Time Series Forecasting.
4. Accuracy of Forecasting. Advanced TS Forecasting

Alexey Romanenko alexromsput@gmail.com

Acknowledgement: Evgeny Riabenko for materials supplied

Содержание

Loss Functions of dotted forecasts

Mean squared error:

$$MSE = \frac{1}{T-R+1}\sum_{t=R}^{T}\left(\hat{y}_t - y_t\right)^2.$$

Mean absolute error:

$$MAE = \frac{1}{T-R+1}\sum_{t=R}^{T}\left|\hat{y}_t - y_t\right|.$$

Mean absolute percentage error:

$$MAPE = \frac{100}{T-R+1}\sum_{t=R}^{T}\left|\frac{\hat{y}_t - y_t}{y_t}\right|.$$

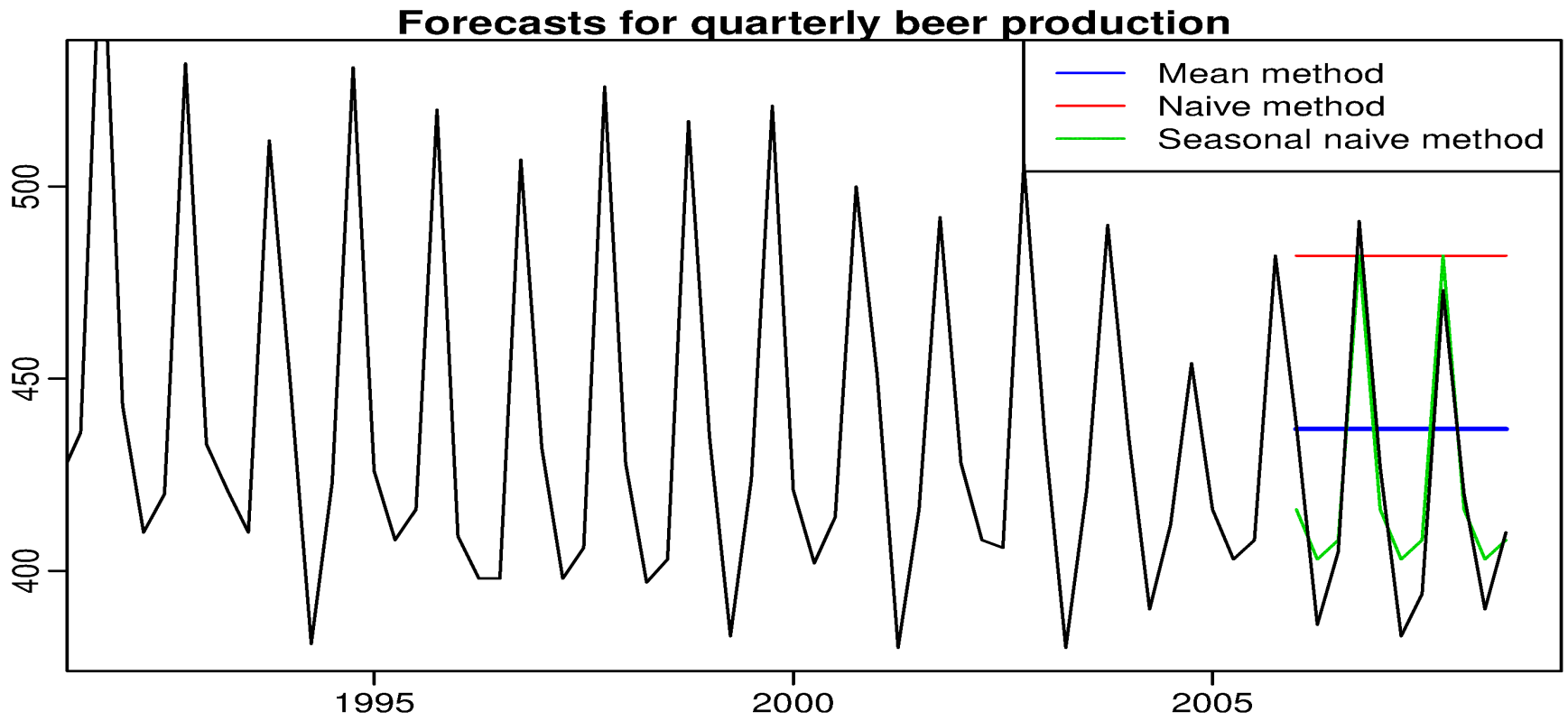
Symmetric mean absolute percentage error (MACAPE):

$$SMAPE = \frac{200}{T-R+1}\sum_{t=R}^{T}\left|\frac{\hat{y}_t - y_t}{\hat{y}_t + y_t}\right|.$$

Mean absolute scaled error:

$$MASE = \frac{1}{T-R+1}\sum_{t=R}^{T}\left|\hat{y}_t - y_t\right| \Bigg/ \frac{1}{T-1}\sum_{t=2}^{T}\left|y_t - y_{t-1}\right|.$$

Loss Functions of dotted forecasts



Algorithm	RMSE	MAE	MAPE	MASE
MA	38.01	33.78	8.17	2.30
Naive	70.91	63.91	15.88	4.35
Seasonal Naive	12.97	11.27	2.73	0.77

Relative Measures

Uncertainty coefficient (Theil's coefficient) estimates accuracy of forecast with respect to naive forecast :

$$U(d) = \sqrt{\frac{\sum_{t=R}^{T-d} \left(\hat{y}_{t+d|t} - y_{t+d}\right)^2}{\sum_{t=R}^{T-d} \left(y_t - y_{t+d}\right)^2}}, \quad d = 1, \ldots, D.$$

If $U(d) = 1$, then $\hat{y}_{t+d|t}$ is close to naive forecast ; if $U(d) < 1$, then forecast $\hat{y}_{t+d|t}$ is better than naive forecast, $U(d) > 1$ — naive forecast is better.

Comparing of two algorithms

$y_1, \ldots, y_T$ — time series,

$\hat{y}_{1R}, \ldots, \hat{y}_{1T}$ — forecasts of the first algorithm for period $R, \ldots, T$

$\hat{\varepsilon}_{1R}, \ldots, \hat{\varepsilon}_{1T}$ — residuals of the first algorithms,

$\hat{y}_{2R}, \ldots, \hat{y}_{2T}$ — forecasts of the second algorithm for period $R, \ldots, T$,

$\hat{\varepsilon}_{2R}, \ldots, \hat{\varepsilon}_{2T}$ — residals of the second algorithm;

$g\left(y_t, \hat{y}_{it}\right)$ — some loss function,

(for example, $|\hat{\varepsilon}_{it}|$ or $\hat{\varepsilon}_{it}^2$),

$$d_t = g\left(y_t, \hat{y}_{1t}\right) - g\left(y_t, \hat{y}_{2t}\right).$$

$$H_0: \text{ average } d_t = 0,$$
$$H_1: \text{ average } d_t <\neq> 0.$$

Wilcoxon signed-rank test:

$$W = \sum_{t=R}^{T} \operatorname{rank}\left(|d_t|\right) \operatorname{sign}\left(d_t\right).$$

