Multiscale time series forecasting using vector auto-regression models

Laboratoire d'Informatique de Grenoble

January 15$^{\text{th}}$, 2016

There given

- a large set of time series,
- the time series are multiscale,
- the time series may vary their sample rate,
- there is a long history of the time series,

Statistical assumptions:

- the time series may have cross- and autho-correlation,
- 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).

One has to forecast all the time series, with minimum MAPE on the test sample set.

$$\mathbf{X}^* = \left[\begin{array}{c|c} \underset{1\times h}{\mathbf{y}} & \underset{1\times n}{\mathbf{x}_{m+1}} \\ \hline \underset{m\times h}{\mathbf{Y}} & \underset{\mathcal{J}=1\dots n}{\underset{m\times n}{\mathbf{X}}} \quad \mathcal{I} \end{array}\right].$$

Here

- $\mathbf{y}$ is a vector of the forecast; it contains all required time series (according to expected time ticks),
- $\mathbf{x}$ is a vector of local history (with generated features); it contains minimum necessary history to make required forecast,
- $\mathbf{Y}, \mathbf{X}$ are the history time series (with aggregated time series); $\mathbf{X}$ contains generated features.

The autoregressive matrix for multi-variable forecasting

$$\mathbf{X}^* = \left[\begin{array}{c|cc} \underset{1\times h}{\mathbf{y}} & \underset{1\times n}{\mathbf{x}_{m+1}} & \\ \hline \underset{m\times h}{\mathbf{Y}} & \underset{\mathcal{J}=1\dots n}{\underset{m\times n}{\mathbf{X}}} & \mathcal{I} \end{array}\right].$$

The models:

1. linear $\mathbf{y} = \mathbf{W}^\mathsf{T}\mathbf{x}$, where $\mathbf{W}$ is trained as $\mathbf{Y} = \mathbf{W}^\mathsf{T}\mathbf{X}$,
2. neural network, etc. $\mathbf{y} = \mathbf{f}(\mathbf{W}, \mathbf{x})$,
3. mixture $\mathbf{y}_i = \sum_k \pi_k \mathbf{f}(\mathbf{w}_k, \mathbf{x}_i), \quad i \in \mathcal{I}$,
4. multi-model $\mathbf{y}_i = \sum_k \pi_{ik} \mathbf{f}_k(\mathbf{w}_{\mathcal{A}_k}, \mathbf{x}_{i\mathcal{A}_k})$,
   $i \in \mathcal{I} = \bigsqcup_k \mathcal{B}_k, \quad j \in \mathcal{A} \subset \mathcal{J}$,

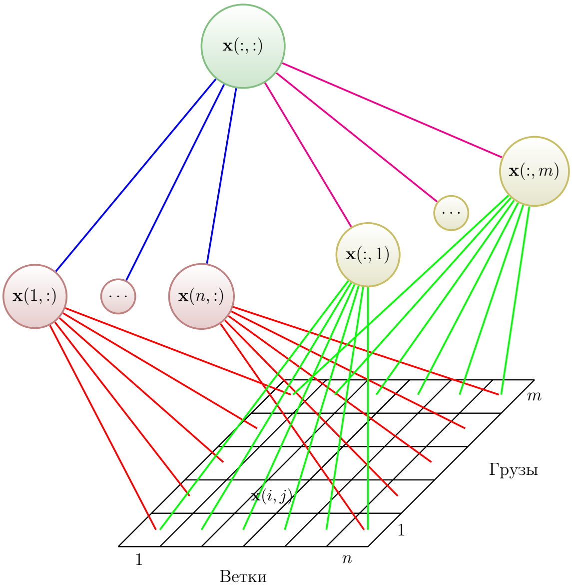
Constructing the feature and the object sets

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 classes of time series.



