

Aggregating of Adaptive Forecasting Algorithms

A. A. Romanenko

Moscow Institute of Physics and Technology

9 November 2016 · Moscow, Russia

Plan

- 1 Competitive Online Prediction
 - Online Predictions
 - Examples of games
 - Idea of Aggregating Algorithm
- 2 Wellcome to Aggregating Algorithm
 - Superpredictions and aggregation function
 - Mixability of some games
 - Making Prediction: Substitution Function
- 3 Experiments with Real Data
 - Parameters of Aggregating Algorithm
 - Comparison with Base Algorithms and Other Compositions
 - Comparison with Base Algorithms

Online Learning

Definition

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

Online learning protocol

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

- 1 predict value $\hat{x}_{t+1} \in \Gamma$;
- 2 obtain outcome $x_{t+1} \in \Omega$;
- 3 calculate loss $\lambda(x_{t+1}, \hat{x}_{t+1})$.

Definition (loss process)

A loss process is cumulative loss at step T

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

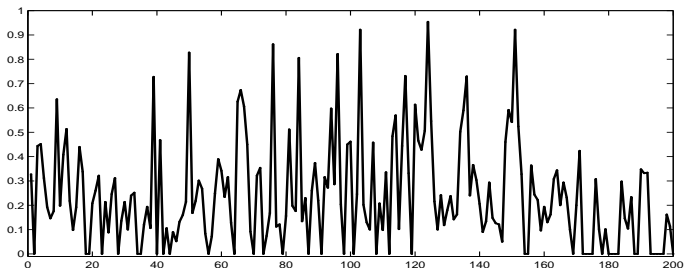
Time Series Prediction Problem

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 $\Omega^T : X = (x_1, \dots, x_T)$, where $x_t \in \Omega$, $t = \overline{1, T}$. Element $x_t \in \Omega$ is a point of the time series.

Time series



Simple games

Простейшие примеры игр:

- двоичные (бинарные) игры $\Omega = \{0, 1\}$, $\Gamma = [0, 1]$;
- 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 $\Omega = \Gamma = \{0, 1\}$,

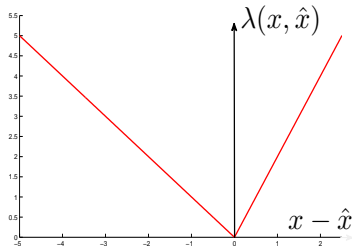
$$\lambda(\omega, \gamma) = \begin{cases} 0, & \omega = \gamma; \\ 1, & \omega \neq \gamma. \end{cases}$$

Asymmetric Linear and Square Games

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

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

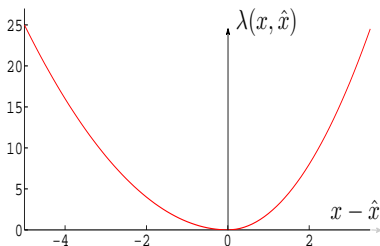
where $k_1 > 0, k_2 > 0$



linear loss function

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

where $k_1 > 0, k_2 > 0$



square loss function

Потери важнее прогнозов

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

1 Задача 1

- базовый алгоритм 1 строит константный прогноз 0;
- как построить прогноз композиции \mathfrak{A} , чтобы

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

- Ответ: ???

2 Задача 2

- базовый алгоритм 1 получает средний штраф $\frac{1}{2}$
- как построить прогноз композиции \mathfrak{A} , чтобы

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

- Ответ: строить константный прогноз $\frac{1}{2}$

Резюме: важнее смотреть на потери, а не на сам прогноз

Смешивание алгоритмов прогнозирования

- пусть имеется N алгоритмов прогнозирования
- $\lambda(y_t, \hat{y}_{j,t})$ — потери алгоритма j при прогнозе элемента y_t
- $\text{Loss}_j(T) = \sum_{t=1}^T \lambda(y_t, \hat{y}_{j,t})$ — суммарные потери алгоритма j к моменту времени T
- \mathfrak{M} — искомая композиция

Задача: как смешать прогнозы базовых алгоритмов, чтобы

$$\text{Loss}_{\mathfrak{M}}(T) \leq \text{Loss}_j(T), \forall j = \overline{1, N}?$$

Идея: ориентироваться в каждый момент времени t на накопленные потери $\text{Loss}_j(t)$ каждого базового алгоритма j

Среднее Колмогорова как обобщение среднего арифметического

Среднее Колмогорова:

$$M(x_1, \dots, x_n) = \varphi^{-1} \left(\frac{1}{n} \sum_{k=1}^n \varphi(x_k) \right) = \varphi^{-1} \left(\frac{\varphi(x_1) + \dots + \varphi(x_n)}{n} \right)$$