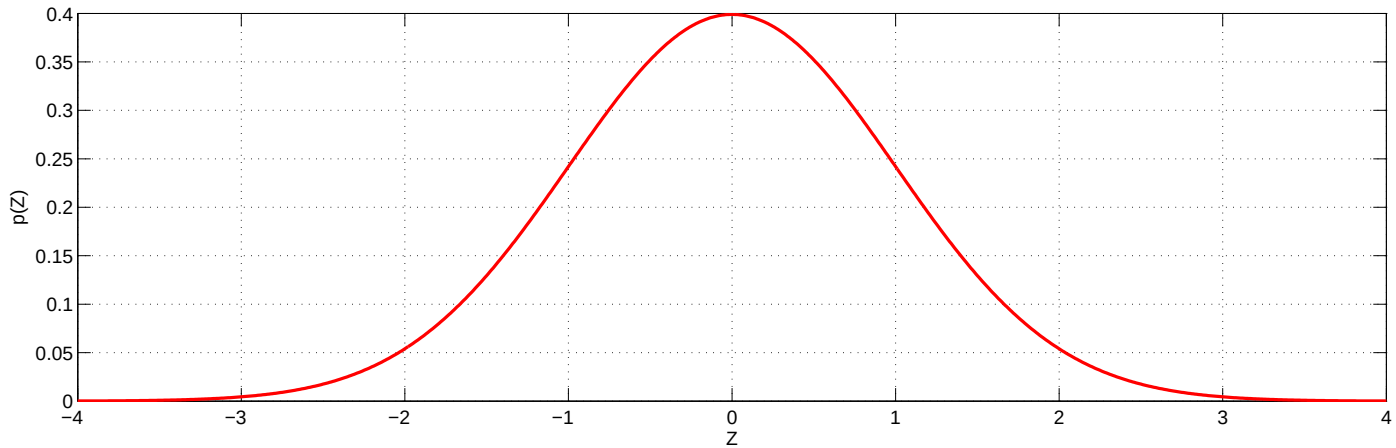
Diebold-Mariano test

null hypothesis: $H_0 : \mathbb{E}d_t = 0$

alternative hypothesis: $H_1 : \mathbb{E}d_t <\neq> 0$

statistics: $B = \frac{\bar{d}}{\sqrt{\hat{f}/T}}, \quad \hat{f} = \sum_{\tau=-M}^{M} \hat{r}_\tau, \quad M = T^{1/3}$

zero distribution: $N(0,1)$



Modification for short time series(Harvey, Leybourne, Newbold):

$$B^* = \frac{B}{\sqrt{\frac{T+1-2d+\frac{d(d-1)}{T}}{T}}}.$$

External Parameters Fittiong and Model Selection for TS

- X — feature space ($\mathbb{R}^n$); Y — answer space ($\mathbb{R}$);
  $X^\ell = (x_i, y_i)_{i=1}^\ell$ — train samples;
  $y_i = y(x_i)$, $y\colon X \to Y$ — unknown function;
  Loss function $\lambda(y_i, \hat{y}_i)$
- Learning method is a function: $\mu\colon 2^{X \times Y} \to \mathfrak{A}$
- loss of algorithm $A \in \mathfrak{A}$:

$$Loss_A = \mathsf{E}_{x,y}\left[\lambda\left(y, \mu\left(X^\ell\right)\right)^2\right]$$

- loss of learning method μ:

$$Q_\mu = \mathsf{E}_{X^\ell}\left[Loss_A\right]$$

Main problem of ML — is to minimize Q_μ
Estimation of loss of algorithm $A = \mu(X^\ell)$

$$Loss_A(X^\ell) = \sum_{i=1}^{\ell} \lambda\left(y_i, A(x_i)\right)$$

Cross Validation: Train, Validate and Test

CV approach: $X^{\ell} = X^{L} \bigcup X^{K}$

$$Q_{\mu}^{CV}(X^{\ell}) = Q\left(\mu(X^{L}), X^{K}\right) = Loss_{\mu(X^{L})}\left(X^{K}\right)$$

Cross Validation: Train, Validate and Test

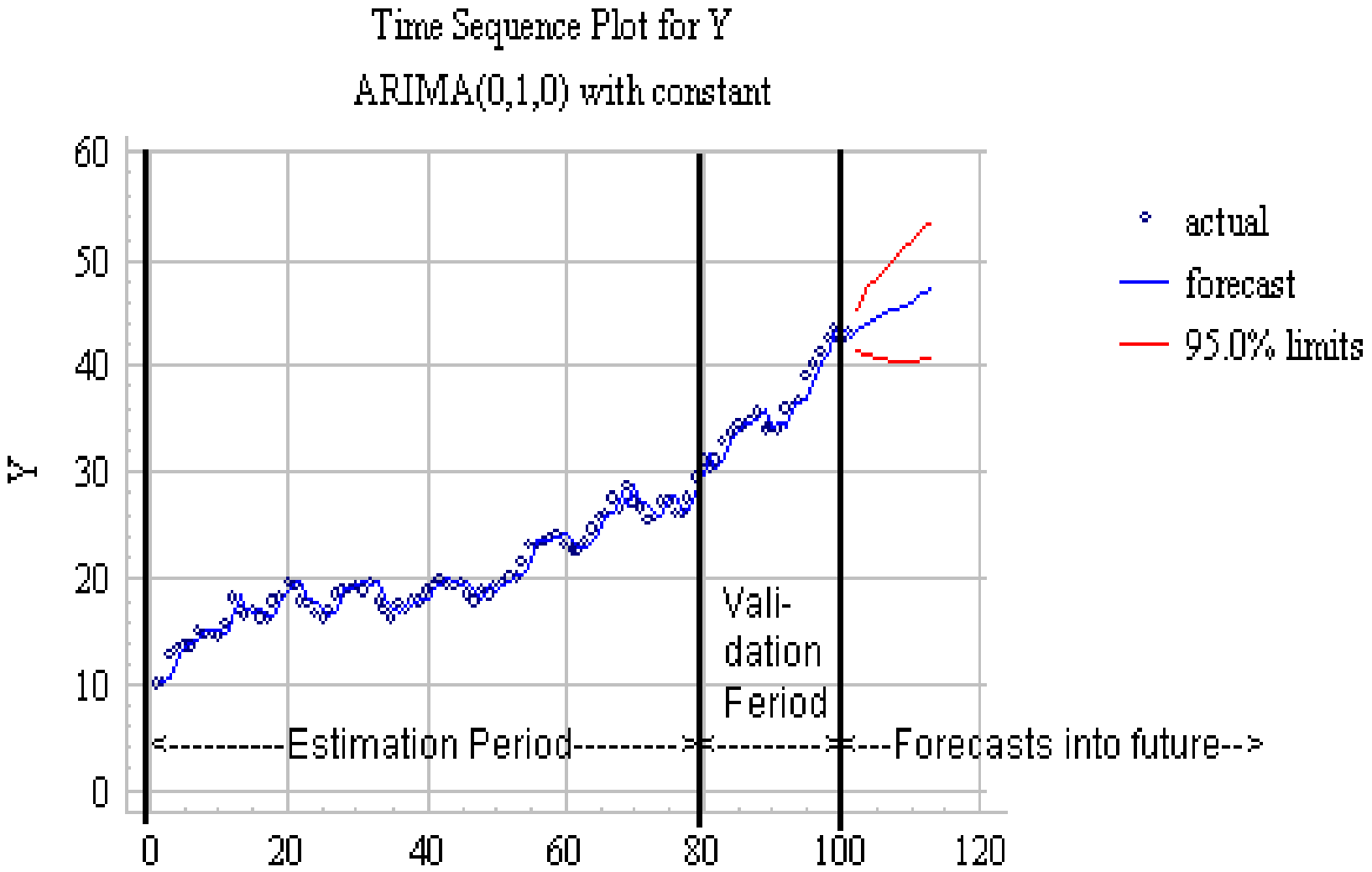
Most popular in ML q-fold CV: $X^\ell = X_1^{\ell_1} \bigcup \cdots \bigcup X_q^{\ell_q}$

$$\mathrm{Q}_\mu^{\mathrm{q-fold}}(X^\ell) = \frac{1}{q}\sum_{i=1}^{q} Q\left(\mu\left(X^\ell \backslash \{X_i^{\ell_i}\}\right), X_i^{\ell_i}\right)$$

Question: why test sample is needed?

