Time Series Forecasting.
3. Advanced TS Forecasting

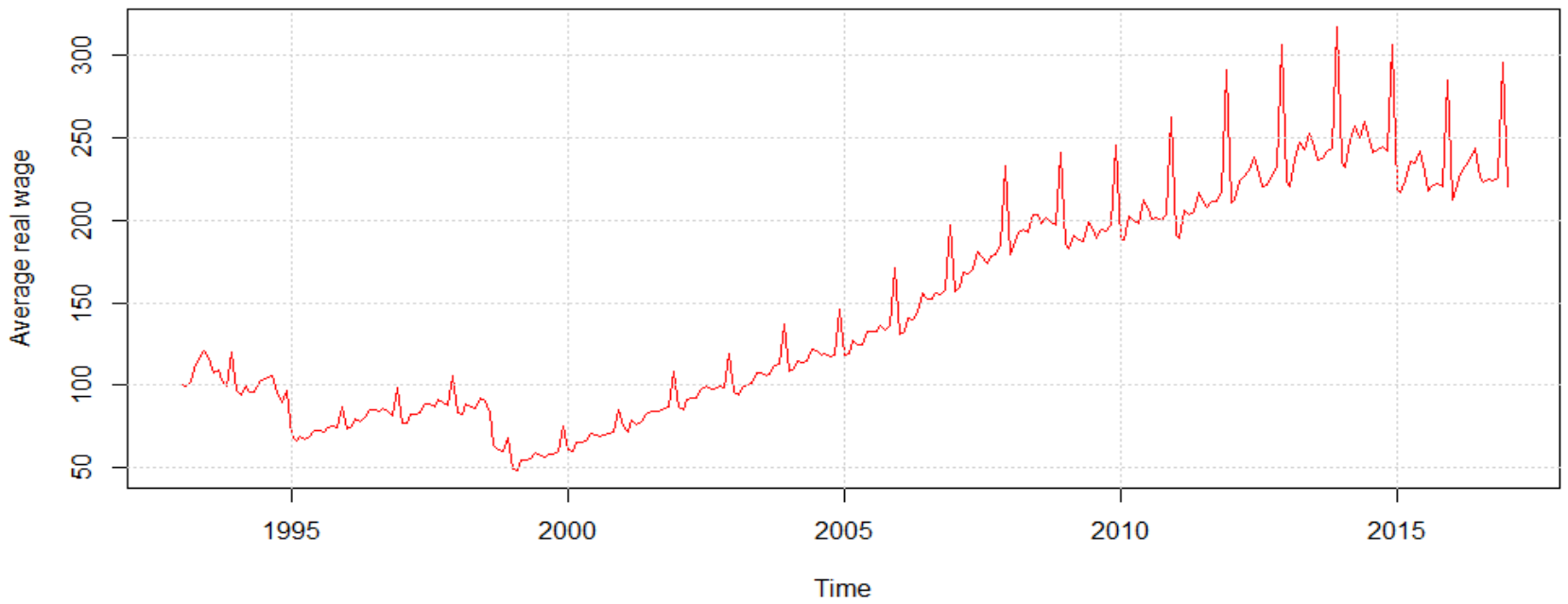
Alexey Romanenko alexromsput@gmail.com

FIVT MIPT, September 2018

Содержание

Time Series definition

Time series: $y_1, \ldots, y_T, \ldots, \quad y_t \in \mathbb{R}$, — a sequence of values of some variable, detected in a constant time interval.



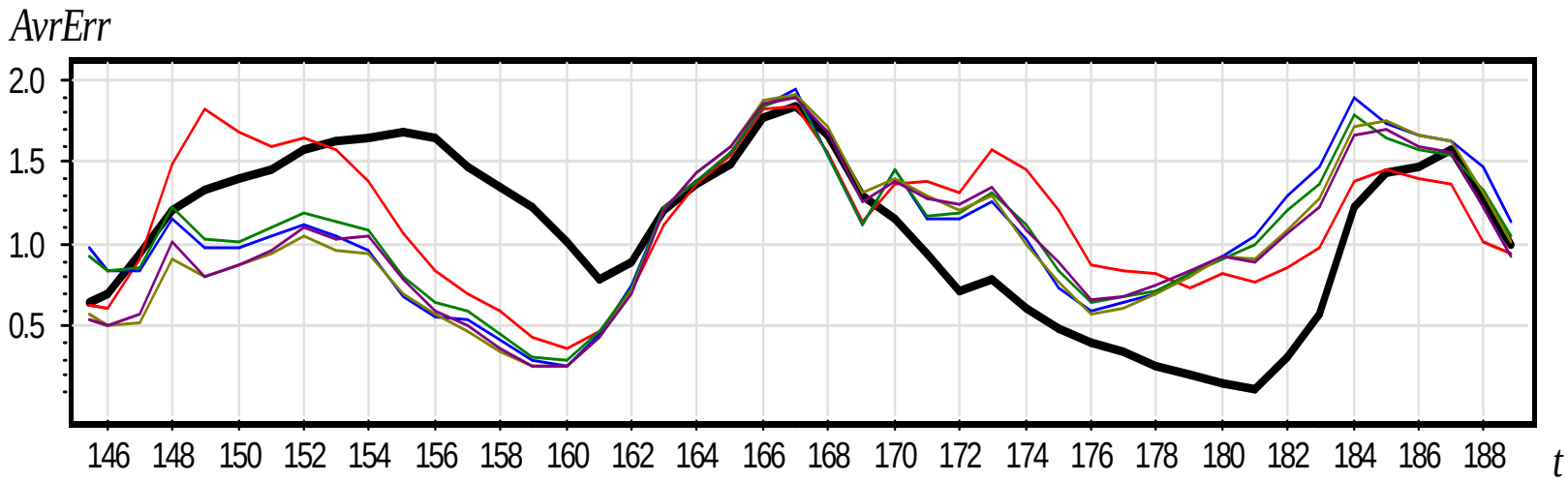
Time series forecasting task — find function f_T:

$$y_{T+d} \approx f_T\left(y_T, \ldots, y_1, d\right) \equiv \hat{y}_{T+d|T},$$

where $d \in \{1, \ldots, D\}$ — delay, D — horizon.

General Idea of Compositions

Dynamics of loss function for 6 TS forecasting algorithms:



Idea: use successful base algorithms and don't use less successful.

Adaptive Selection

There is M base algorithms $A_1, \ldots, A_M$,
$\hat{y}^j_{t+d}$ — forecast of A_j for the moment $t+d$,
$e^j_t = y_t - \hat{y}^j_t$ — error of A_j at the moment t,
$\tilde{e}^j_t = \delta \sum_{l=1}^{t} (1-\delta)^{t-l} |e^j_l|$ — exponentially weighted absolute error,
δ — smoothing parameter.

The best base algorithm in the moment t:

$$j^*_t = \underset{j=1,\ldots,M}{\operatorname{argmin}} \tilde{e}^j_t.$$

Best indistinctive algorithms:

$$\mathfrak{A}^*_t(\varepsilon) = \left\{ A_i \in \mathfrak{A} | \tilde{e}^i_t \leq \tilde{e}^{j^*_t}_t + \varepsilon \right\}.$$

Adaptive Selection (composition):

$$\hat{y}^C_{t+d} := \frac{1}{|\mathfrak{A}^*_t(\varepsilon)|} \sum_{A_i \in \mathfrak{A}^*_t(\varepsilon)} \hat{y}^i_{t+d}.$$

Adaptive combination

There is M base algorithms $A_1, \ldots, A_M$,
$\hat{y}^j_{t+d}$ — forecast of A_j for the moment $t+d$,
$e^j_t = y_t - \hat{y}^j_t$ — error of A_j at the moment t,
$\tilde{e}^j_t = \delta \sum\limits_{l=1}^{t} (1-\delta)^{t-l} |e^j_l|$ — exponentially weighted absolute error,
δ — smoothing parameter.

Adaptive combination:

$$\hat{y}^C_{t+d} = \sum_{j=1}^{M} w^j_t \hat{y}^j_{t+d}, \qquad \sum_{j=1}^{M} w^j_t = 1, \quad \forall t.$$

Adaptive weights:

$$w^j_t = \frac{(\tilde{e}^j_t)^{-1}}{\sum\limits_{s=1}^{M} (\tilde{e}^s_t)^{-1}}.$$

Other Examples of Compositions

Other approaches:

- exponentially weighted squared errors;
- moving averaged squared/absolute errors;
- LSE of weights with regularization;
- ...

Well-known Compositions:

- AFTER (Aggregated Forecast Through Exponential Reweighing) [Yang Y., 2004];
- Averaging according to Inverse Weights , [Timmermann A.G., 2006];
- LAWR (locally adaptive weights with regularization), [Vorontsov K.V., 2006];
- Adaptive selection [Лукашин Ю.П., 2001].
- QR (Quantile Regression)

Loss is more important than forecast

Binary squared game $\Omega = \{0, 1\}$, $\Gamma = [0, 1]$, $\lambda = (\omega - \gamma)^2$;

Total loss (loss process):

$$\text{Loss}_A(T) = \sum_{t=1}^{T} \lambda(y_t, \hat{x}_t^A)$$

.

- Task 1
  - base algorithm 1 builds constant forecast 0;
  - how can we build forecast of composition AA such that

$$\text{Loss}_{AA} \leq \frac{1}{2}\text{Loss}_1?$$

  - *Answer: ???*

Loss is more important than forecast

Binary squared game $\Omega = \{0, 1\}$, $\Gamma = [0, 1]$, $\lambda = (\omega - \gamma)^2$;

Total loss (loss process):

$$\text{Loss}_A(T) = \sum_{t=1}^{T} \lambda(y_t, \hat{x}_t^A)$$

.

