

Прикладной статистический анализ данных.  
10. Анализ временных рядов, часть вторая.

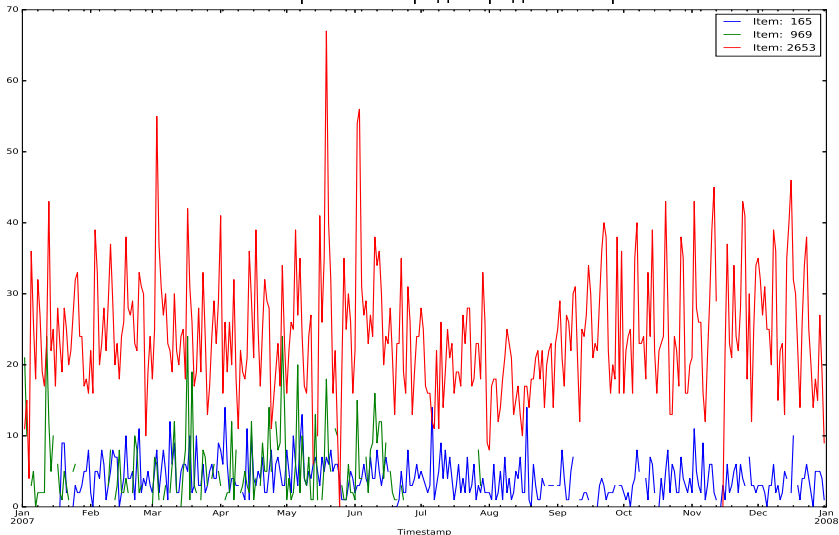
Алексей Романенко , alexromsput@gmail.com  
Евгений Рябенко, riabenko.e@gmail.com

2017



# Прогнозирование временного ряда

## Реальные временные ряды продаж товаров:





# Простое экспоненциальное сглаживание (метод Брауна)

Наивный прогноз:

$$\hat{y}_{T+1|T} = y_T.$$

Прогноз средним значением:

$$\hat{y}_{T+1|T} = \sum_{t=1}^T y_t.$$

Прогноз с помощью взвешенного среднего с экспоненциально убывающими весами:

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \dots$$

$\alpha \uparrow 1 \Rightarrow$  больший вес последним точкам,

$\alpha \downarrow 0 \Rightarrow$  большее сглаживание.

Наблюдение	$\alpha = 0.2$	$\alpha = 0.4$	$\alpha = 0.6$	$\alpha = 0.8$
$y_T$	0.2	0.4	0.6	0.8
$y_{T-1}$	0.16	0.24	0.24	0.16
$y_{T-2}$	0.128	0.144	0.096	0.032
$y_{T-3}$	0.1024	0.0864	0.0384	0.0064
$y_{T-4}$	0.08192	0.05184	0.01536	0.00128
$y_{T-5}$	0.065536	0.031104	0.006144	0.000256







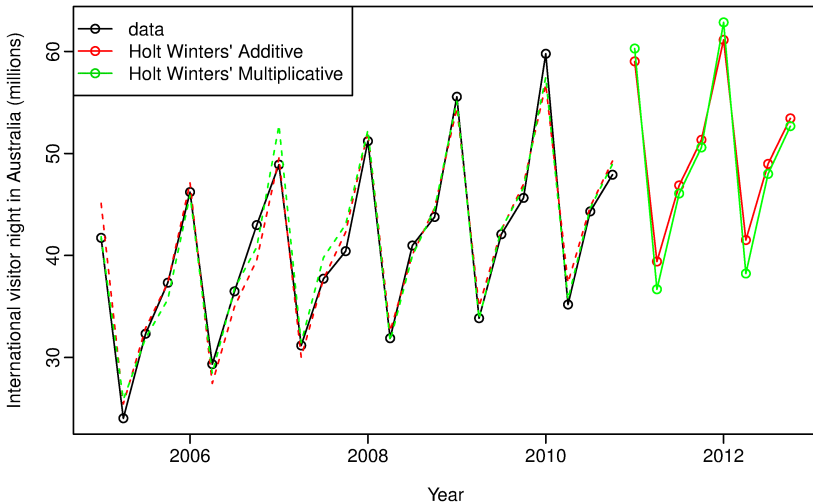








# Методы, учитывающие сезонность



Прогнозы с учётом тренда и сезонности количества ночей, проведённых туристами в Австралии.