Cross Validation: Train, Validate and Test

TS Forecasting:



Information Criteria

AIC (Akaike Information Criterion)

$$Q_{\mu}^{AIC}(X^{\ell}) = Q\big(\mu\left(X^{\ell}\right), X^{\ell}\big) + \frac{2\hat{\sigma}^2}{\ell} \cdot d$$

BIC (Bayes Information Criterion)

$$Q_{\mu}^{BIC}(X^{\ell}) = \frac{\ell}{\hat{\sigma}^2} Q\big(\mu\left(X^{\ell}\right), X^{\ell}\big) + \ln(\ell) \cdot d$$

HQIC (Hannan–Quinn information criterion)

$$Q_{\mu}^{HQIC}(X^{\ell}) = \frac{\ell}{\hat{\sigma}^2} Q\big(\mu\left(X^{\ell}\right), X^{\ell}\big) + \ln\ln(\ell) \cdot d$$

GARCH models

Generalized Autoregressive Conditional Heterescedastic model:

$$y_t = \mu + u_t \sim N(\mu, \sigma_t^2)$$

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \beta_i u_{t-i}^2 + \sum_{j=1}^{q} \gamma_i \sigma_{t-i}^2$$

In this model u_t^2 can be found as $ARMA(max(p,q),q)$

Overview of statistical models for TS forecasting

- ESM
- ARMA, ARIMA, ARIMAX, SARIMAX
- Adaptive Composition
- Adaptive Selection
- Aggregating Algorithm
- Dynamic Autoregressive models
- ARCH, GARCH, EGARCH ...
- VAR (vector autoregression)
- Gaussian State Space Models (UCM)
- GAS, GASX (generalized autoregression)
- Гусеница [Голяндина, 2003]

Jonathan D. Cryer, Kung-Sik Chan Time Series Analysis With Applications in R. Second Edition. Springer, 2008

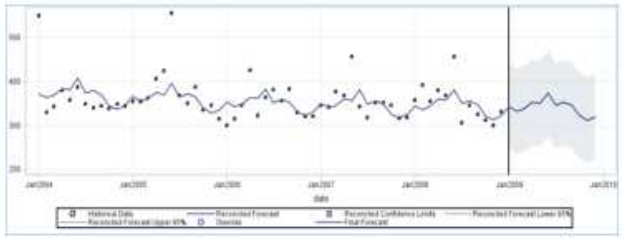
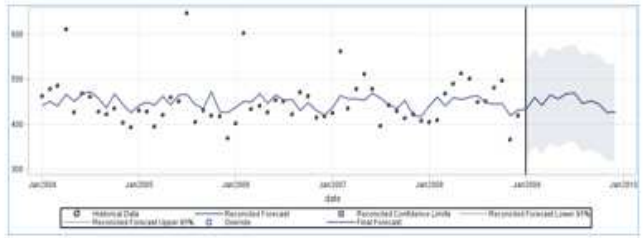
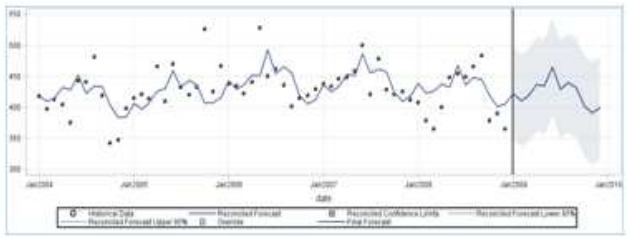
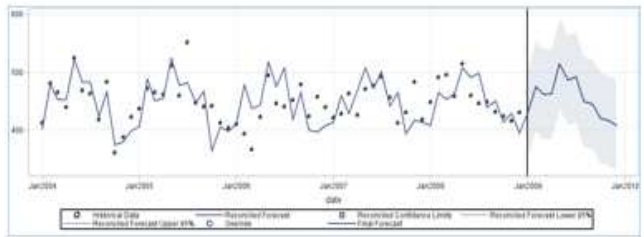
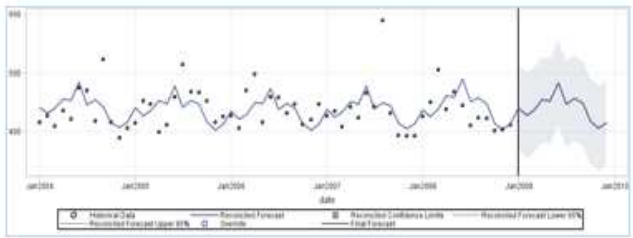
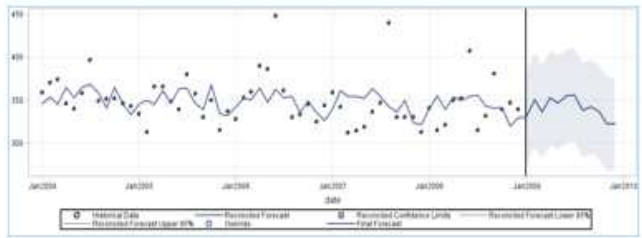
Магнус Я.Р. и др. Эконометрика. Начальный Курс М.: Дело, 2007

