad | k = 1,\dots,K\}$, such that for each k

$$\mathsf{E}(y_{i\in\mathcal{B}_k}|\mathbf{x}) = \mathbf{W}_{\mathcal{A}_k}\mathbf{x}_{i\in\mathcal{B}_k} \qquad \text{with} \quad \mathcal{I} = \sqcup_{k=1}^K \mathcal{B}_k \ni i.$$

State the multimodel selection problem as follows. An optimal single model is

$$\hat{\mathbf{f}}(\mathbf{w},\mathbf{x}) = \arg\max_{\mathcal{A}\subseteq\mathcal{J}} \mathcal{E}\left(\mathbf{f}(\mathbf{w}_{\mathcal{A}},\mathbf{x})\right),$$

where $\mathcal{E}$ denotes the model evidence in coherent Bayesian inference. An optimal multilevel model i s

$$\hat{\mathfrak{f}}(\mathbf{w}_1,\dots,\mathbf{w}_K,\mathbf{x}) = \arg\max_{\sqcup_{k=1}^K \mathcal{B}_k = \mathcal{I}} \prod_{k=1}^K \mathcal{E}\left(\mathbf{f}(\mathbf{w}_k,\mathbf{x}_{\mathcal{B}_k})\right).$$

~~The model difference must be statistically significant~~

$$\mathcal{F} \supset \hat{\mathfrak{f}} = \arg \max_{\mathcal{B}_1, \mathcal{B}_2 \subset \mathcal{B}} \rho(f_1, f_2)$$

~~given set of indices $\hat{\mathcal{A}}$, such that~~

$$\hat{\mathcal{A}} = \arg \max_{\mathcal{A} \subseteq \mathcal{J}} \mathcal{E}\left(\mathbf{f}_1(\mathbf{w}_{\mathcal{A}}, \mathbf{x}^{\mathcal{B}_1})\right) \mathcal{E}\left(\mathbf{f}_2(\mathbf{w}'_{\mathcal{A}}, \mathbf{x}^{\mathcal{B}_2})\right).$$