- $\varphi(x) = x \Rightarrow M(x_1, \dots, x_n) = \frac{x_1 + \dots + x_n}{n}$ — среднее арифметическое;
- $\varphi(x) = x^{-1} \Rightarrow M(x_1, \dots, x_n) = \frac{n}{1/x_1 + \dots + 1/x_n}$ — среднее гармоническое;
- $\varphi(x) = \log(x) \Rightarrow M(x_1, \dots, x_n) = \sqrt[n]{x_1 \cdot \dots \cdot x_n}$ — среднее геометрическое;
- $\varphi(x) = e^x \Rightarrow \ln \left(\frac{1}{n} \sum_{k=1}^n e^{(x_k)} \right)$

Какой выбрать функцию агрегирования (смешивания), чтобы по ней строить предсказания?

Идея агрегирующего алгоритм В. Вовка

- "усреднять" (смешивать) не прогнозы, а потери;
- взвешивать потери в экспоненциальном пространстве $p_j \sim \exp^{-\eta \text{Loss}_j(T)}$;

Итоговая композиция АА строится на основе функции смешивания (generalized function):

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

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

Super-prediction (супер-предсказание)

Введём несколько терминов

- назовём **бабка-предсказанием** любую функцию вида

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

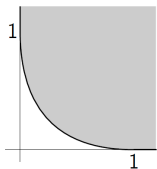
- задание множества предсказаний Γ и функции потерь λ выделяет из них допустимое (реальное) предсказание:

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

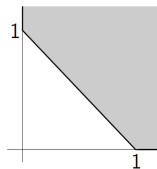
- назовём **супер-предсказанием** те **бабка-предсказания**, которые мажорируют некоторое допустимое предсказание:

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

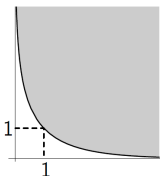
Example of super-prediction



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

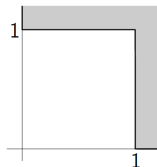


абсолютная игра
 $\lambda(\omega, \gamma) = |\omega - \gamma|$



логарифмическая игра

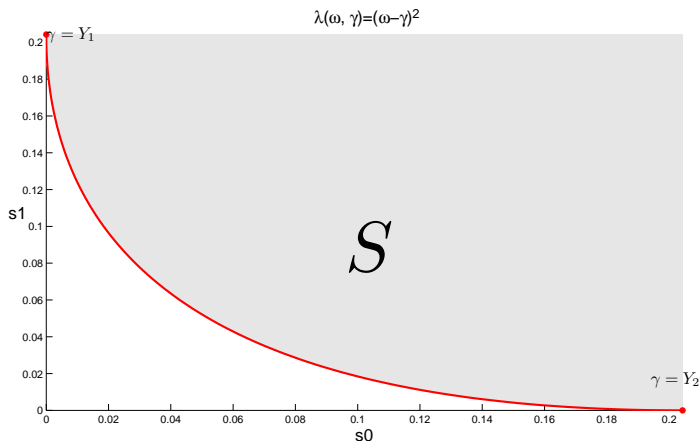
$$\lambda(\omega, \gamma) = \int -\log_2(1 - \gamma), \quad \omega = 0$$



простая предсказательная
 игра

Super-prediction set for squared game

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



Main theoretical result

Theorem (V. Vovk)

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

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

That means

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

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

- $c(\beta) \geq 1$
- if $c(\beta) = 1$ for some $\beta \in (0, 1)$ then game is (called) **mixable**

Mixable Games

- бинарная логарифмическая игра смешиваемая ($\beta \geq 1/2$)
- бинарная квадратичная игра $\Omega = \{0, 1\}$, $\Gamma = [0, 1]$
смешиваемая ($\beta \geq 1$);
- квадратичная игра $\langle \Omega = \Gamma = [Y_2, Y_2], \lambda = (\omega - \gamma)^2 \rangle$
смешиваемая

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

- квадратичная игра несимметричная игра $\langle \Omega = \Gamma = [Y_2, Y_2]$
смешиваемая