- Task 1
  - base algorithm 1 builds constant forecast 0;
  - how can we build forecast of composition AA such that
$$\text{Loss}_{AA} \leq \frac{1}{2}\text{Loss}_1?$$
  - *Answer: ???*
- Task 2
  - base algorithm 1 gets an average penalty $\frac{1}{2}$
  - how can we build forecast of composition AA such that
$$\text{Loss}_{AA} \leq \frac{1}{2}\text{Loss}_1?$$
  - *Answer: ???*

Loss is more important than forecast

Binary squared game $\Omega = \{0, 1\}$, $\Gamma = [0, 1]$, $\lambda = (\omega - \gamma)^2$;

Total loss (loss process):

$$\text{Loss}_A(T) = \sum_{t=1}^{T} \lambda(y_t, \hat{x}_t^A)$$

.

- Task 1
  - base algorithm 1 builds constant forecast 0;
  - how can we build forecast of composition AA such that
$$\text{Loss}_{AA} \leq \frac{1}{2}\text{Loss}_1?$$
  - *Answer: ???*
- Task 2
  - base algorithm 1 gets an average penalty $\frac{1}{2}$
  - how can we build forecast of composition AA such that
$$\text{Loss}_{AA} \leq \frac{1}{2}\text{Loss}_1?$$
  - *Answer: we build a constant forecast* $\frac{1}{2}$

Loss is more important than forecast

Binary squared game $\Omega = \{0, 1\}$, $\Gamma = [0, 1]$, $\lambda = (\omega - \gamma)^2$;

Total loss (loss process):

$$\text{Loss}_A(T) = \sum_{t=1}^{T} \lambda(y_t, \hat{x}_t^A)$$

.

- Task 1
  - base algorithm 1 builds constant forecast 0;
  - how can we build forecast of composition AA such that
$$\text{Loss}_{AA} \leq \frac{1}{2}\text{Loss}_1?$$
  - *Answer: ???*
- Task 2
  - base algorithm 1 gets an average penalty $\frac{1}{2}$
  - how can we build forecast of composition AA such that
$$\text{Loss}_{AA} \leq \frac{1}{2}\text{Loss}_1?$$
  - *Answer: we build a constant forecast* $\frac{1}{2}$

Conclusion: it is more important to look at losses rather than at the forecast itself

Kolmogorov Mean as an Aggregation of Arithmetic Mean

Kolmogorov Mean:

$$M(y_1, \ldots, y_n) = \varphi^{-1}\left(\frac{1}{n}\sum_{k=1}^{n}\varphi(y_k)\right) = \varphi^{-1}\left(\frac{\varphi(y_1) + \ldots + \varphi(y_n)}{n}\right)$$

- $\varphi(x) = x \Rightarrow M(y_1, \ldots, y_n) = \frac{y_1 + \cdots + y_n}{n}$ — arithmetic mean;
- $\varphi(x) = x^{-1} \Rightarrow M(y_1, \ldots, y_n) = \frac{n}{1/y_1 + \cdots + 1/y_n}$ — harmonic mean;
- $\varphi(x) = \log(x) \Rightarrow M(y_1, \ldots, y_n) = \sqrt[n]{y_1 \cdot \cdots \cdot y_n}$ — geometric mean;
- $\varphi(x) = e^x \Rightarrow \ln\left(\frac{1}{n}\sum_{k=1}^{n} e^{(y_k)}\right)$

What aggregation (mixability) function should we choose in order to build forecasts?

The Idea of V. Vovk Aggregating Algorithm

- "average"(aggregate) losses instead of forecasts;
- weigh losses in exponential space $p_j \sim \exp^{-\eta \text{Loss}_j(T)}$;

Final composition AA is built based on generalized mixability function:

$$g(y) = \log_\beta \left(\sum_{j=1}^{N} \frac{1}{N} \beta^{\text{Loss}_j(T) + \lambda(y, \hat{y}_{j,T+1})} \right)$$

where $\beta = e^{-\eta} \in (0,1)$, $\eta \in (0, \infty)$ — learning rate

Compositions based on Aggregating Algorithm

Forecasts AA_1 и AA_2

Initialization of weights $p_{j,0} = 1/N$

For $t = 0, \ldots, T-1$

1. obtain prediction of experts $\hat{y}_{j,t+1}, \forall j = \overline{1, N}$;
2. calculate mixability function:

$$g(x) = \log_\beta \left(\sum_{j=1}^{N} p_{j,t} \cdot \beta^{\lambda(y, \hat{y}_{j,t+1})} \right)$$

3. $\hat{y}_{AA_1,t+1} = \frac{Y_2\sqrt{g(Y_1)} + Y_1\sqrt{g(Y_2)}}{\sqrt{g(Y_1)} + \sqrt{g(Y_2)}}$; $\hat{y}_{AA_2,t+1} = \frac{g(Y_1) - g(Y_2)}{2(Y_2 - Y_1)} + \frac{Y_1 + Y_2}{2}$;
4. obtain actual value y_{t+1}; calculate loss $\lambda(y_{t+1}, \hat{y}_{t+1})$;
5. update weights of experts $p_{j,t+1} = \beta^{\lambda(y_{t+1}, \hat{y}_j, t+1)} \cdot p_{j,t}$.

Loss Process Estimation

- Consider base forecast algorithms $\{A^1, \ldots, A^N\}$.
- Assign $p_0^j = 1/N$ where $j = \overline{1, N}$.
- Get appropriate β and $S(g)$
- We obtain a composition AA.
- Time complexity of the composition is $O(NT)$.

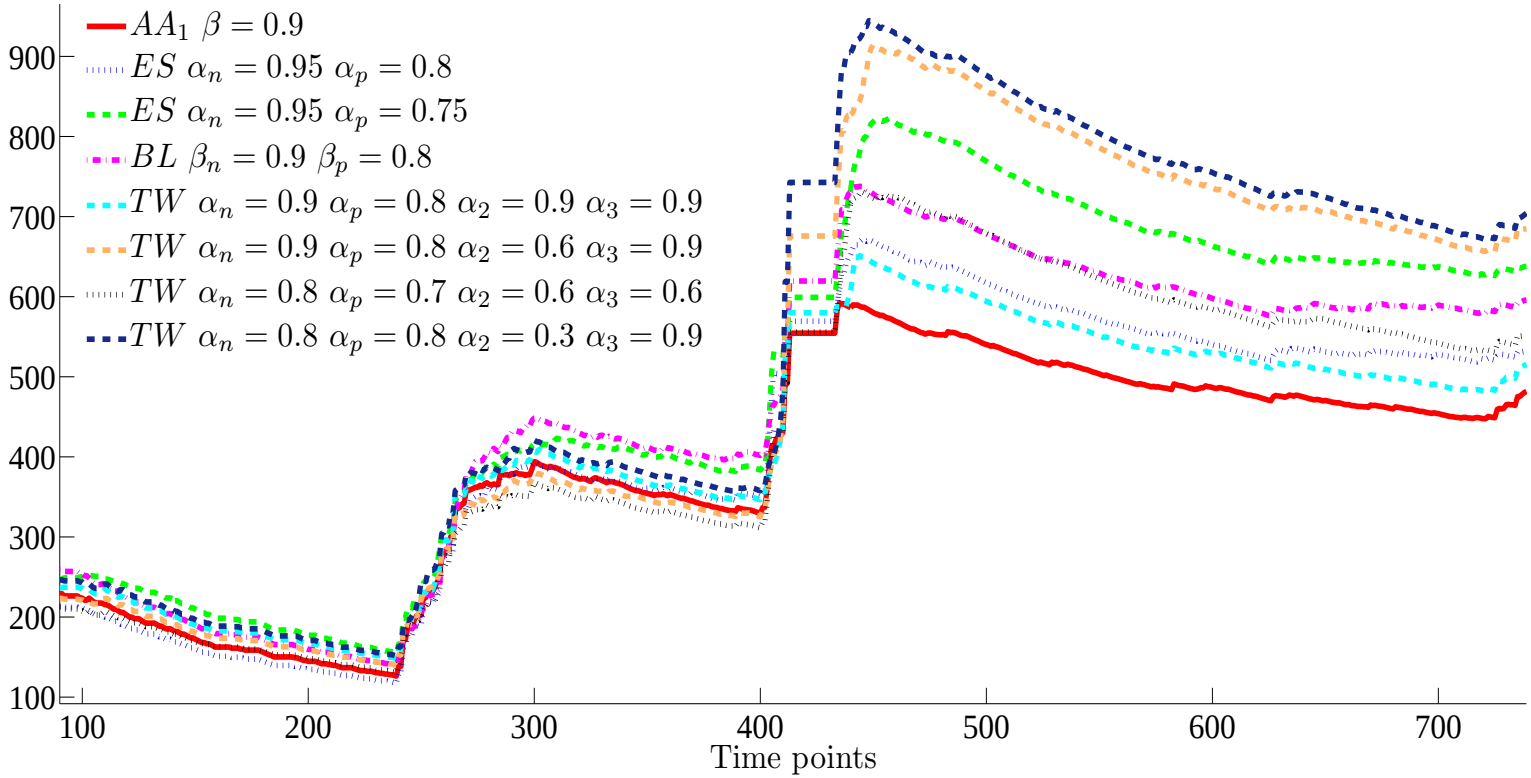
Theorem

The loss process AA in a game with SSE as loss process for $\forall(y_1, \ldots, y_T) \in [Y_1, Y_2]^T$, $\forall\{A^1, \ldots, A^M\}$ satisfies inequality:

$$\text{Loss}_{AA}(T) \leq \min_{i=1,\ldots,M} \text{Loss}_{A^i}(T) + O\left(\ln(N)\right). \tag{1}$$