# Модели экспоненциального сглаживания

	Сезонность		
Тренд	N (None)	A (Additive)	M (Multiplicative)
N (None)	(N,N)	(N,A)	(N,M)
A (Additive)	(A,N)	(A,A)	(A,M)
Ad (Additive damped)	(Ad,N)	(Ad,A)	(Ad,M)
M (Multiplicative)	(M,N)	(M,A)	(M,M)
Md (Multiplicative damped)	(Md,N)	(Md,A)	(Md,M)

Дополнительно можно предположить аддитивную (A) или мультипликативную (M) ошибку (тип ошибки не влияет на точечный прогноз). Мультипликативная ошибка подходит только для строго положительных рядов.

Итоговую модель можно записать в виде  $ETS(\cdot, \cdot, \cdot)$ .

# Настройка модели экспоненциального сглаживания

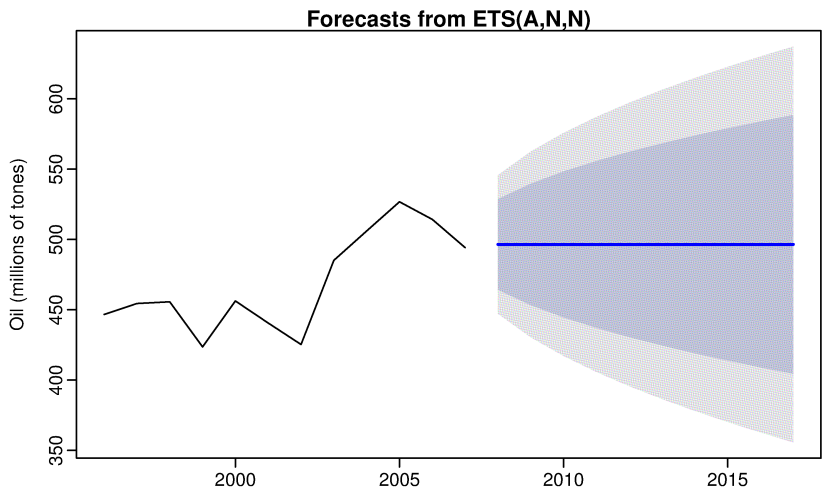
Функция ets пакета forecast в R позволяет автоматически выбрать лучшую модель экспоненциального сглаживания:

```
ets(y, model="ZZZ", damped=NULL, alpha=NULL, beta=NULL, gamma=NULL,
    phi=NULL, additive.only=FALSE, lambda=NULL,
    lower=c(rep(0.0001,3), 0.8), upper=c(rep(0.9999,3),0.98),
    opt.crit=c("lik","amse","mse","sigma","mae"), nmse=3,
    bounds=c("both","usual","admissible"), ic=c("aicc","aic","bic"),
    restrict=TRUE, allow.multiplicative.trend=FALSE,
    use.initial.values=FALSE, ...)
```

Построить прогноз можно с помощью функции forecast:

```
forecast(object, h=ifelse(frequency(object)>1,2*frequency(object),10),
    level=c(80,95), fan=FALSE, robust=FALSE, lambda=NULL,
    find.frequency=FALSE, allow.multiplicative.trend=FALSE, ...)
```