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

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

Not-Mixable Games

- простая бинарная игра не смешиваемая

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

- бинарная абсолютная игра не смешиваемая

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

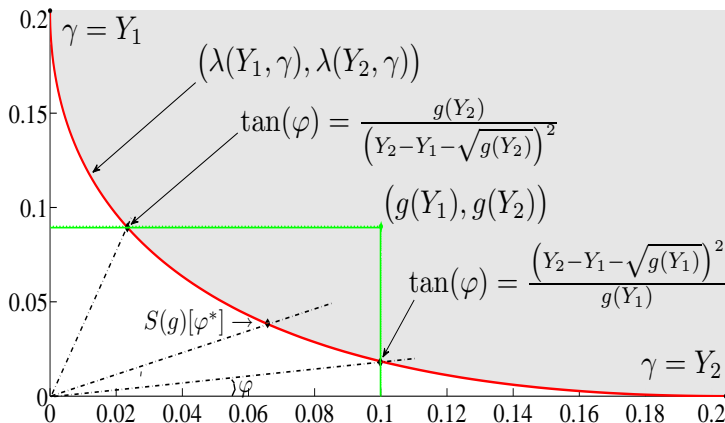
- абсолютная игра $\Omega = \Gamma = [Y_1, Y_2]$, $\lambda(\omega, \gamma) = |\omega - \gamma|$ не смешиваемая

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

- абсолютная несимметричная игра не смешиваемая

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

Idea of Substitution Function

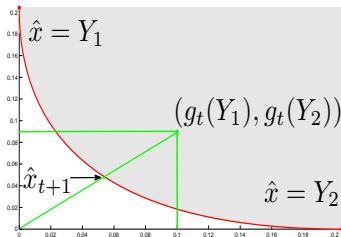


Условие выбора $S(g)$:

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

Substitution Function for Squared Game

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

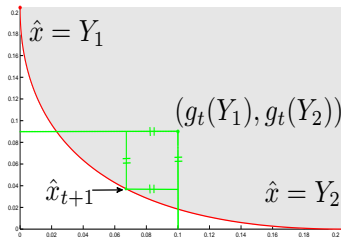


$$S(g) = \frac{Y_2 \sqrt{g(Y_1)} + Y_1 \sqrt{g(Y_2)}}{\sqrt{g(Y_1)} + \sqrt{g(Y_2)}}$$

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

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

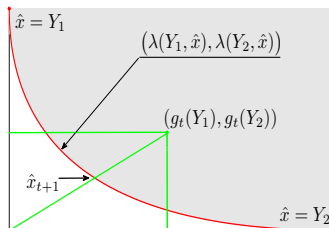
$$v = ((\hat{x} - Y_1)^2, (\hat{x} - Y_2)^2)$$



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

Substitution Function for Asymmetric Squared Game

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

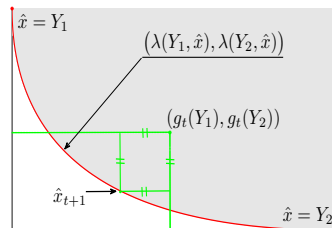


$$S_1(g) = \frac{Y_2 \sqrt{k_2 g(Y_1)} + Y_1 \sqrt{k_1 g(Y_2)}}{\sqrt{k_2 g(Y_1)} + \sqrt{k_1 g(Y_2)}}$$

$$S(g) = \arg \min_{\hat{x}} \|u - v\|_{\infty}, \text{ where}$$

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

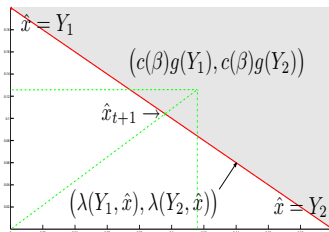
$$v = (\lambda(Y_1, \hat{x}), \lambda(Y_2, \hat{x}))$$



$$S_2(g) = \frac{k_2 Y_1 - k_1 Y_2}{k_1 - k_2} - \frac{\sqrt{k_2 k_1 (Y_1 - Y_2)^2 + g(Y_1) - g(Y_2)}}{k_1 - k_2}$$

Substitution Function for Asymmetric Linear Game

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

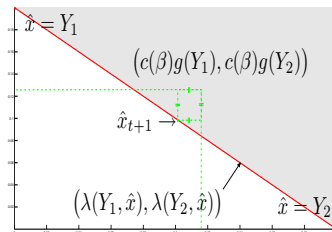


$$S(g) = \frac{Y_2 k_2 g(Y_1) + Y_1 k_2 g(Y_2)}{k_2 g(Y_1) + k_1 g(Y_2)}$$

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

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

$$v = (\lambda(Y_1, \hat{x}), \lambda(Y_2, \hat{x}))$$



$$S(g) = \frac{c(\beta)(g(Y_1) - g(Y_2))}{(k_1 + k_2)} + \frac{k_1 Y_1 + k_2 Y_2}{k_1 + k_2}$$