$$x_t(:,:) = \sum_{i=1}^{n} x_t(i,:);$$

$$x_t(:,:) = \sum_{j=1}^{m} x_t(:,j);$$

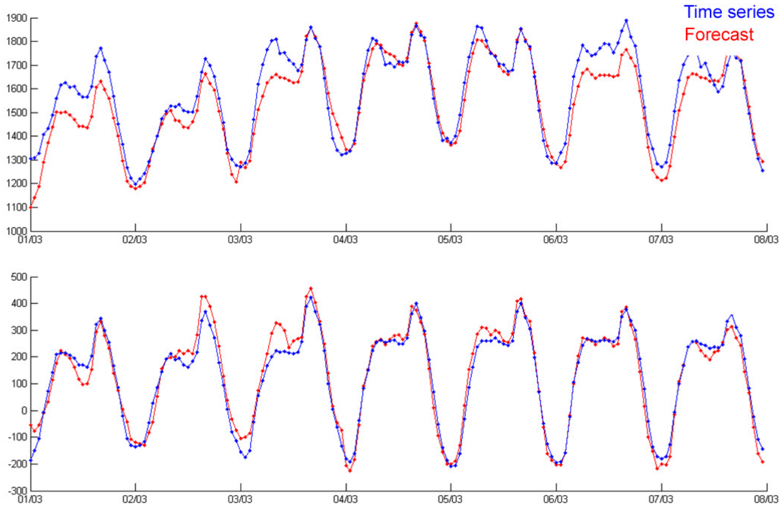
$$x_t(i,:) = \sum_{j=1}^{m} x_t(i,j), \quad i = 1,\dots n;$$

$$x_t(:,j) = \sum_{i=1}^{n} x_t(i,j), \quad j = 1,\dots m;$$

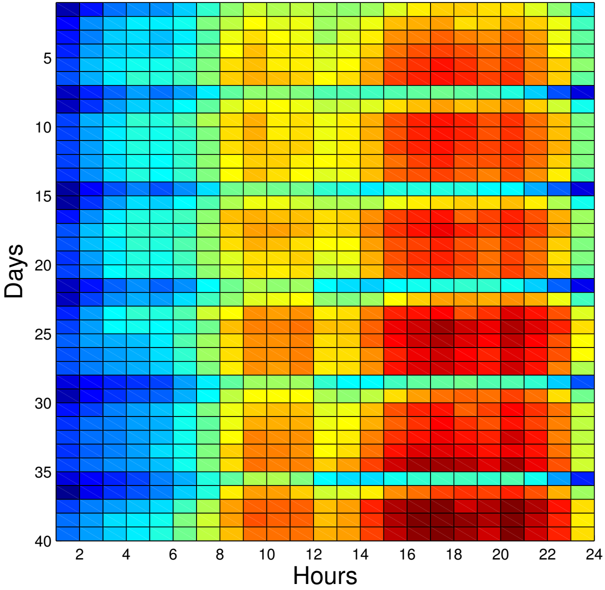
$$t = 1,\dots,T.$$

Detailed independent forecasts could no be concorded according to the time series hierarchical structure.

A brief introduction to auto-regressive forecasting



The autoregressive matrix, five week-ends



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

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:

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

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

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

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

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

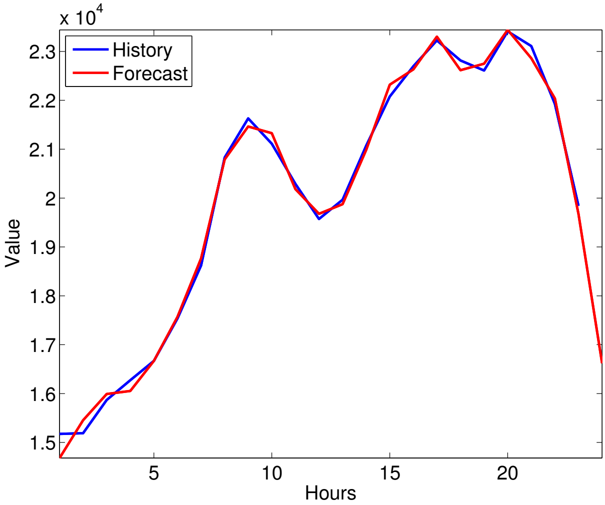
Kolmogorov-Gabor polynomial as a variant for model generation

