Time Series Forecasting.
3. Compositions

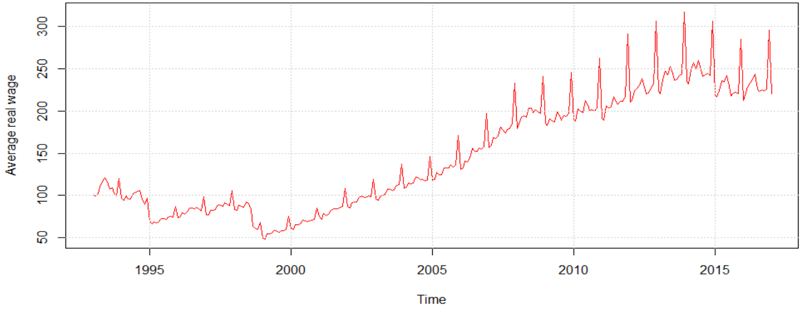
Alexey Romanenko alexromsput@gmail.com

FIVT MIPT, September 2017

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.

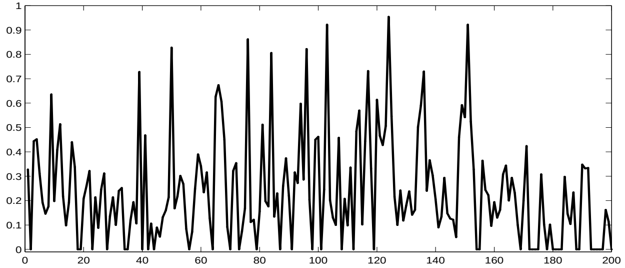
Another view to TS Forecasting Problem

An outcome space and a prediction space: $\Omega = \Gamma = [Y_1, Y_2] \subset \mathbb{R}$.

Definition

Time series is a sequence of elements from Ω^T : $Y = (y_1, \dots, y_T)$, where $y_t \in \Omega,\ t = \overline{1, T}$. Element $y_t \in \Omega$ is a point of the time series.

Time series



Online learning

Definition (Game)

Game G comprises $\langle \Omega, \Gamma, \lambda \rangle$ where Ω is a set of outcomes, Γ is a prediction set and $\lambda : \Omega \times \Gamma \to \mathbb{R}^+ \cup \{\infty\}$ is a loss function.

Definition (Forecasting Algorithm)

Forecasting Algorithm is function $A : \Omega^* \to \Gamma$, $\hat{y}^A_{T+1} = A(y_1, \ldots, y_T)$, where $\hat{y}^A_{T+1}$ — forecast of TS point for the moment $T + 1$.

Online learning

Online learning protocol

For $t = 0, \dots, T, \dots$

1. predict value $\hat{y}_{t+1} \in \Gamma$;
2. obtain outcome $y_{t+1} \in \Omega$;
3. calculate loss $\lambda\left(y_{t+1}, \hat{y}_{t+1}\right)$.

Definition (loss process)

A loss process is cumulative loss at step T $\mathrm{Loss}_{\mathsf{A}}(T) = \sum_{t=1}^{T} \lambda(y_t, \hat{y}_t^A)$.

Simple games

Simple games examples:

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

$$\lambda(\omega, \gamma) = \begin{cases} -\log_2(1-\gamma), & \omega = 0; \\ -\log_2(\gamma), & \omega = 1. \end{cases}$$

- simple prediction game $\Omega = \Gamma = \{0, 1\}$,

$$\lambda(\omega, \gamma) = \begin{cases} 0, & \omega = \gamma; \\ 1, & \omega \neq \gamma. \end{cases}$$

Asymmetric Linear and Square Games

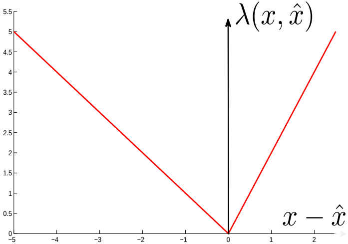
1. Game $G = \langle [Y_1, Y_2], [Y_1, Y_2], \lambda \rangle$ where

$$\lambda(y, \hat{y}) = \begin{cases} k_1 \cdot |y - \hat{y}|, & y - \hat{y} < 0, \\ k_2 \cdot |y - \hat{y}|, & y - \hat{y} \geq 0, \end{cases}$$