Агрегирующий алгоритм В. Вовка

Прогнозы композиций AA_1 и AA_2

Инициализация: веса базовых алгоритмов $p_{j,0} = 1/N$

Для $t = 0, \dots, T - 1$

- ❶ получить предсказания экспертов $\hat{y}_{j,t+1}, \forall j = \overline{1, N}$;
- ❷ построить функцию смешивания:

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

- ❸ $\hat{y}_{AA_1,t+1} = \frac{Y_2 \sqrt{g(Y_1)} + Y_1 \sqrt{g(Y_2)}}{\sqrt{g(Y_1)} + \sqrt{g(Y_2)}};$
 $\hat{y}_{AA_2,t+1} = \frac{g(Y_1) - g(Y_2)}{2(Y_2 - Y_1)} + \frac{Y_1 + Y_2}{2};$
- ❹ получить исход y_{t+1} ; вычислить ошибку $\lambda(y_{t+1}, \hat{y}_{t+1})$;
- ❺ пересчитать веса экспертов $p_{j,t+1} = \beta^{\lambda(y_{t+1}, \hat{y}_{j,t+1})} \cdot p_{j,t}.$

Loss Process Estimation

- Consider base forecast algorithms $\{A^1, \dots, A^N\}$.
- 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 $O(NT)$.

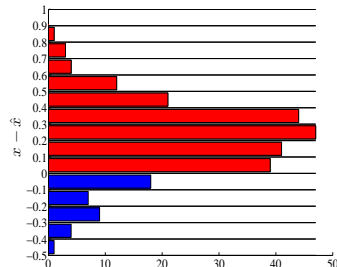
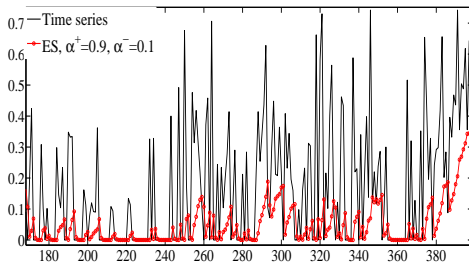
Theorem

The loss process **AA** in a **asymmetric loss game** G for $\forall (x_1, \dots, x_T) \in [Y_1, Y_2]^T, \forall \{A^1, \dots, A^M\}$ satisfies inequality:

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

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.

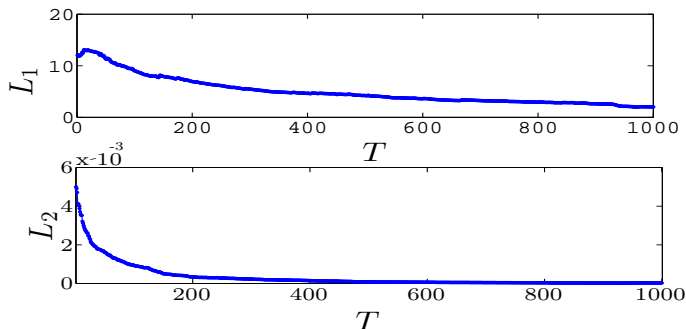


Initial Distribution of Expert Weights

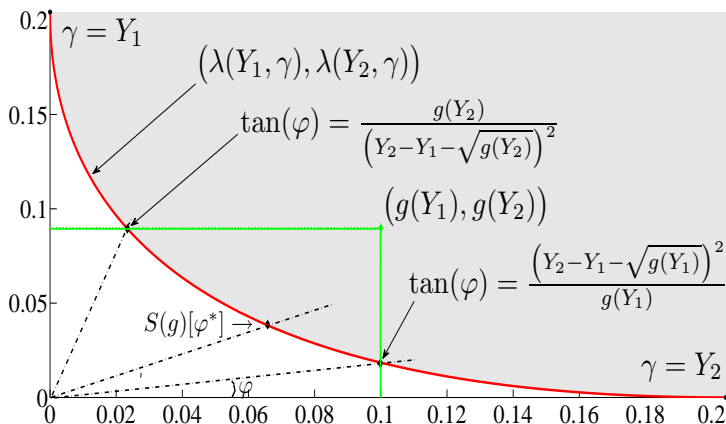
Let $\{AA_j\}_{j=1}^N$ be a set of compositions with different ρ_0 , $N \approx 10^4$

$$L_1(T) = \frac{1}{T} \left(\max_{j=1, N} \text{Loss}_{AA_j}(T) - \min_{j=1, N} \text{Loss}_{AA_j}(T) \right),$$

$$L_2(T) = \frac{1}{T} \left(\frac{\max_{j=1, N} \text{Loss}_{AA_j}(T) - \min_{j=1, 10} \text{Loss}_{AA_j}(T)}{\max_{j=1, N} \text{Loss}_{AA_j}(T) + \min_{j=1, 10} \text{Loss}_{AA_j}(T)} \right).$$