# Примеры

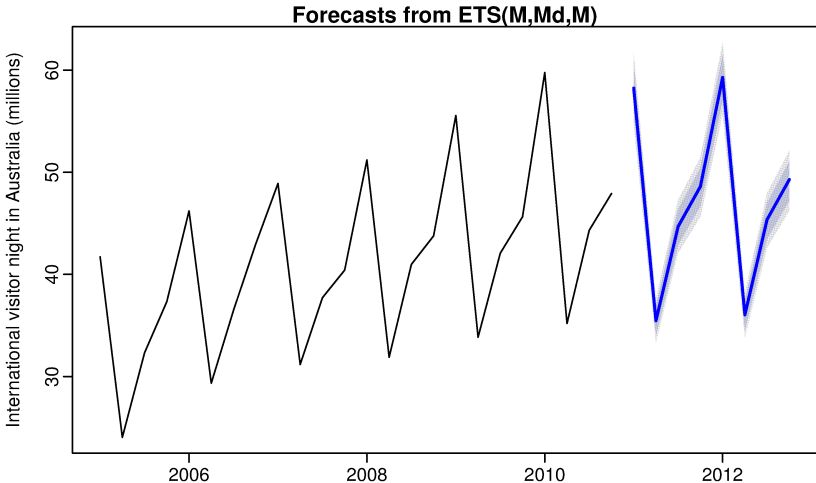


Для данных о добыче нефти в Саудовской Аравии функция ets выбирает простое экспоненциальное сглаживание.





# Примеры



Для количества ночей, проведённых туристами в Австралии, функция ets выбирает модель с мультипликативными ошибкой, сезонностью и затухающим трендом.

# ТВАТС

Метод Хольта-Уинтерса (ETS(A,A,A)):

$$\begin{aligned} \hat{y}_{t+d|t} &= l_t + db_t + s_{t-m+(d \bmod m)}, \\ l_t &= \alpha (y_t - s_{t-m}) + (1 - \alpha) (l_{t-1} + b_{t-1}), \\ b_t &= \beta (l_t - l_{t-1}) + (1 - \beta) b_{t-1}, \\ s_t &= \gamma (y_t - l_{t-1} - b_{t-1}) + (1 - \gamma) s_{t-m}. \end{aligned}$$

Если  $m$  велико, число сезонных коэффициентов можно сократить с помощью разложения Фурье:

$$\begin{aligned} s_t &= \sum_{j=1}^k s_{j,t}, \\ s_{j,t} &= s_{j,t-1} \cos \lambda_j + s_{j,t-1}^* \sin \lambda_j + \gamma_1 d_t, \\ s_{j,t}^* &= -s_{j,t-1} \sin \lambda_j + s_{j,t-1}^* \cos \lambda_j + \gamma_2 d_t, \\ \lambda_j &= \frac{2\pi j}{m}, \end{aligned}$$

$d_t$  — ARMA(p,q)-процесс,  $\gamma_1, \gamma_2$  — параметры сглаживания.

## TBATS

- реализован с автоматическим подбором параметров в пакете `forecast`, функция `tbats`;
- позволяет моделировать ряды с нецелым периодом сезонности;
- может учитывать сезонность с несколькими разными периодами (для этого ряд должен быть объектом типа `msts`, а не `ts`);
- наследует недостаток методов экспоненциального сглаживания — не может включать регрессоры.

Пример — прогнозирование недельных данных:

```
y <- ts(x, frequency=52.18)
fit <- tbats(y)
fc <- forecast(fit)
```

## regARIMA

regARIMA позволяет моделировать сложные сезонности за счёт включения фурье-гармоник по длинным сезонным периодам в регрессоры. Кроме того, можно добавить индикаторы плавающих праздников.

Пример — прогнозирование дневных данных с годовой сезонностью:

```
y <- ts(x, frequency=7)
z <- fourier(ts(x, frequency=365.25), K=5)
zf <- fourierf(ts(x, frequency=365.25), K=5, h=100)
fit <- auto.arima(y, xreg=cbind(z,holiday))
fc <- forecast(fit, xreg=cbind(zf,holidayf), h=100)
```

$K$  можно выбрать, минимизируя  $AIC$ .