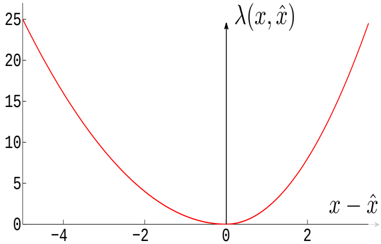
where $k_1 > 0, k_2 > 0$

$$\lambda(y, \hat{y}) = \begin{cases} k_1 \cdot (y - \hat{y})^2, & y - \hat{y} < 0, \\ k_2 \cdot (y - \hat{y})^2, & y - \hat{y} \geq 0, \end{cases}$$

where $k_1 > 0, k_2 > 0$



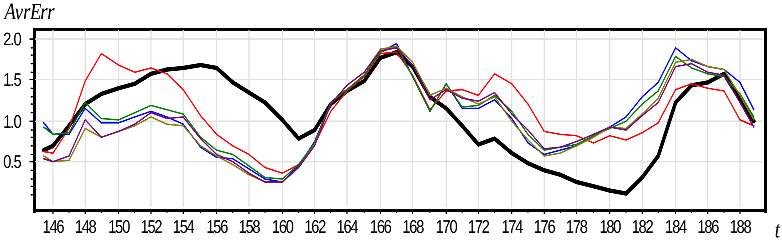
linear loss function



square loss function

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 N base algorithms $A_1, \dots, A_N$,
$\hat{y}^j_{t+d}$ — 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.

The best base algorithm in the moment t:

$$j^*_t = \underset{j=1,\dots,N}{\text{argmin}}\ \tilde{e}^j_t.$$

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 N base algorithms $A_1, \dots, A_N$,
$\hat{y}^j_{t+d}$ — 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:

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

Adaptive weights:

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

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

1. 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: ???*
2. 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

Mixability of forecast algorithms

- let us have N forecast algorithms
- $\lambda(y_t, \hat{y}_{j,t})$ — loss of algorithm j at forecast of element y_t
- $\text{Loss}_j(T) = \sum_{t=1}^{T} \lambda(y_t, \hat{y}_{j,t})$ — cumulative loss of algorithm j by the time T
- AA — desired composition

Task: how can we mix forecasts of base algorithms so that

$$\text{Loss}_{AA}(T) \preceq \text{Loss}_j(T), \ \forall j = \overline{1, N}?$$

Idea: we can focus on cumulative loss $\text{Loss}_j(t)$ of each base algorithm j at every time point t

Kolmogorov Mean as an Aggregation of Arithmetic Mean

Kolmogorov Mean:

$$KM(z_1, \dots, y_n) = \varphi^{-1}\left(\frac{1}{n}\sum_{k=1}^{n}\varphi(y_k)\right) = \varphi^{-1}\left(\frac{\varphi(y_1) + \ldots + \varphi(y_n)}{n}\right)$$

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

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 $p_j \sim \exp^{-\eta \mathsf{Loss}_j(T)}$;

Final composition AA is built based on generalized mixability function:

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

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

Super-Prediction

Let us introduce several terms

- pseudo–prediction is a function:

$$f(\omega) : \Omega \to [0, +\infty];$$

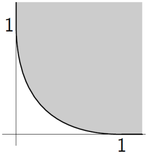
- set of outcomes Γ and loss function λ define real–predictions:

$$\lambda(\cdot, \gamma) : \Omega \to [0, +\infty];$$

- let us call superprediction those pseudo–predictions, which dominate some real-prediction:

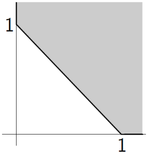
$$\exists \gamma \in \Gamma \colon \lambda(\omega, \gamma) \leq g(\omega), \forall \omega \in \Omega;$$