Comparison with Base Algorithms Example 1

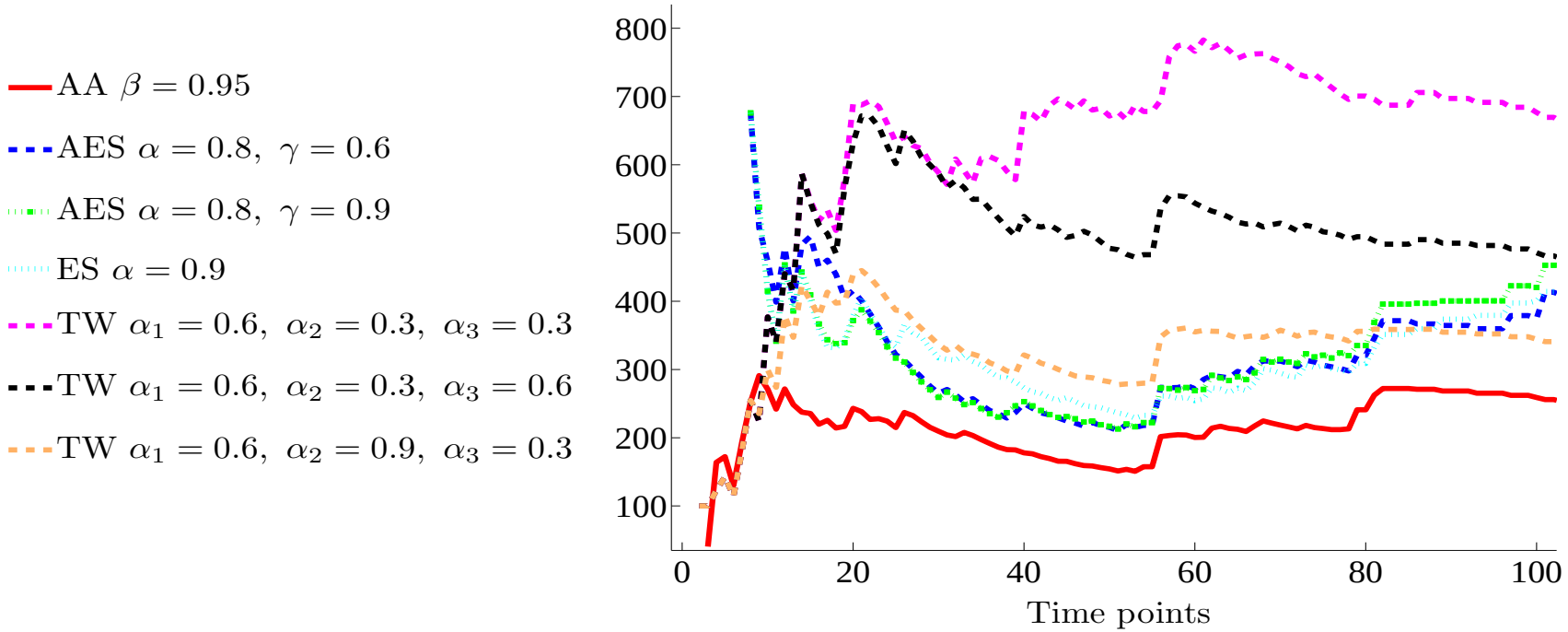
An experiment with real data (1 of 1000 time series)



$$\text{MSE} = \frac{1}{T}\text{Loss}(T)$$

Comparison with Base Algorithms Example 1

An experiment with real data (1 of 1000 time series)



$$\text{MSE} = \frac{1}{T}\text{Loss}(T)$$

Comparison with Other Compositions

Таблица: Comparison of compositions under a symmetric loss function, MSE

M	AFTER	IW	LAWR	BI	AA_1	AA_2
10	6,57	6,66	6,74	6,75	6,43	6,37
25	6,50	6,62	6,92	6,71	6,39	6,31
40	6,55	6,57	6,90	6,66	6,35	6,37
	100%	100%	105%	103%	95%	97%

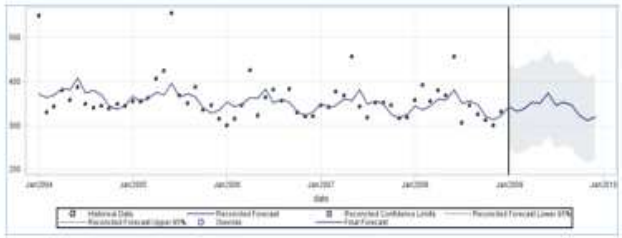
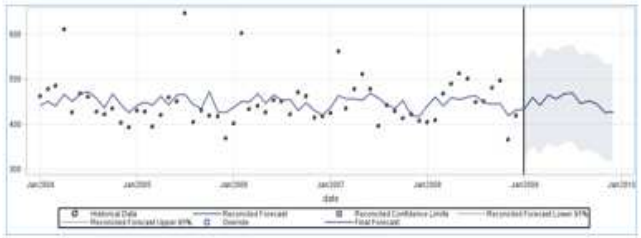
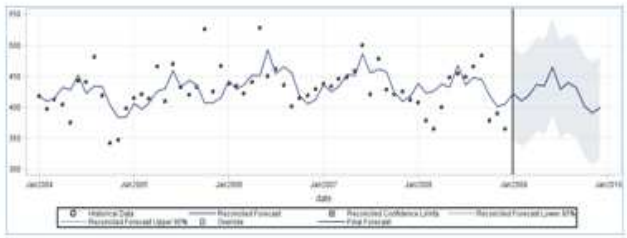
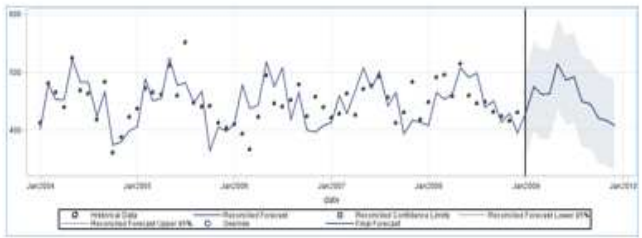
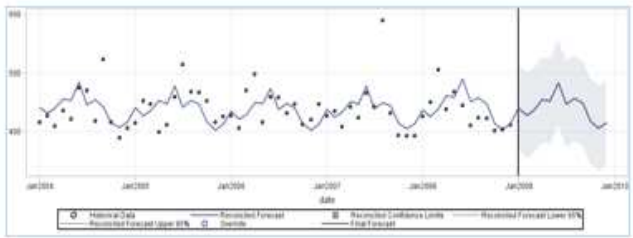
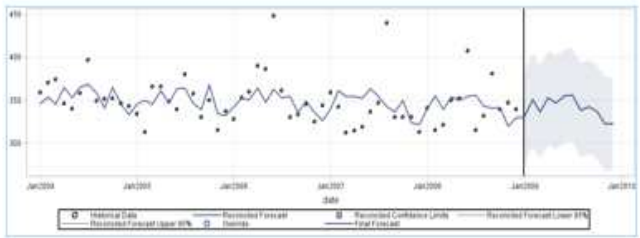
Таблица: Comparison of compositions under an asymmetric loss function

k_1/k_2	AA_1	AA_2	QR
2	**2344**	2375	2804
10	**2694**	2863	4978
100	**7700**	8605	12223