Python package for TS `http://www.pyflux.com/docs/index.html`

Hierarchy in Retail

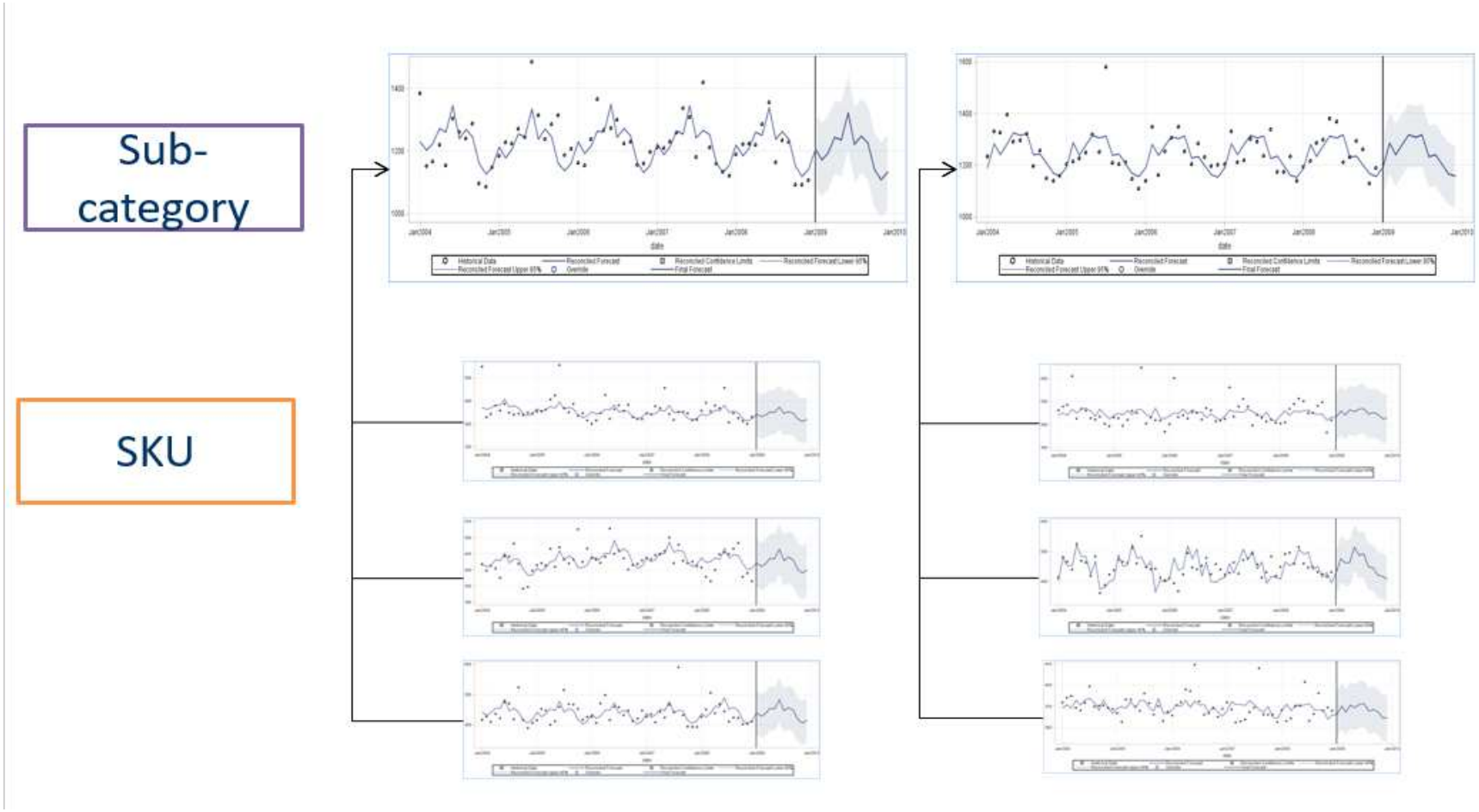
Example of hierarchy

SKU

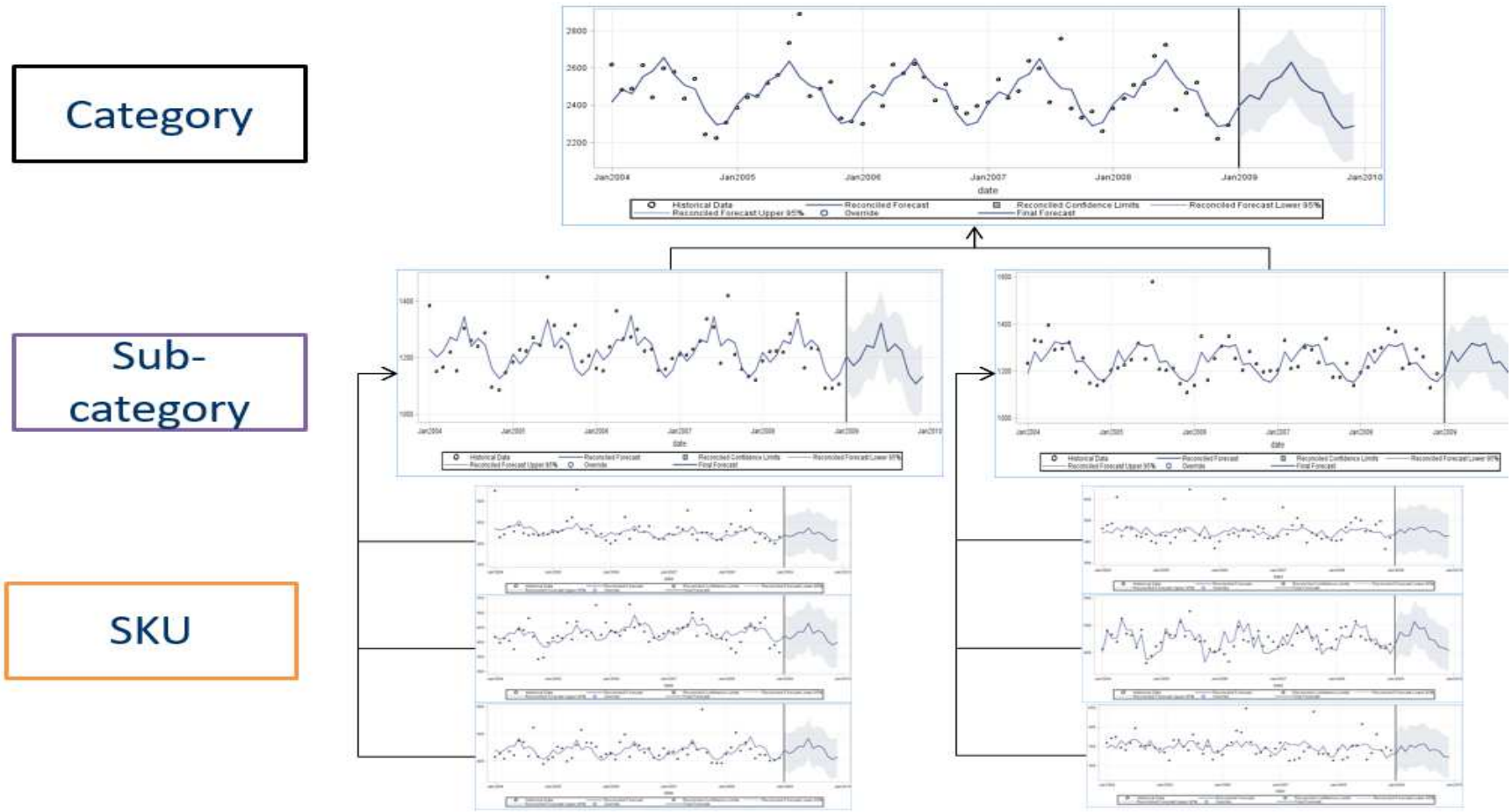
Hierarchy in Retail

Example of hierarchy



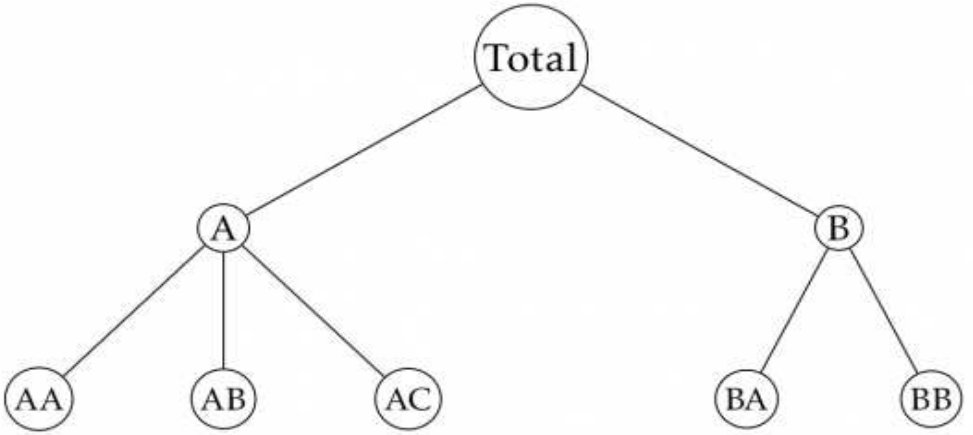
Hierarchy in Retail

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

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

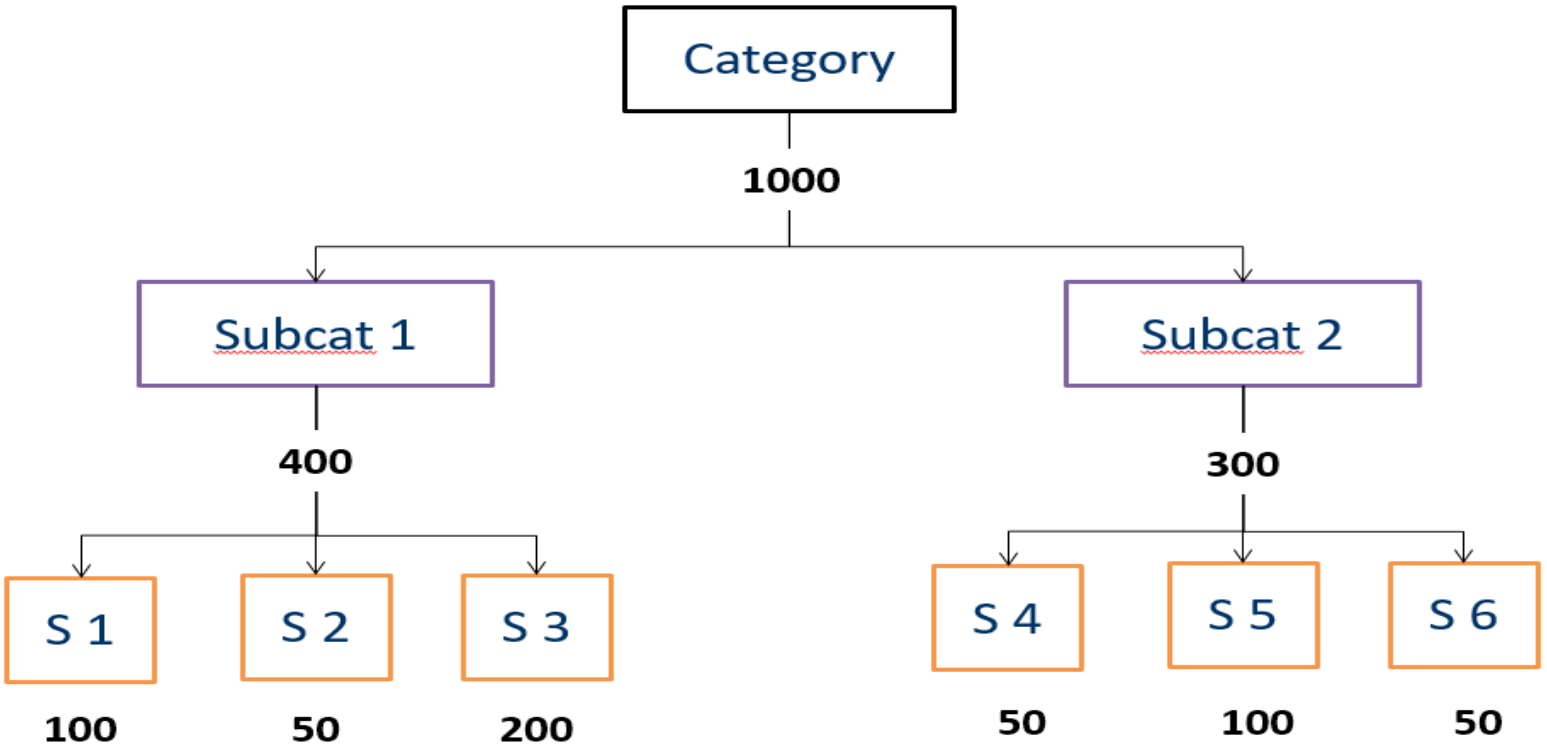
$$\varepsilon_h \approx S\varepsilon_{K,h} \Rightarrow \tilde{\mathbf{y}}_h = S\left(S^T S\right)^{-1} \hat{\mathbf{y}}_h.$$