Example of super-prediction



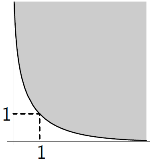
квадратичная игра

$\lambda(\omega, \gamma) = (\omega - \gamma)^2$



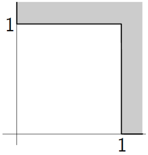
абсолютная игра

$\lambda(\omega, \gamma) = |\omega - \gamma|$



логарифмическая игра

$$\lambda(\omega, \gamma) = \begin{cases} -\log_2(1-\gamma), & \omega = 0 \\ -\log_2 \gamma, & \omega = 1 \end{cases}$$

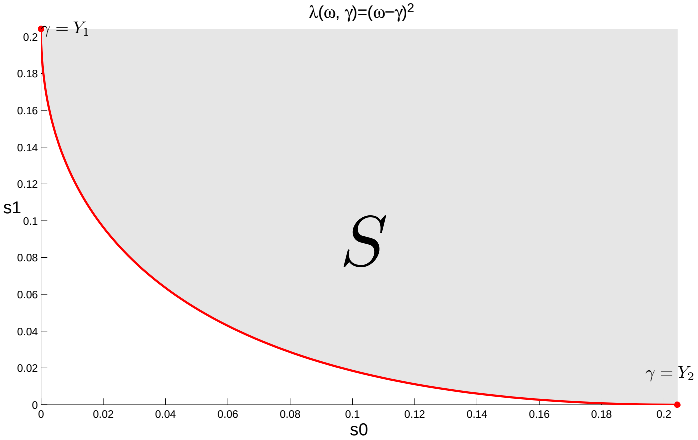


простая предсказательная игра

$$\lambda(\omega, \gamma) = \begin{cases} 0, & \omega = \gamma \\ 1, & \omega \neq \gamma \end{cases}$$

Super-prediction set for squared game

Game $G = \left\langle [Y_1, Y_2], [Y_1, Y_2], \lambda = (\omega - \gamma)^2 \right\rangle$



Main theoretical result

Theorem (V. Vovk)

If $g(\omega) = \log_{\beta}\left(\sum_{j=1}^{N} \frac{1}{N}\beta^{\mathsf{Loss}_j(T)+\lambda(\omega,\hat{\gamma}_{j,T+1})}\right)$, *then*

$c(\beta)\cdot g(\omega)$ — *super–prediction*;

That means

- in all observable games: $\exists\gamma\in\Gamma\quad\forall\omega\in\Omega$

$$\lambda(\omega,\gamma) \leq c(\beta)\cdot\log_{\beta}\left(\sum_{j=1}^{N}\frac{1}{N}\beta^{\mathsf{Loss}_j(T)+\lambda(\omega,\hat{\gamma}_{j,T+1})}\right)$$

- $c(\beta)\geq 1$
- if $c(\beta)=1$ for some β then game is (called) mixable

Mixable Games

- binary log-game is mixable ($\beta \geq 1/2$)
- binary squared game $\Omega = \{0,1\}$, $\Gamma = [0,1]$ is mixable ($\beta \geq 1$);
- (symmetric) squared game $\langle \Omega = \Gamma = [Y_2, Y_2], \lambda = (\omega - \gamma)^2 \rangle$ is mixable

$$\beta \geq \exp\left(-\frac{2}{(Y_2 - Y_1)^2}\right);$$

- asymmetric squared game $\langle \Omega = \Gamma = [Y_2, Y_2]$ is mixable

$$\beta \geq \exp\left(-\frac{1}{2 \cdot K \cdot (Y_2 - Y_1)^2}\right),$$

$$K = \tfrac{2k_1 - k_2 - k^*}{3(k_1 - k_2)} \cdot \tfrac{k_1 - 2k_2 + k^*}{3(k_1 - k_2)} \cdot \tfrac{k_1 + k_2 + k^*}{3}, k^* = \sqrt{(k_1 - k_2)^2 + k_1 \cdot k_2}.$$