Тенденция последних лет — полностью сводить задачу массового прогнозирования рядов к регрессии. Пример:

[https://yadi.sk/i/qMEJcs7\\_zfyU8](https://yadi.sk/i/qMEJcs7_zfyU8)

# Модель Хольта = линейный тренд

Линейный тренд без сезонных эффектов:

$$\hat{y}_{t+d} = l_t + b_t d,$$

где  $l_t, b_t$  — адаптивные коэффициенты линейного тренда

Рекуррентная формула:

$$l_t := \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1}) = \hat{y}_t + \alpha e_t;$$

$$b_t := \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} = b_{t-1} + \alpha\beta e_t.$$

Доказательство

$$b_t := \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} = b_{t-1} +$$

$$+ b_{t-1} + \beta(\underbrace{\hat{y}_t - l_{t-1} - b_{t-1}}_{\hat{y}_t} + \alpha e_t) = b_{t-1} + \alpha\beta e_t$$

# Модель Уинтерса = мультипликативная сезонность

Мультипликативная сезонность периода  $s$ :

$$\hat{y}_{t+d} = l_t \cdot s_{t-m+(d \bmod m)},$$

$s_0, \dots, s_{m-1}$  — сезонный профиль периода  $m$ .

Рекуррентная формула:

$$l_t := \alpha(y_t/s_{t-m}) + (1 - \alpha)l_{t-1} = a_{t-1} + \alpha e_t/s_{t-m};$$

$$s_t := \beta(y_t/l_t) + (1 - \beta)s_{t-m} = s_{t-m} + \beta(1 - \alpha)e_t/l_t.$$

Доказательство последнего равенства:

$$\begin{aligned} s_t &:= s_{t-m} + \beta(y_t/l_t - s_{t-m}) = s_{t-m} + \beta(y_t - s_{t-m}l_t)/l_t = \\ &= s_{t-m} + \beta(y_t - s_{t-m}(a_{t-1} + \alpha e_t/s_{t-m}))/l_t = s_{t-m} + \\ &+ \beta \left( \underbrace{y_t - s_{t-m}l_{t-1}}_{e_t} - \alpha e_t \right) / l_t \end{aligned}$$

## Модель Тейла–Вейджа

Линейный тренд с аддитивной сезонностью периода  $s$ :

$$\hat{y}_{t+d} = (l_t + b_t d) + s_{t+(d \bmod m)-m}.$$

$l_t + b_t d$  — тренд, очищенный от сезонных колебаний,

Рекуррентная формула:

$$l_t := \alpha(y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1}) = l_{t-1} + b_{t-1} + \alpha e_t;$$

$$b_t := \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} = b_{t-1} + \alpha\beta e_t;$$

$$s_t := \gamma(y_t - l_t) + (1 - \gamma)s_{t-m} = s_{t-m} + \gamma(1 - \alpha)e_t.$$

## Модель Уинтерса с линейным трендом

Мультипликативная сезонность периода  $s$  с линейным трендом:

$$\hat{y}_{t+d} = (l_t + b_t d) \cdot s_{t+(d \bmod m)-m},$$

$l_t + b_t d$  — тренд, очищенный от сезонных колебаний,  
 $s_0, \dots, s_{m-1}$  — сезонный профиль периода  $m$ .