Hierarchy in Retail

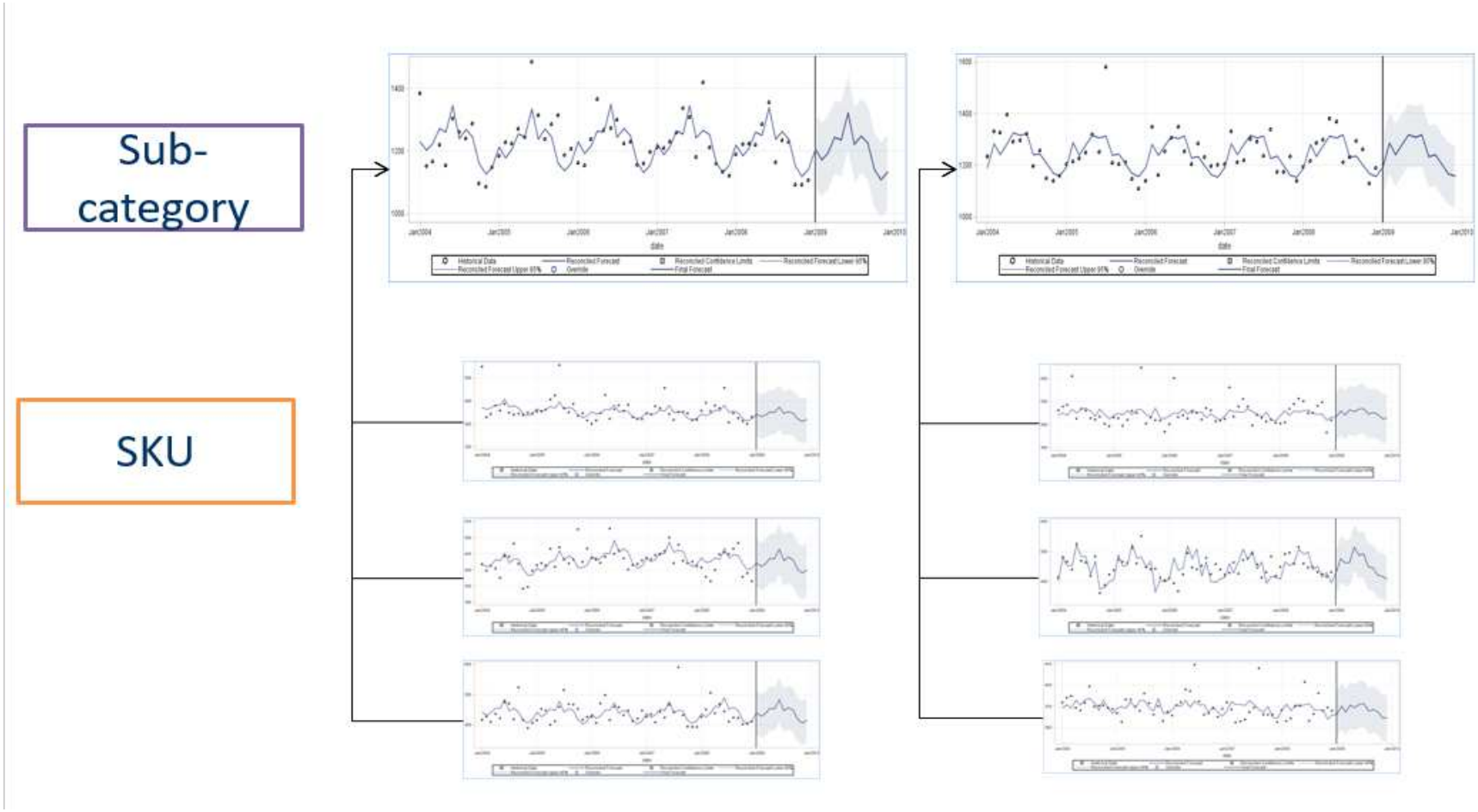
Example of hierarchy

SKU

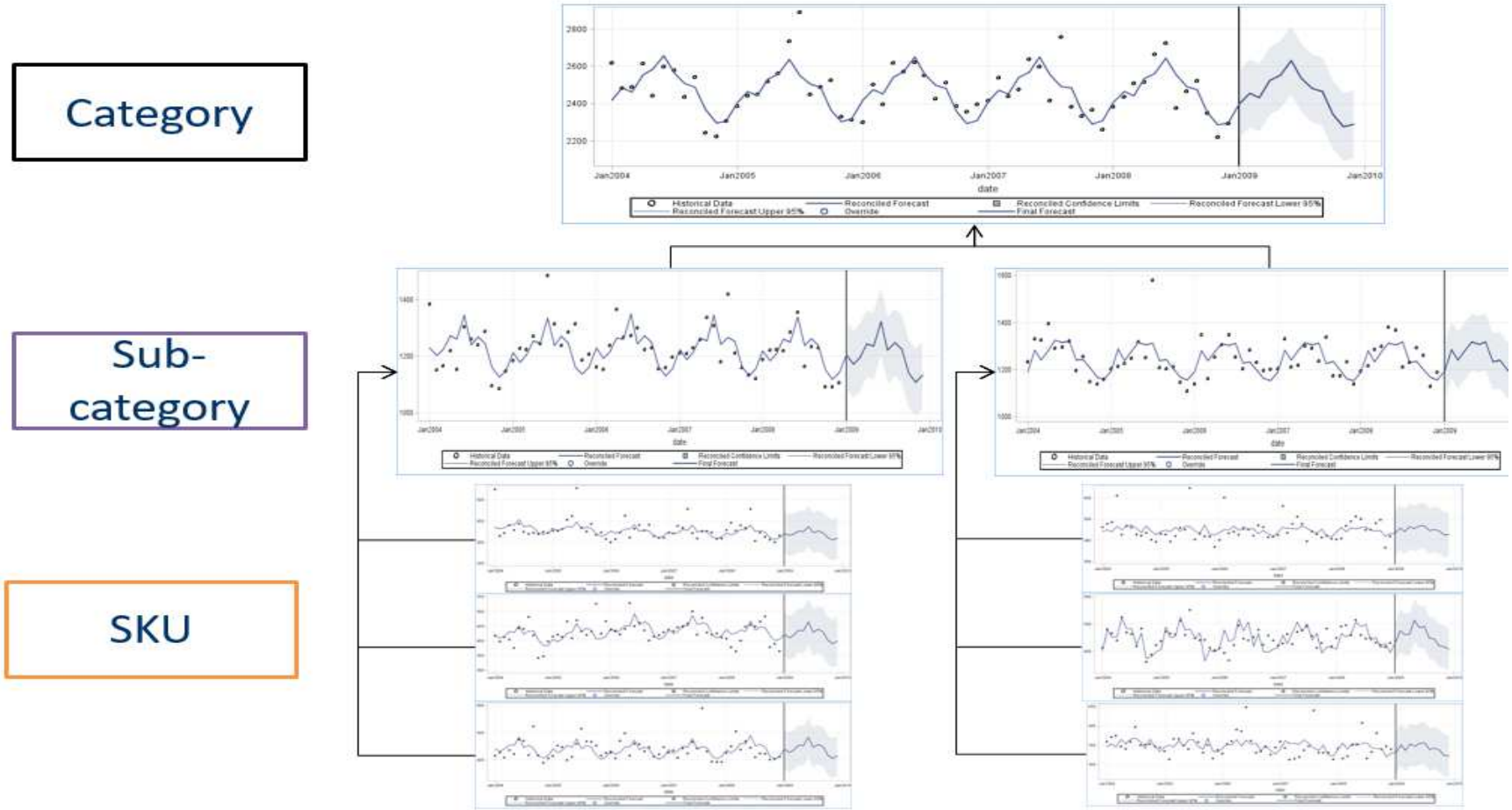
Hierarchy in Retail

Example of hierarchy



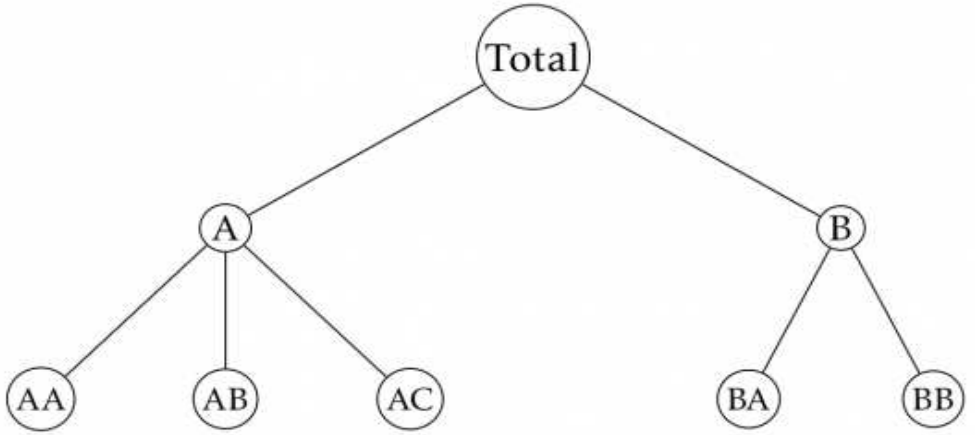
Hierarchy in Retail

Example of hierarchy



Hierarchy of TS in Retail

Часто необходимо прогнозировать совокупности временных рядов иерархической структуры. Например, продажи могут группироваться по товарным группам, складам, поставщикам и т. д.



$$\begin{pmatrix} y_t \\ y_{A,t} \\ y_{B,t} \\ y_{AA,t} \\ y_{AB,t} \\ y_{AC,t} \\ y_{BA,t} \\ y_{BB,t} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} y_{AA,t} \\ y_{AB,t} \\ y_{AC,t} \\ y_{BA,t} \\ y_{BB,t} \end{pmatrix}, \qquad \mathbf{y}_t = S\mathbf{y}_{K,t}.$$

Подходы к реконсиляции прогнозов

Снизу вверх: прогнозы рядов более высоких уровней иерархии получаются суммированием прогнозов нижнего уровня.

- информация не теряется из-за агрегирования, но
- прогнозировать ряды нижнего уровня часто сложнее.

Сверху вниз: прогноз суммарного ряда y_t распределяется согласно средним долям:

$$p_j = \frac{1}{T} \sum_{t=1}^{T} \frac{y_{j,t}}{y_t}$$

или долям средних:

$$p_j = \sum_{t=1}^{T} \frac{y_{j,t}}{T} \Big/ \sum_{t=1}^{T} \frac{y_t}{T}.$$

- прогнозировать суммированный ряд легко, но
- из-за агрегирования теряется информация (например, если компоненты имеют разную сезонность).