Not-Mixable Games

- simple binary game is not mixable

$$c(\beta) = (\ln \beta) \,/\, \left(\ln \frac{1+\beta}{2}\right)$$

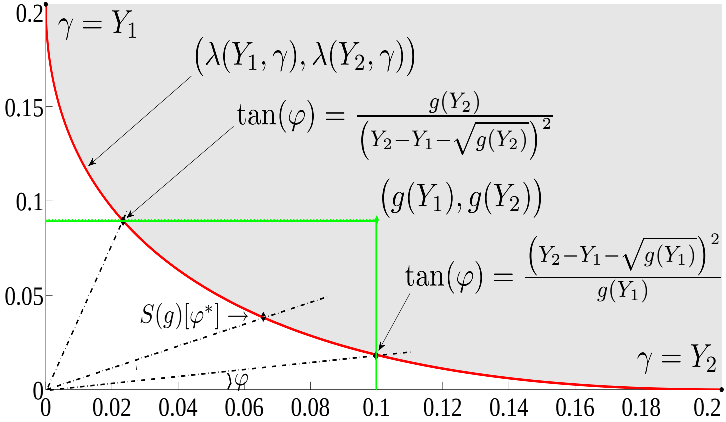
- binary absolute game is not mixable $c(\beta) = (\ln \beta) \,/\, \left(2 \ln \frac{1+\beta}{2}\right)$
- absolute game $\Omega = \Gamma = [Y_2, Y_2], \lambda(\omega, \gamma) = |\omega - \gamma|$ не смешиваемая

$$c(\beta) = ((Y_2 - Y_1) \ln \beta) \,/\, \left(2 \ln \frac{1 + \beta^{(Y_2 - Y_1)}}{2}\right)$$

- absolute asymmetric game is not mixable

$$c(\beta) = \frac{k_1 k_2 (Y_2 - Y_1) \ln(\beta)}{k_1 \ln\left(\frac{k_1}{k_1+k_2} \frac{1-\beta^{(k_1+k_2)(Y_2-Y_1)}}{1-\beta^{(k_1)(Y_2-Y_1)}}\right) + k_2 \ln\left(\frac{k_2}{k_1+k_2} \frac{1-\beta^{(k_1+k_2)(Y_2-Y_1)}}{1-\beta^{(k_2)(Y_2-Y_1)}}\right)}$$

How to build Substitution Function

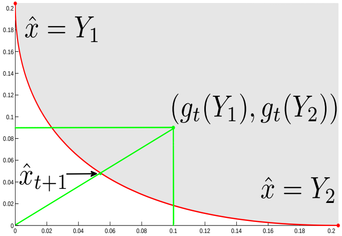


Condition for $S(g)$:

$$\lambda\big(Y_1, S(g)\big) \in [0, g(Y_1)]; \quad \lambda\big(Y_2, S(g)\big) \in [0, g(Y_2)]$$

Substitution Function for Squared Game

$$S(g) = \arg\min_{\hat{y}} \sup_{y} \left(\frac{\lambda(y,\hat{y})}{g(y)} \right)$$

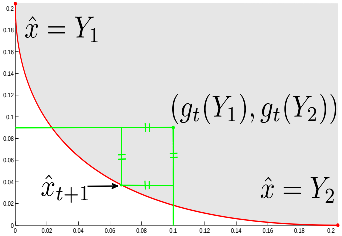


$$S\left(g\right) = \frac{Y_2\sqrt{g(Y_1)}+Y_1\sqrt{g(Y_2)}}{\sqrt{g(Y_1)}+\sqrt{g(Y_2)}}$$