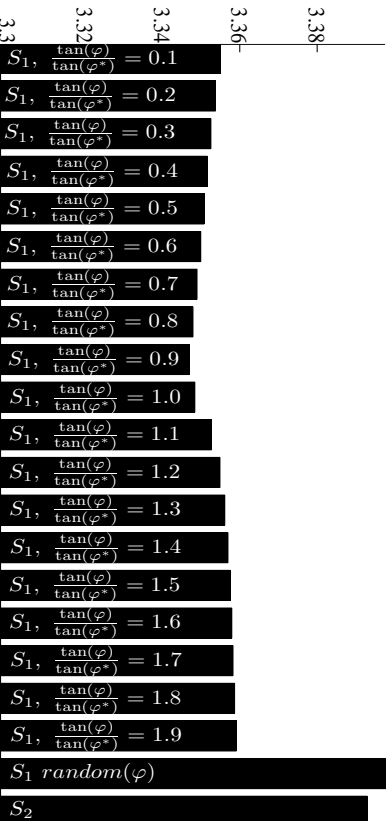


Idea of Substitution Function



Optimal Substitution Function

An experiment with real data (1 of 200 time series), $k_1 = 1$, $k_2 = 2$



In the next slides: AA_1 corresponds to S_1 , AA_2 corresponds to S_2

Theoretical Bounds

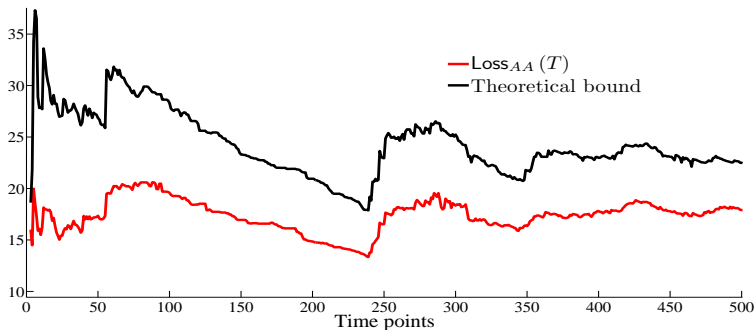
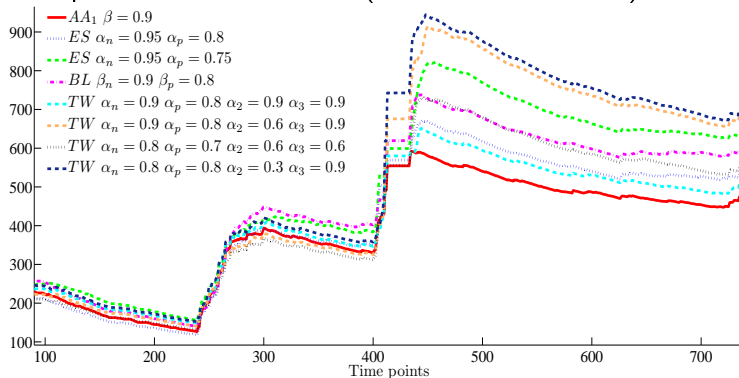


Table: MSE of AA_1 and the best expert averaged by 1000 time series

k_1/k_2	1	2	5	10	15	20
AA_1	21.69	32.24	57.33	94.17	110.4	139.9
BE	22.05	32.63	58.24	95.23	111.5	140.6
TB	25.16	38.2	71.80	99.44	141.1	179.7

Comparison with Base Algorithms Example 1

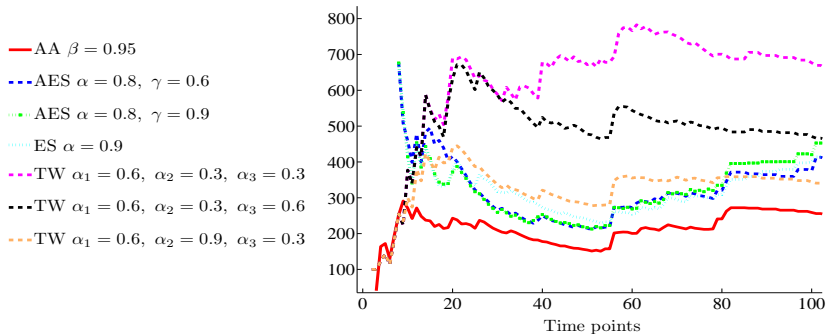
An experiment with real data (1 of 1000 time series)



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

Comparison with Base Algorithms Example 2

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



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

Comparison with Other Compositions

Table: 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%

Table: 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-consmng
- 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**