Подходы к реконсиляции прогнозов

Оптимальная комбинация: ряд каждого уровня прогнозируется отдельно, затем прогнозы корректируются в сторону большей согласованности с помощью регрессии

$$\hat{\mathbf{y}}_h = S\beta_h + \varepsilon_h, \quad \mathbb{E}\varepsilon_h = 0, \quad \operatorname{cov}\varepsilon = \Sigma_h;$$

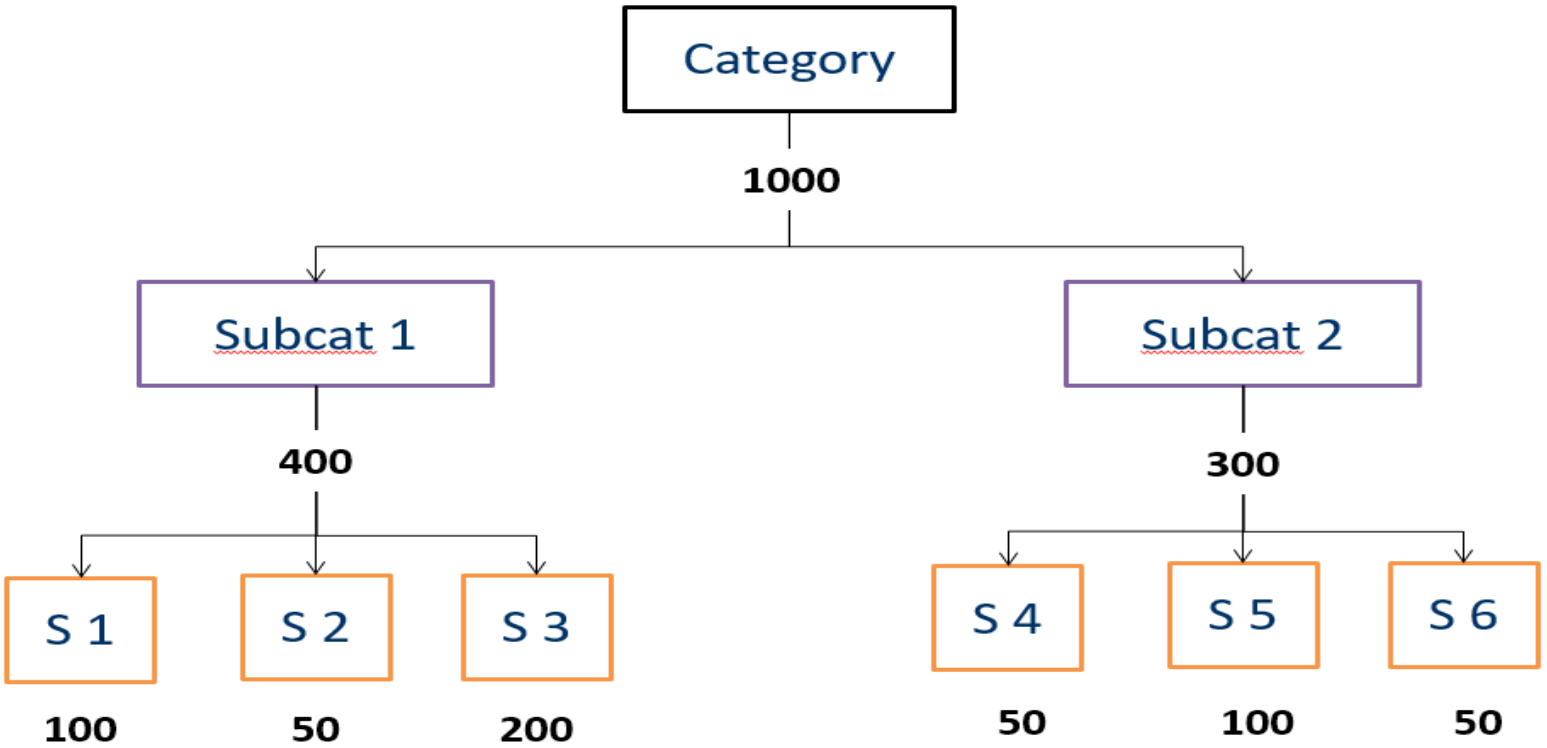
$$\varepsilon_h \approx S\varepsilon_{K,h} \Rightarrow \tilde{\mathbf{y}}_h = S\left(S^T S\right)^{-1}\hat{\mathbf{y}}_h.$$

Метод реализован в пакете hts.

Метод с теоретическими гарантиями (Стенина, Стрижов, 2015): если суммарные потери при прогнозировании всех рядов иерархии измеряются с помощью функции из класса дивергенций Брегмана, проецирование вектора прогнозов на множество векторов, удовлетворяющих структуре иерархии, не увеличивает суммарные потери.