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.

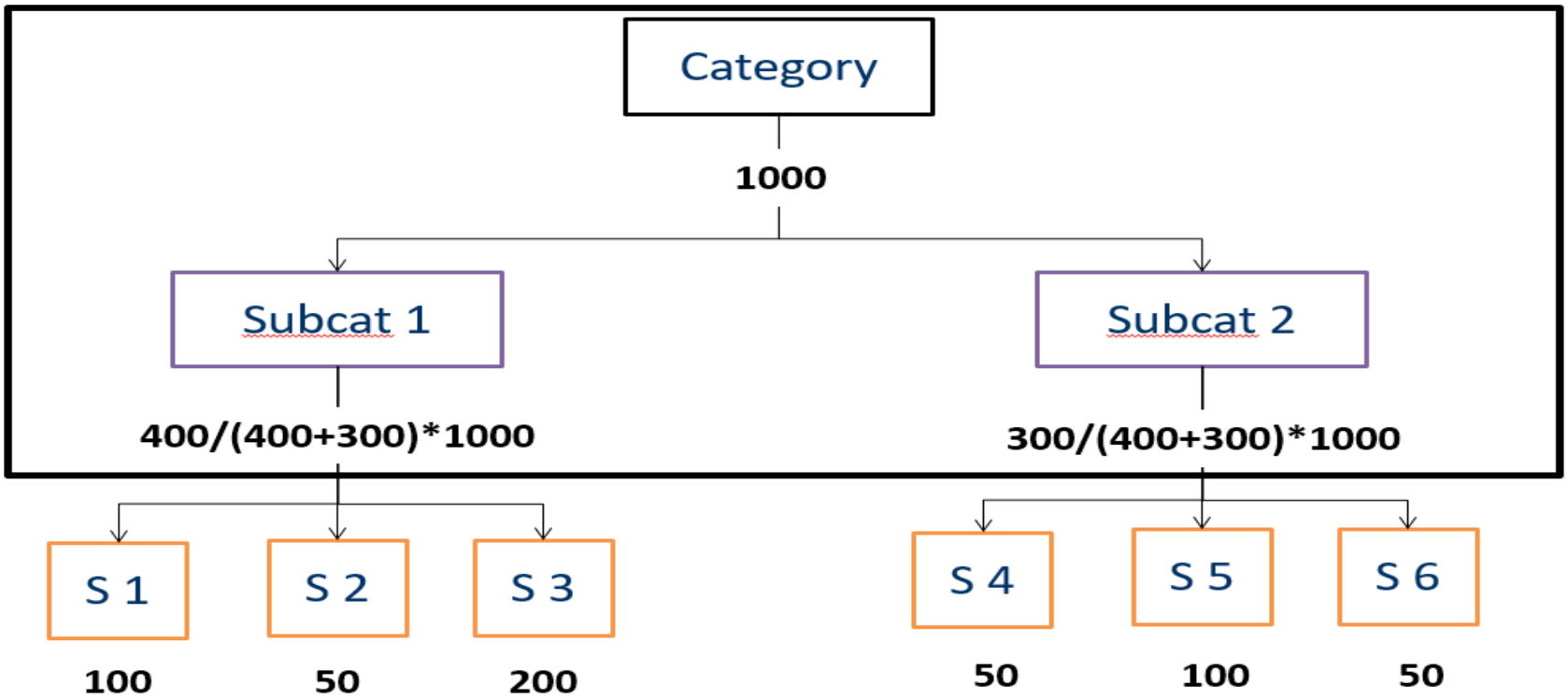
Example of forecast reconciliation

Top-down reconciliation



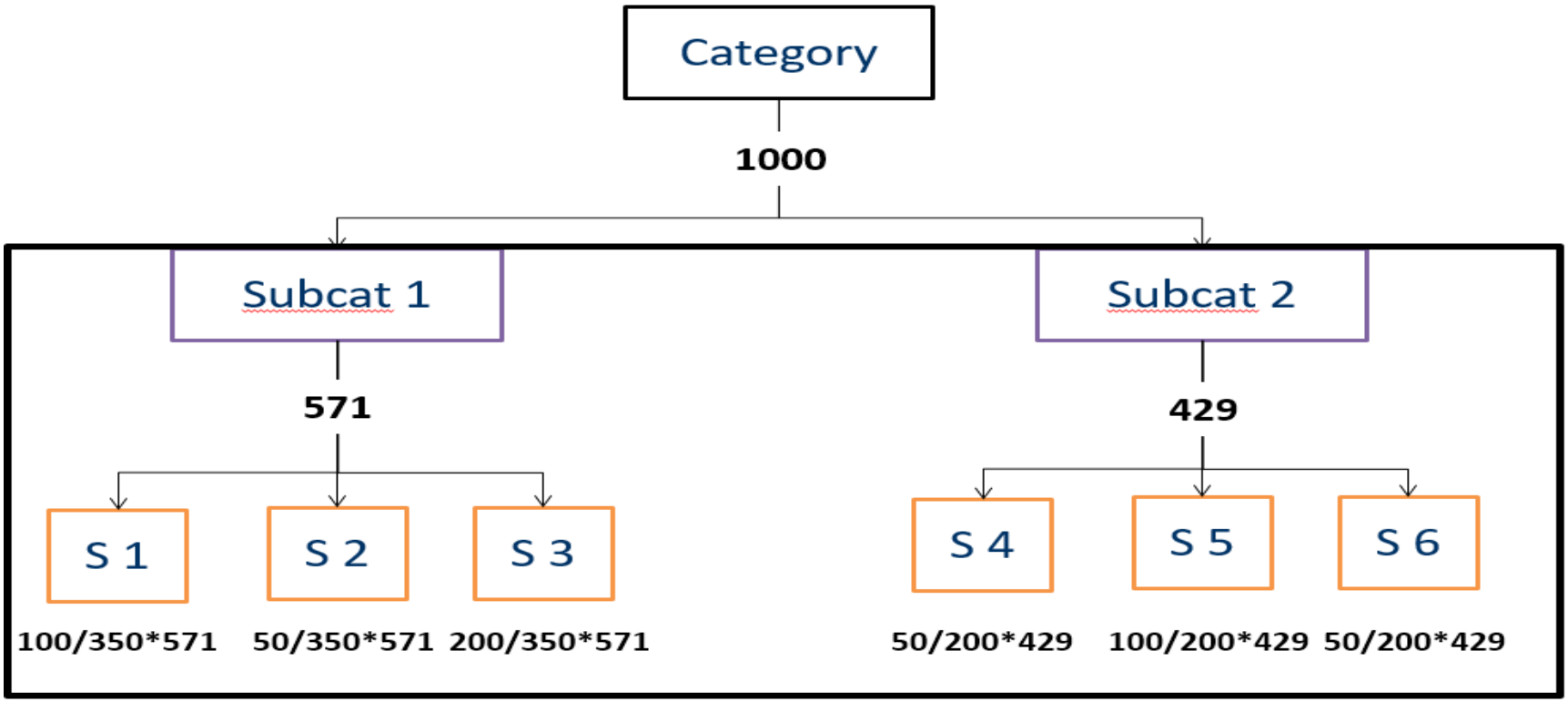
Example of forecast reconciliation