$S(g) = \arg\min_{\hat{y}} \|u - v\|_{\infty}$, где

$u = \big(g(Y_1), g(Y_2)\big)$,

$v = \big((\hat{y} - Y_o 1)^2, (\hat{y} - Y_2)^2\big)$



$$S\left(g\right) = \frac{g(Y_1)-g(Y_2)}{2(Y_2-Y_1)} + \frac{Y_1+Y_2}{2}$$

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

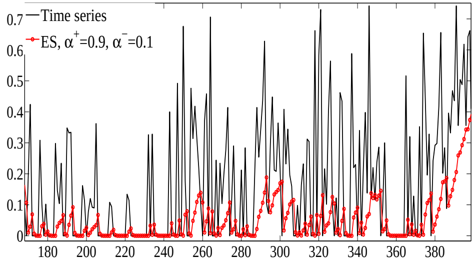
Theorem

The loss process AA in a asymmetric squared loss game G for $\forall (y_1, \ldots, y_T) \in [Y_1, Y_2]^T$, $\forall \{A^1, \ldots, A^N\}$ satisfies inequality:

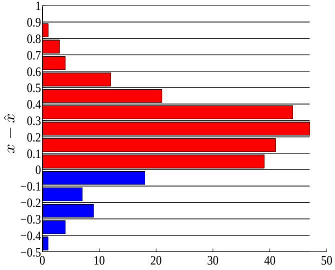
$$\text{Loss}_{AA}(T) \leq \min_{i=1,\ldots,N} \text{Loss}_{A^i}(T) + O\left(\ln(N)\right). \tag{1}$$

Data Description

1. 1913 time series from retail nets;
2. Length of time series varies from 50 to 1500 points;
3. Base algorithms: Exponential Smoothing (ES), Brown's Linear model (BL), Theil-Wage model (TW);
4. Training set for base algorithm: 200 time series;
5. Training set for parameters of compositions: 1000 time series.



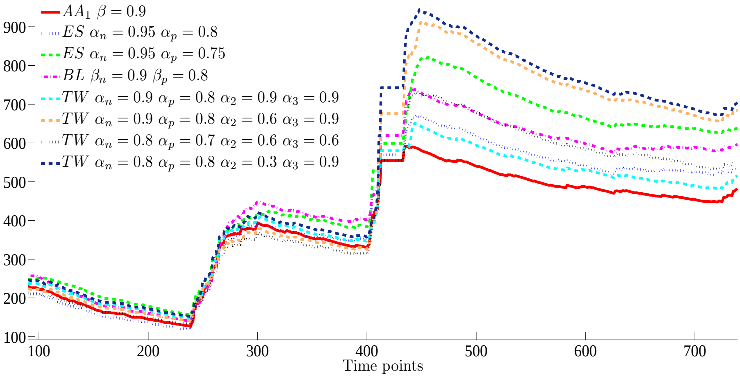
Time series forecast



Deviations

Comparison with Base Algorithms Example 1

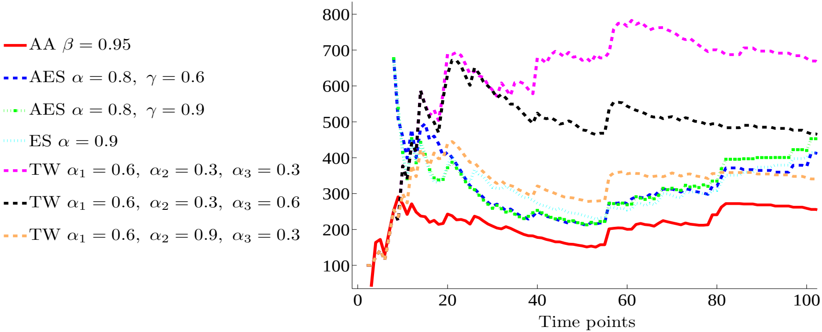
An experiment with real data (1 of 1000 time series)



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

Comparison with Base Algorithms Example 2

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



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

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

Conclusion

1. Aggregating Algorithm 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. theoretical bound of loss process slightly exceeds the actual loss process of compositions
5. **Compositions based on the aggregating algorithm can be applied in practice for different loss functions**

Literature

- 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`