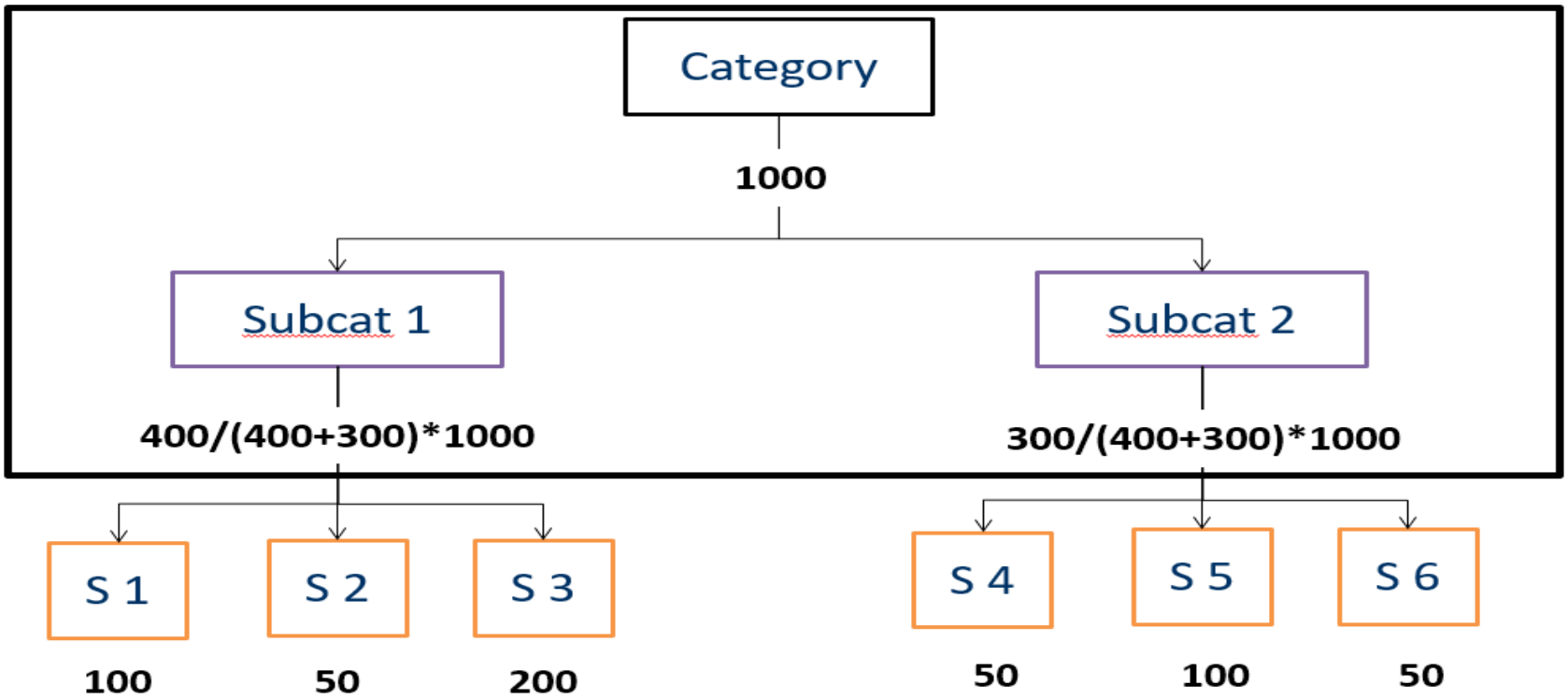
Example of forecast reconciliation

Top-down reconciliation



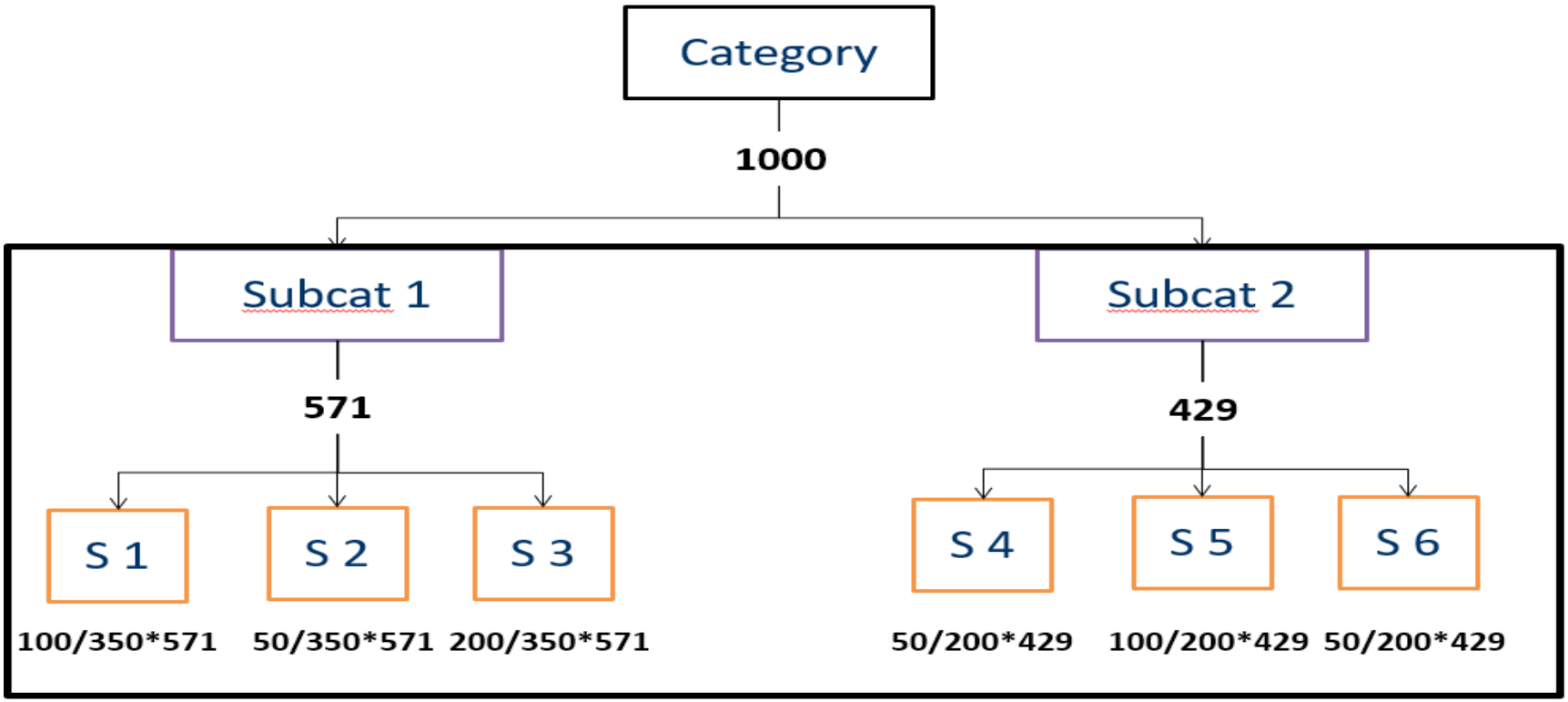
Example of forecast reconciliation

Top-down reconciliation



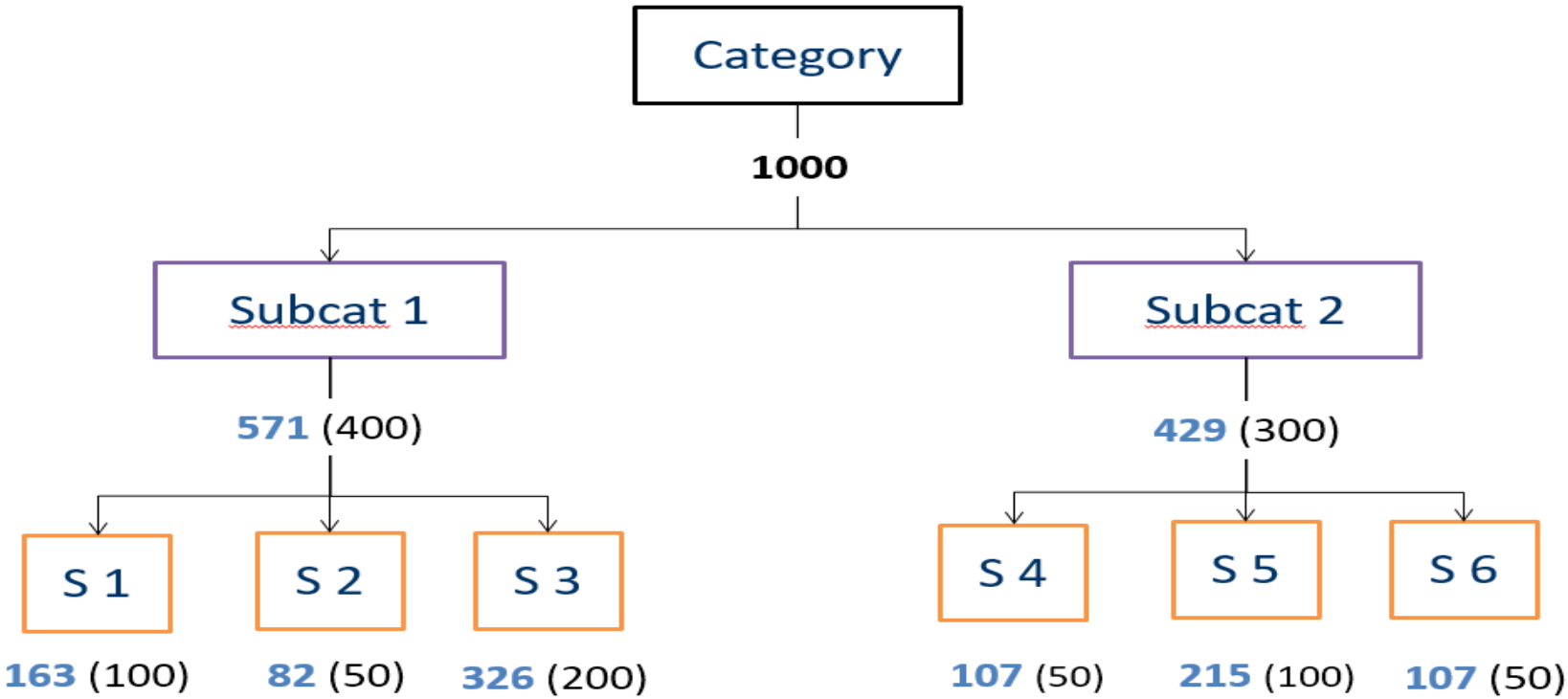
Example of forecast reconciliation

Top-down reconciliation



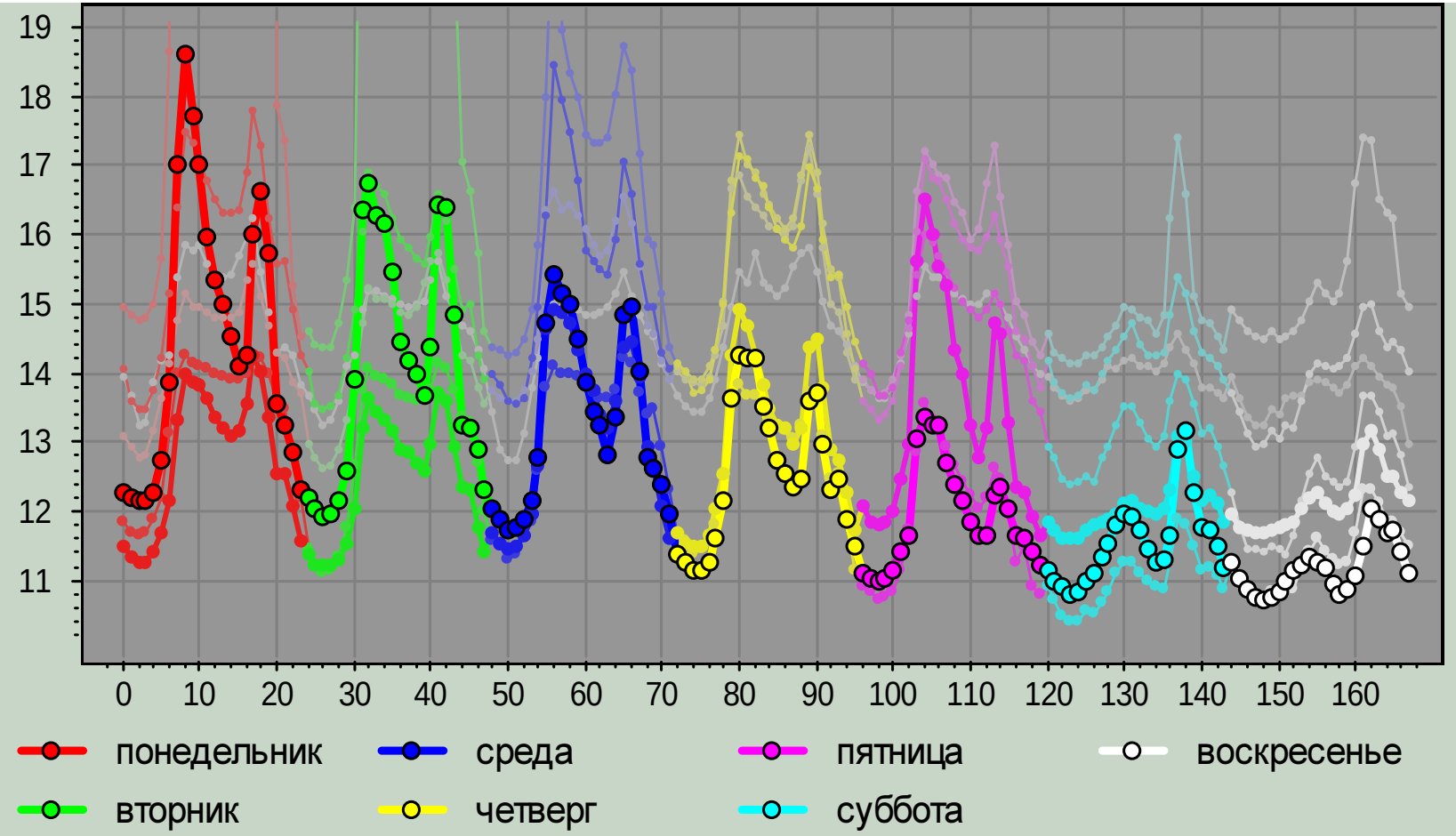
Example of forecast reconciliation

Top-down reconciliation



Forecasting of Energy Consumption

Energy Consumption in NordPool, 2000:



Linear Autoregression for Energy Consumption

Regressors (features) — n previous points of time series:

$$\hat{y}_{t+1}(\alpha) = \sum_{j=1}^{n} \alpha_j y_{t-j+1}, \quad \alpha \in \mathbb{R}^n$$

Samples are $\ell = t - n + 1$ moments of time series:

$$\underset{\ell \times n}{F} = \begin{pmatrix} y_t & y_{t-1} & y_{t-2} & \cdots & y_{t-n+1} \\ y_{t-1} & y_{t-2} & y_{t-3} & \cdots & y_{t-n} \\ y_{t-2} & y_{t-3} & y_{t-4} & \cdots & y_{t-n-1} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ y_n & y_{n-1} & y_{n-2} & \cdots & y_1 \end{pmatrix}, \quad \underset{\ell \times 1}{y} = \begin{pmatrix} y_{t+1} \\ y_t \\ y_{t-1} \\ \cdots \\ y_{n+1} \end{pmatrix}$$

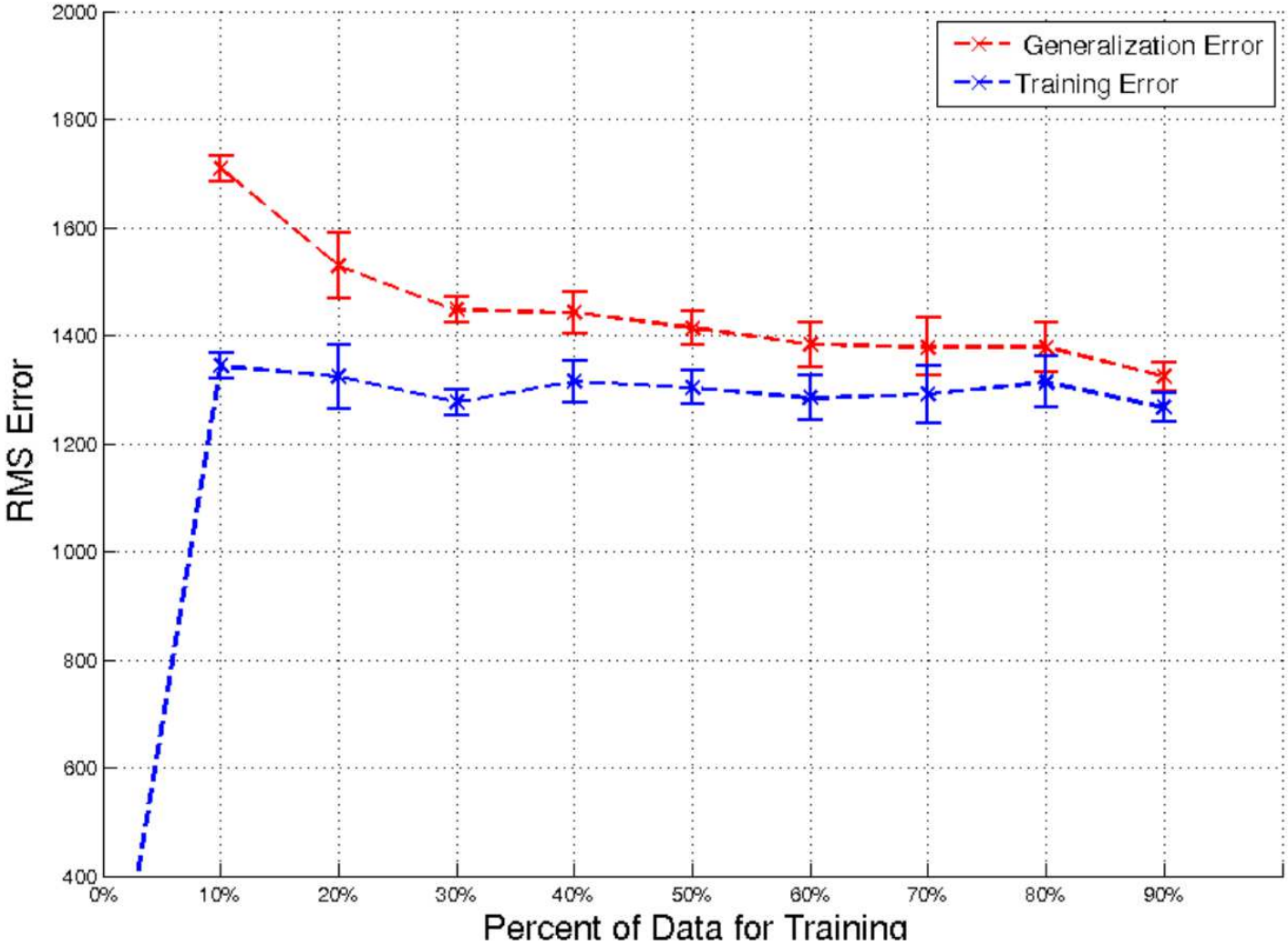
Loss Functional:

$$Q_t(\alpha, X^\ell) = \sum_{i=n+1}^{t+1} \big(\hat{y}_i(\alpha) - y_i\big)^2 = \|Fw - y\|^2 \to \min_\alpha$$

See example of LR for TS forecasting in `1_intro.ipnb`

When is some complicated model needed?

The more data in train set the better forecast (in test set)



Accuracy of Energy Consumption Forecasting

Energy Consumption Forecasting:

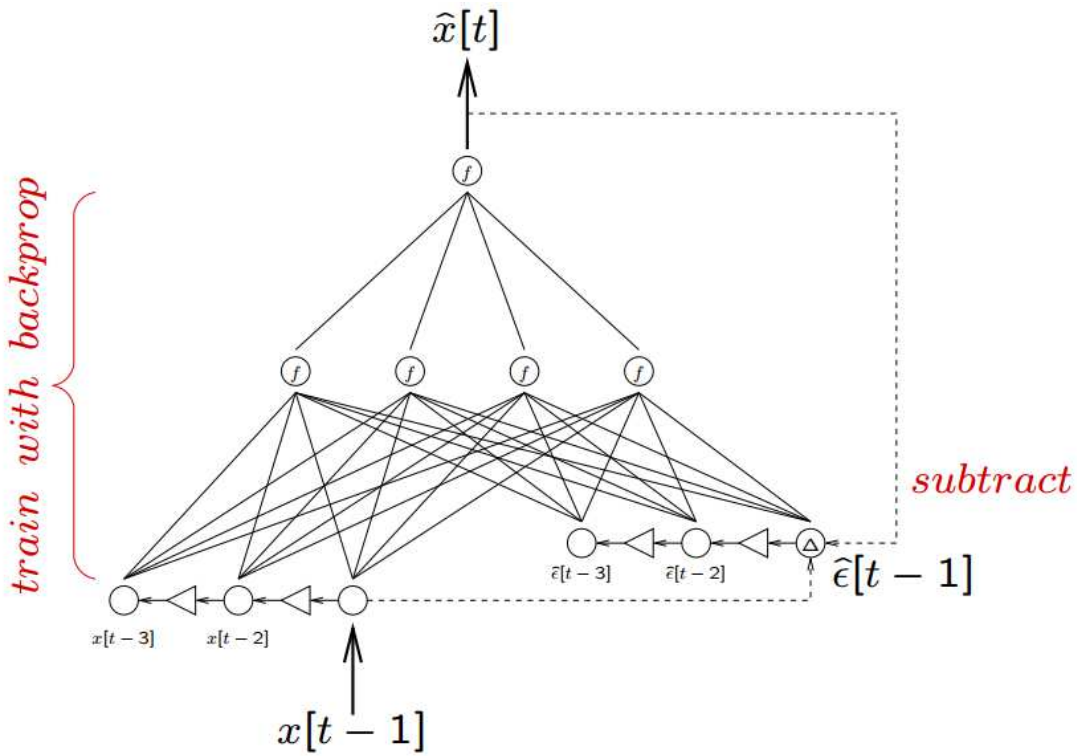
Таблица: Accuracy of forecast for different models

Learning method	RMSE	% RMSE
Kernelized Regression	1 540	8.3%
NN	1250	6.7%
Deep Forward NN	1130	5.9%
Deep Recurent NN	**530**	2.8%

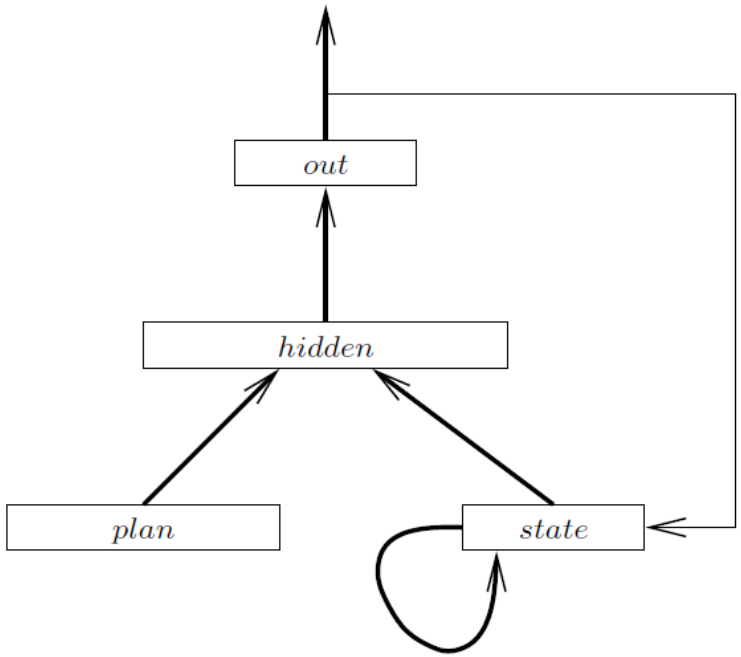
Enzo Busetti etc. Deep Learning for Time Series Modeling. CS 229 final Project Report, 2012.