Top-down reconciliation



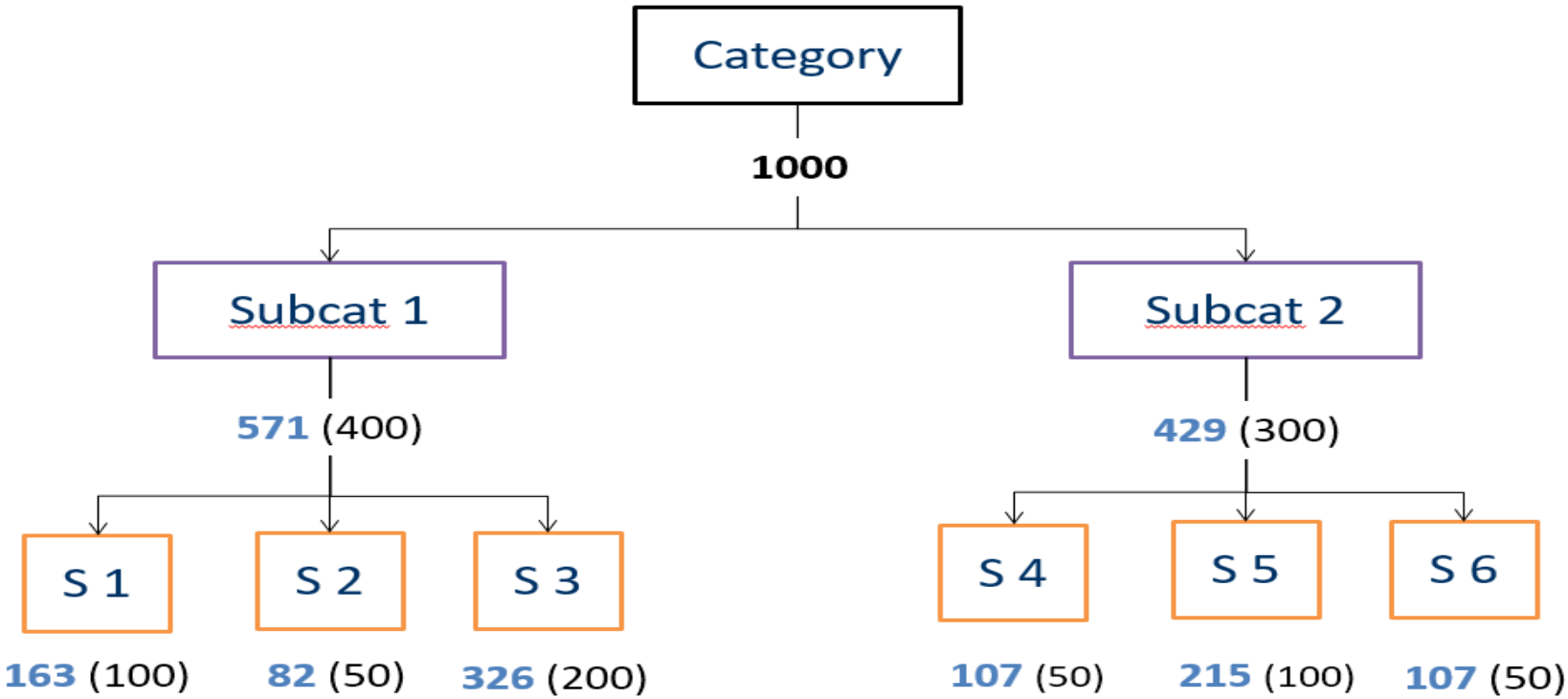
Example of forecast reconciliation

Top-down reconciliation



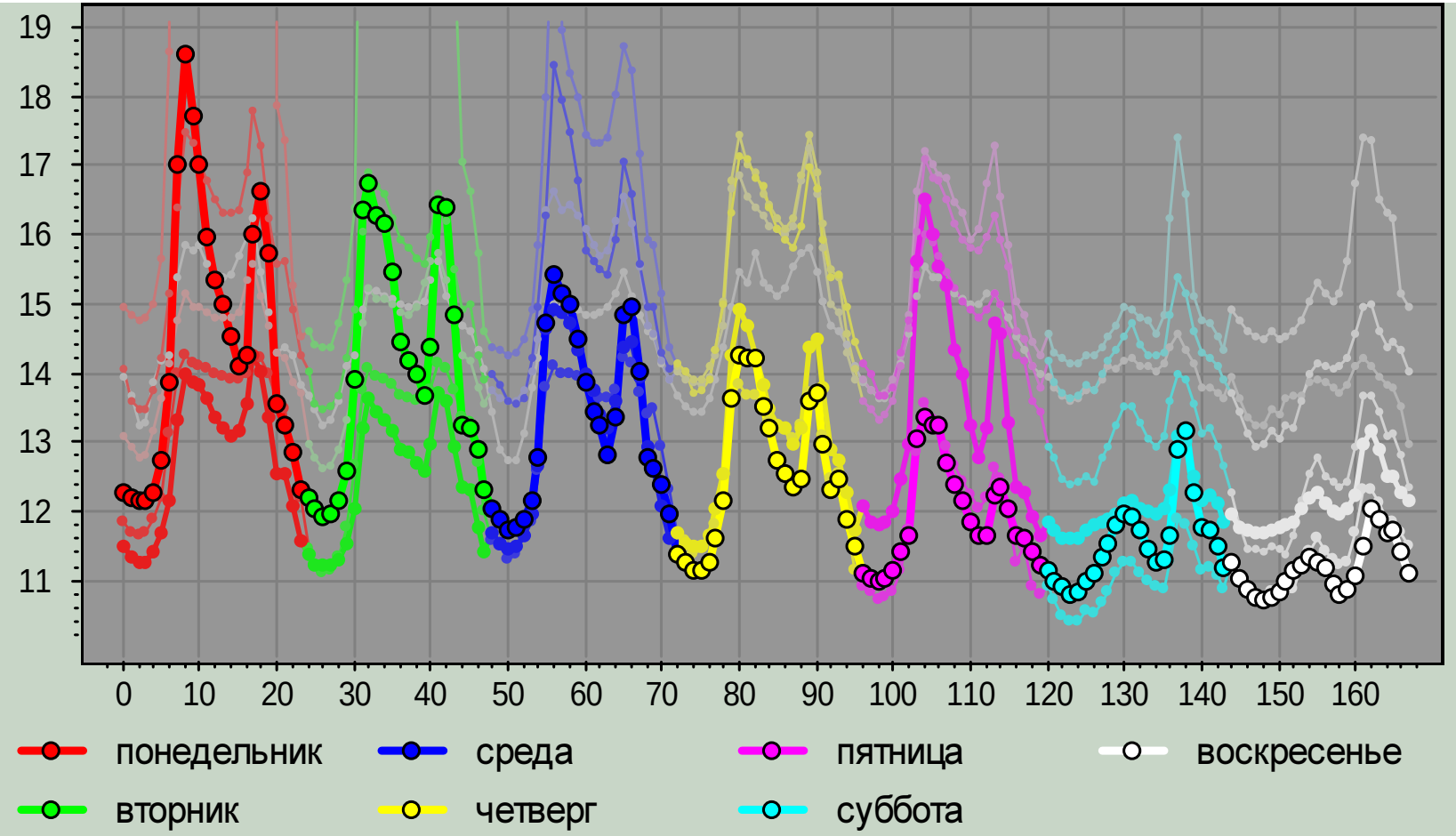
Example of forecast reconciliation

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:

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

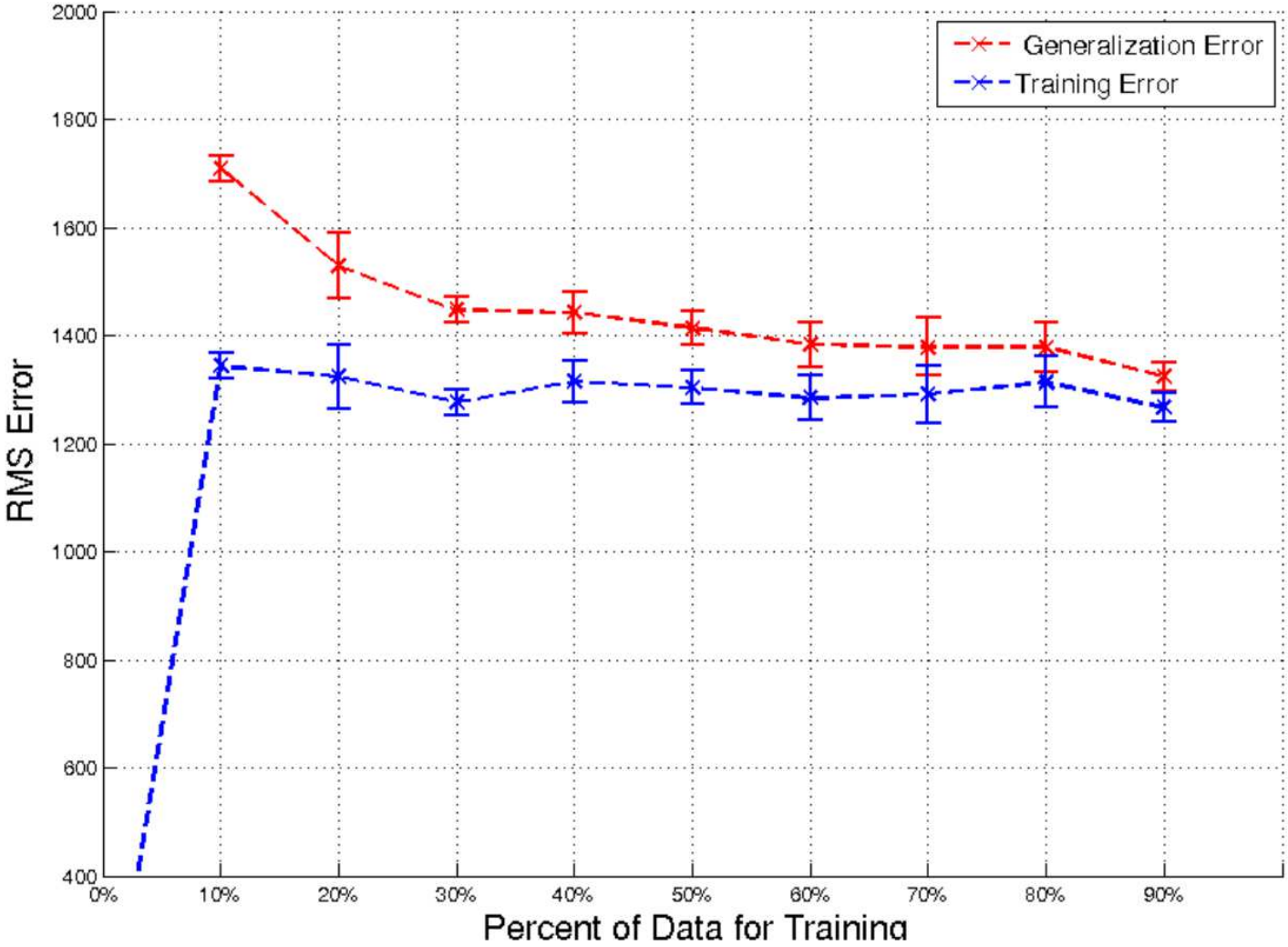
Loss Functional:

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

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:

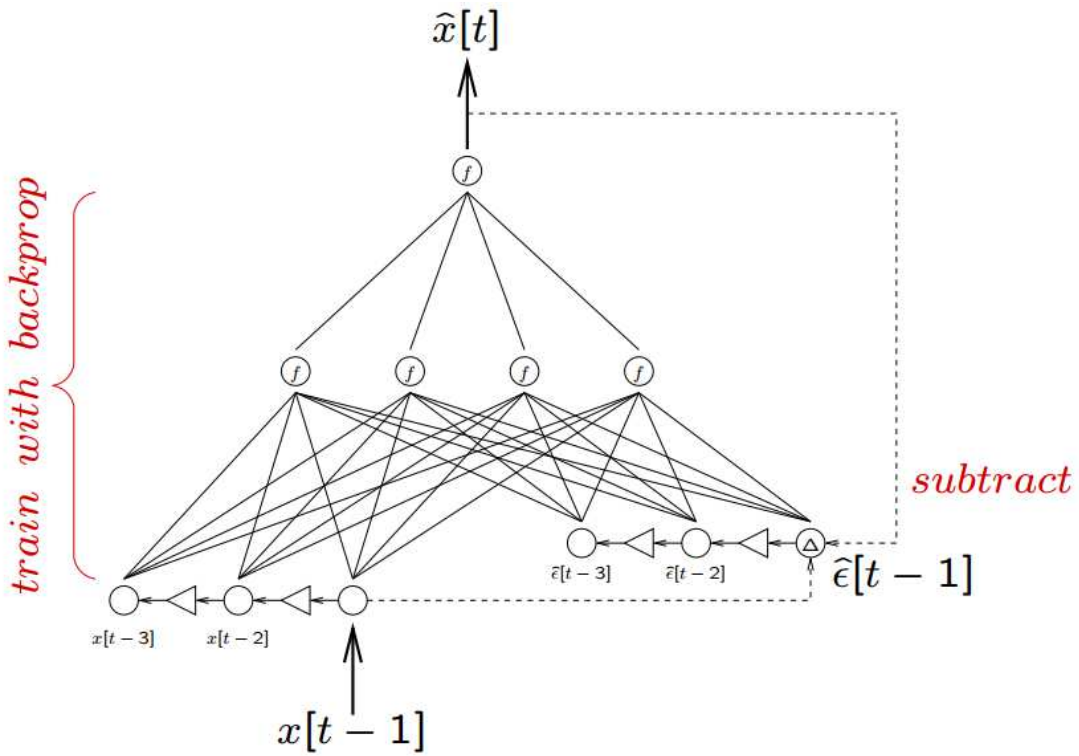
Таблица: Accuracy of forecast for different models

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%

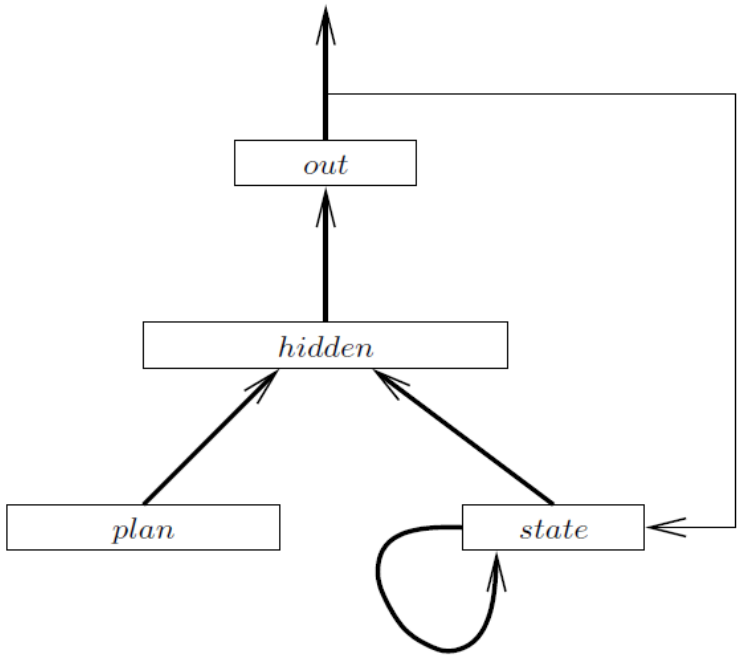
Enzo Busetti etc. Deep Learning for Time Series Modeling. CS 229 final Project Report, 2012.