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

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

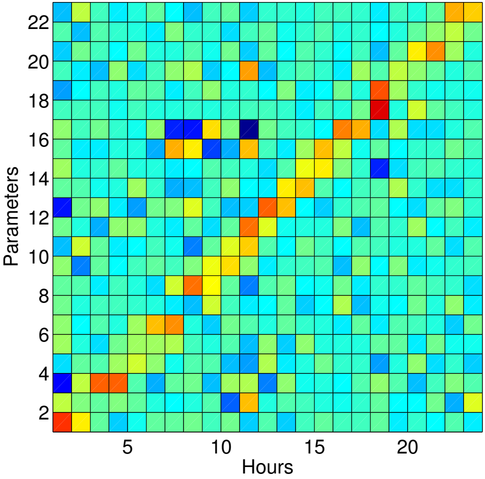
$$156 \times 168, \text{ in details: } 52\text{w} \cdot 3\text{y} \times 24\text{h} \cdot 7\text{d};$$

- 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.

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.

The initial model includes all independent variables

$$f(\mathbf{w},\mathbf{x}) = \alpha_1 w_1 x_1 + \alpha_2 w_2 x_2 + \ldots + \alpha_n w_n x_n.$$

The hyperparameter $\alpha \in \{0,1\}$ is included in the model. The **exhaustive search** procedure counts

$$\begin{array}{cccc} \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \hline 1 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 1 & 1 & \ldots & 1 \end{array}.$$

Add (append a feature)

Step 0. The active set $\mathcal{A}_0 = \emptyset$.

Step $k = 1, \ldots, n$. Select the next best feature index

$$\hat{j} = \arg \min_{j \in \{1,\ldots,n\} \setminus \mathcal{A}_k} \min_{\mathbf{w} \in \mathbb{R}^{|\mathcal{A}|}} \|[\mathbf{X}_{\mathcal{A}_k} \chi_j]\mathbf{w} - \mathbf{y}\|_2^2,$$

according to minimum of the error function $S(\mathbf{w})$; then

$$\mathcal{A}_{k+1} = \mathcal{A}_k \cup \hat{j}.$$

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:

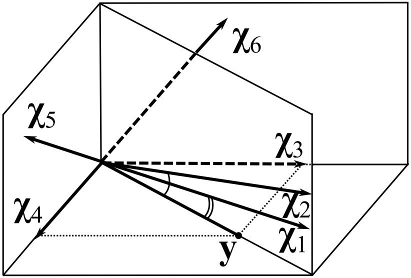
$$[a_{p,1}, \dots, a_{p,\nu}, a_{q,\nu+1}, \dots, a_{q,n}] \to \mathbf{a}'_p,$$

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

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.

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.



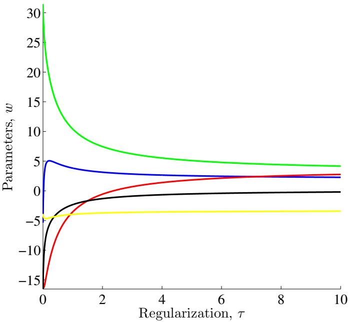
Stability and accuracy for a fixed complexity

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

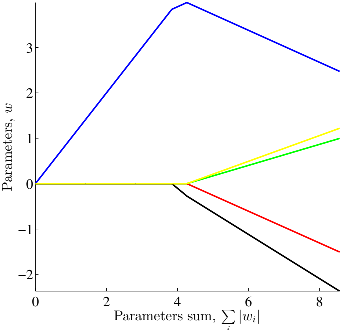
Algorithms: GMDH, Stepwise, Ridge, Lasso, Stagewise, FOS, LARS, Genetics, ...

Model parameter values with regularization

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



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



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