Non-linear ARMA with NN

ARMA: $$x_t = \underbrace{c + \sum_{i=1}^{p} \alpha_i x_{t-i}}_{AR} + \underbrace{\sum_{j=1}^{q} \beta_j \varepsilon_{t-j}}_{MA} + \varepsilon_t;$$



Memory term for NN



Memory term: $\overline{x}_i(t) = \sum_{\tau=1}^{t} c_{t-\tau} \cdot x_\tau$
Weights for memory terms:

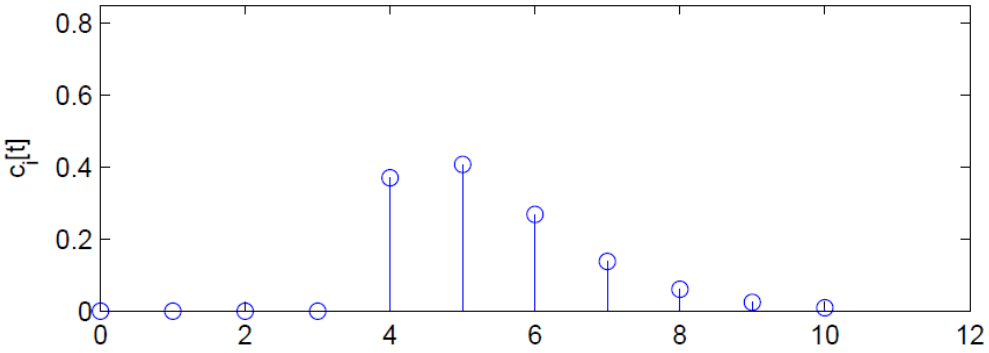
- delta-function $c_t = \delta_d(t)$
- exponential weights: $c_t = (1-\alpha)\alpha^t$

Weights of memory term

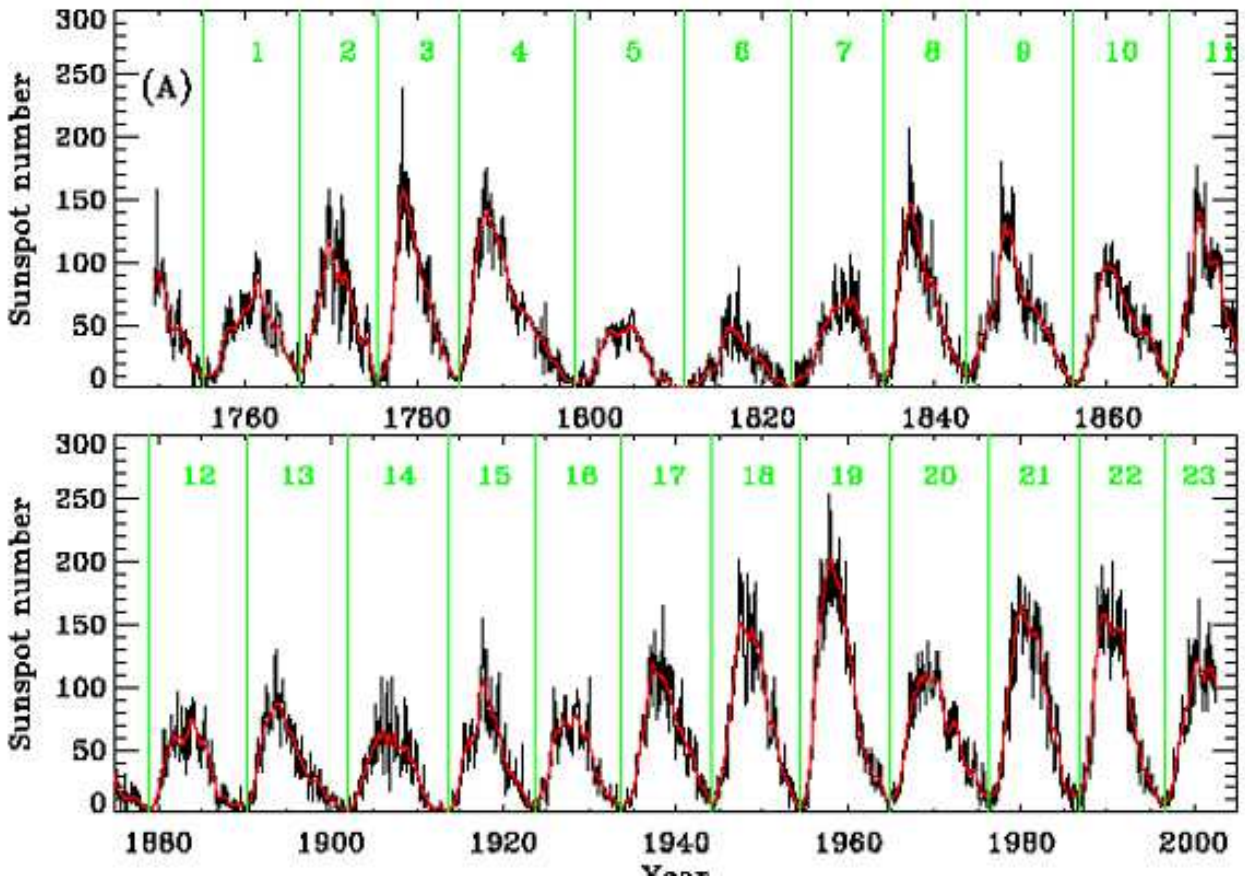
- $c_t = \begin{cases} \binom{t}{d}(1-\alpha)^{d+1} \cdot \alpha^{t-d} & \text{если } t \geq d \\ 0, & \text{иначе} \end{cases}$

  by $d = 0$ we obtain exponential weights;
  by $\alpha \to 0$ we obtain $\delta_d(t)$

For example $d = 4, \alpha = 0.2$



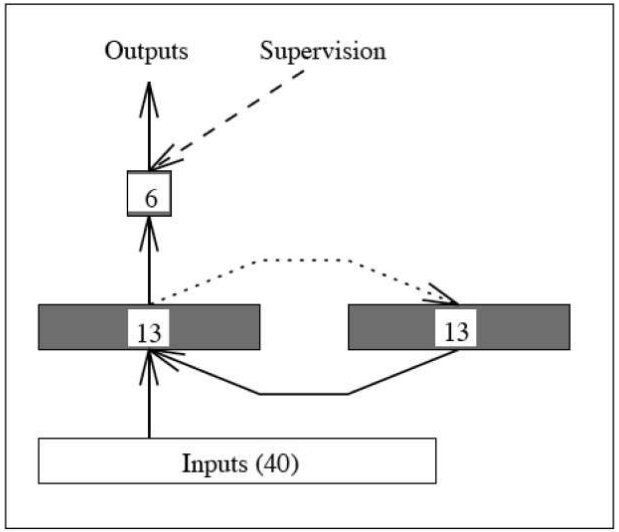
Prediction of sunspots



- 1 time series
- cycle: minimums are observed each 9–14 years;
- there are a lot of cycles (since 1849).

Elman NET

Output: $\{\widehat{x}[t], \cdots, \widehat{x}[t+5]\}$



Input: $\{x[t-40], \cdots, x[t-1]\}$

Learning method	CNET heuristic	Simple NN	Modular NN (simple CNN)	Elman Net
ARV	0.1130	0.0884	0.0748	0.0737
No Strong Errors	12	12	4	4

Conclusion

1. other well-known methods of TS forecasting can be explained in terms of ARMA models;
2. DeepNN can be interpreted as generalization of ARIMA model
3. complicated algorithms (regressions, RNN) are useful for time series of complicated structure (several types of seasonality, high modality of time series);
4. a large history is required to train complicated algorithms.
5. there are some specific approaches (hierarchy forecasting)
6. Aggregating Algorithm for compositions:
   1. is based on loss process mixing rather forecasts
   2. it is possible to build theoretical assessment
   3. compositions based on the aggregating algorithm are adaptive and not time-consuming
   4. outperforms base algorithms

Conclusion

Отзывы о лекции: https://goo.gl/forms/TpY4aaojXLszGPQy2

Literature

- Hyndman R.J., Athanasopoulos G. *Forecasting: principles and practice*, 2016. `https://www.otexts.org/book/fpp`
- Sullivan R., Timmermann A., White H. (2003). *Forecast evaluation with shared data sets*. International Journal of Forecasting, 19(2), 217–227.
- V.Vovk and C.J.H.C.Watkins. Universal Portfolio Selection. In *Proceedings of the 11th Annual Conference on Computional Learning Theory*, pages 12-23, 1998.
- V.Vovk. Competitive on-line statistics. *International Statistic Review*, 69(2):213-248, 2001.
- A. A. Romanenko. Aggregation of Adaptive Forecasting Algorithms Under Asymmetric Loss Function Lecture Notes in Computer Science, LNCS 9047, Springer International Publishing, 2015. — pp. 137–146.
- В. В Вьюгин. МАТЕМАТИЧЕСКИЕ ОСНОВЫ ТЕОРИИ МАШИННОГО ОБУЧЕНИЯ И ПРОГНОЗИРОВАНИЯ, `http://iitp.ru/upload/publications/6256/vyugin1.pdf`