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:

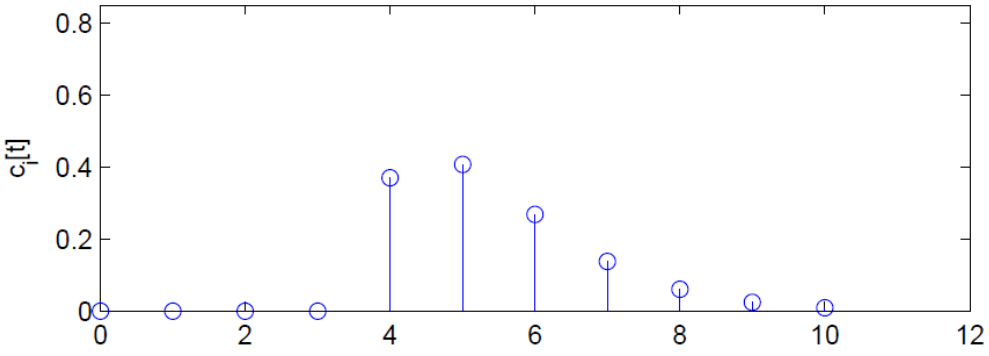
- delta-function $c_t = \delta_d(t)$
- exponential weights: $c_t = (1-\alpha)\alpha^t$

Weights of memory term

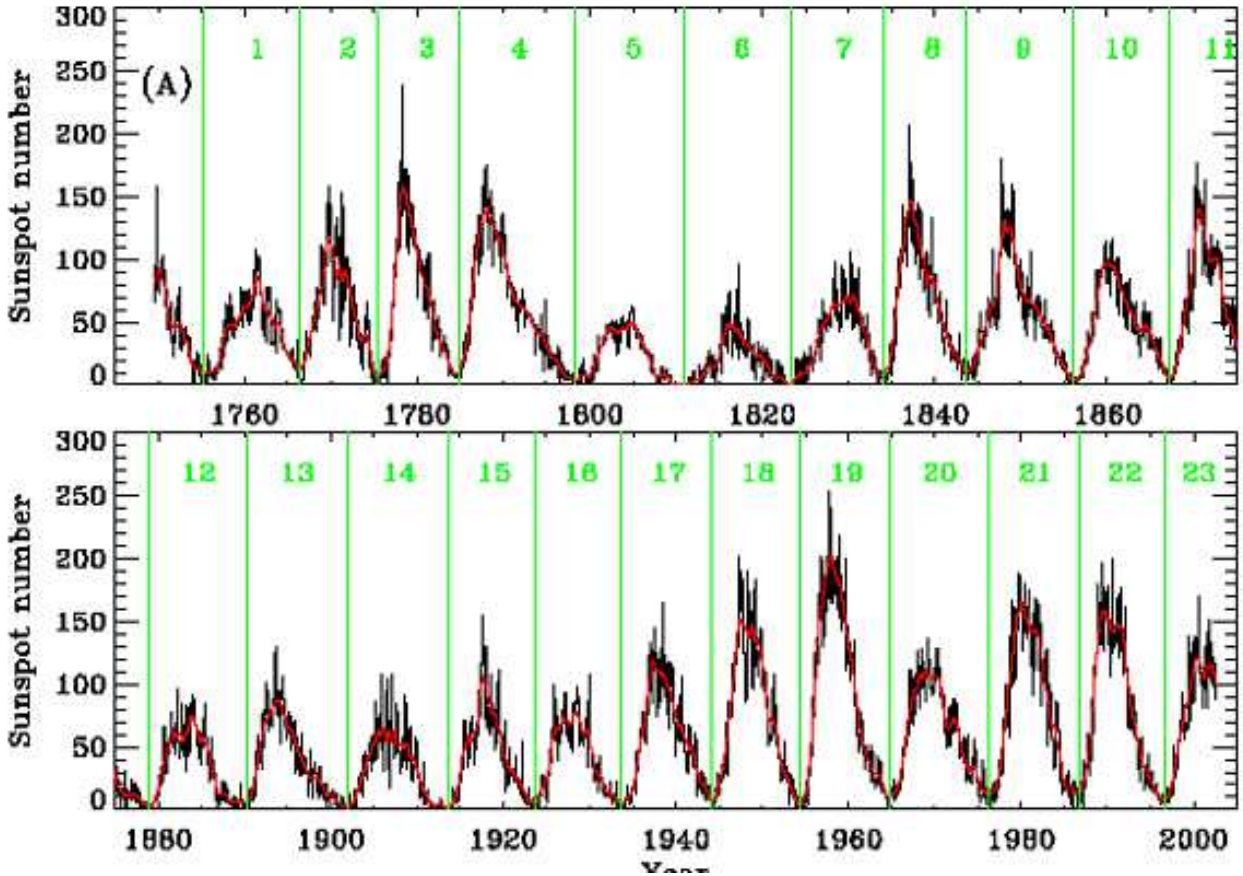
- $c_t = \begin{cases} \binom{t}{d}(1-\alpha)^{d+1} \cdot \alpha^{t-d} & \text{если } t \geq d \\ 0, & \text{иначе} \end{cases}$

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

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



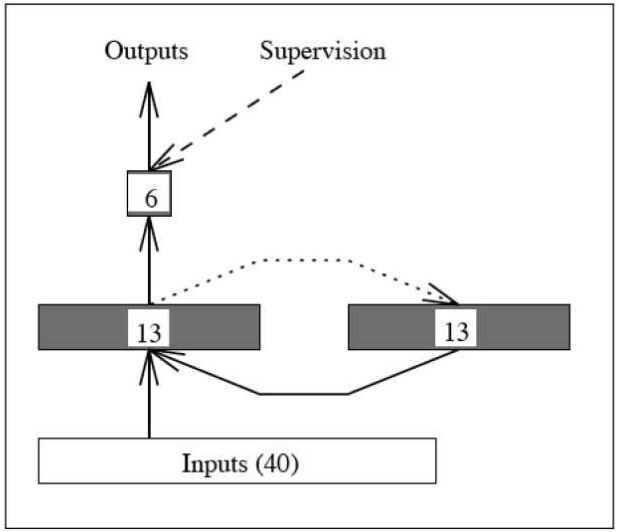
Prediction of sunspots



- 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. there are a lot of measures for accuracy of forecasts;
2. other well-known methods of TS forecasting can be explained in terms of AR-MA models;
3. DNN can be interpreted as generalization of ARIMA model
4. complicated algorithms (regressions, RNN) are useful for time series of complicated structure (several types of seasonality, high modality of time series);
5. for TS forecasting has some specific approaches (hierarchy forecasting)

Отзывы о лекции: `https://goo.gl/forms/akd1qfiKRnNR53Af1`

Literature

- Hyndman R.J., Athanasopoulos G. *Forecasting: principles and practice*, 2016. `https://www.otexts.org/book/fpp`
- Diebold-Mariano criterion and its modifications for short time series — Harvey;
- White reality check Step-down procedure — Romano;
- Chow test — Chow;
- Sullivan R., Timmermann A., White H. (2003). *Forecast evaluation with shared data sets*. International Journal of Forecasting, 19(2), 217–227.