Рекуррентная формула:

$$l_t := \alpha(y_t/s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1}) = l_{t-1} + b_{t-1} + \alpha e_t/s_{t-m};$$

$$b_t := \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1} = b_{t-1} + \alpha\beta e_t/s_{t-m};$$

$$s_t := \gamma(y_t/l_t) + (1 - \gamma)s_{t-m} = s_{t-m} + \gamma(1 - \alpha)e_t/l_t.$$



## Эквивалентность некоторым моделям ЭСС

Семейство ARIMA содержит в себе (как множество) все модели из семейства ЭС с аддитивным трендом и аддитивной сезонностью

- простое ЭСС: ARIMA( $p=0, d=1, q=1$ )

$$(1 - L)y_t = (1 - \phi_1 L)\varepsilon_t$$

$$\phi_1 = 1 - \alpha$$

Доказательство:

$$y_t - y_{t-1} = \varepsilon_t - \phi_1 \varepsilon_{t-1} = y_t - \hat{y}_t - (1 - \alpha) \cdot (y_{t-1} - \hat{y}_{t-1})$$

$$\hat{y}_t = y_{t-1} - y_{t-1} + \alpha y_{t-1} + (1 - \alpha) \cdot \hat{y}_{t-1} = \hat{y}_{t-1} + \alpha \cdot e_{t-1}$$

- модель линейного тренда (Хольта): ARIMA( $p=0, d=2, q=2$ )

$$(1 - L)^2 Y_t = (1 - \phi_1 L - \phi_2 L^2)\varepsilon_t$$

$$\phi_1 = 2 - \alpha - \alpha\beta, \quad \phi_2 = \alpha - 1$$

## Эквивалентность некоторым моделям ЭСС

- damped-trend linear exponential smoothing is the ARIMA(1,1,2) model

$$(1 - \phi B)(1 - B)Y_t = (1 - \theta_1 B - \theta_2 B^2)\epsilon_t$$

$$\theta_1 = 1 + \phi - \alpha - \alpha\beta\phi, \theta_2 = (\alpha - 1)\phi$$

$\phi$  — коэффициент затухания тренда;

- seasonal exponential smoothing is the ARIMA(0,1,p+1)(0,1,0)<sub>p</sub> model

$$(1 - B)(1 - B^p)Y_t = (1 - \theta_1 B - \theta_2 B^p - \theta_3 B^{p+1})\epsilon_t$$

$$\theta_1 = 1 - \alpha$$

$$\theta_2 = 1 - \gamma(1 - \alpha)$$

$$\theta_3 = (1 - \alpha)(\gamma - 1)$$

# Эквивалентность некоторым моделям ЭСС

- модель Уинтерса (аддитивная сезонность) эквивалентна ARIMA(0, 1, p + 1)(0, 1, 0)<sub>p</sub> model

$$(1 - B)(1 - B^p)Y_t = [1 - \sum_{i=1}^{p+1} \theta_i B^i] \epsilon_t$$

$$\theta_j = \begin{cases} 1 - \alpha - \alpha\beta & j = 1 \\ -\alpha\beta & 2 \leq j \leq p - 1 \\ 1 - \alpha\beta - \gamma(1 - \alpha) & j = p \\ (1 - \alpha)(\gamma - 1) & j = p + 1 \end{cases}$$







# Сравнение качества двух прогнозов

- $y_1, \dots, y_T$  — временной ряд,
- $\hat{y}_{1R}, \dots, \hat{y}_{1T}$  — прогноз на период  $R, \dots, T$  первым методом,
- $\hat{\epsilon}_{1R}, \dots, \hat{\epsilon}_{1T}$  — остатки первого прогноза,
- $\hat{y}_{2R}, \dots, \hat{y}_{2T}$  — прогноз на период  $R, \dots, T$  вторым методом,
- $\hat{\epsilon}_{2R}, \dots, \hat{\epsilon}_{2T}$  — остатки второго прогноза;
- $g(y_t, \hat{y}_{it})$  — произвольная функция потерь,  
 (например,  $|\hat{\epsilon}_{it}|$  или  $\hat{\epsilon}_{it}^2$ ),
- $d_t = g(y_t, \hat{y}_{1t}) - g(y_t, \hat{y}_{2t})$ .

- $H_0$ : среднее  $d_t = 0$ ,
- $H_1$ : среднее  $d_t < \neq > 0$ .

Критерий знаковых рангов Уилкоксона:

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









# Критерий reality check Уайта

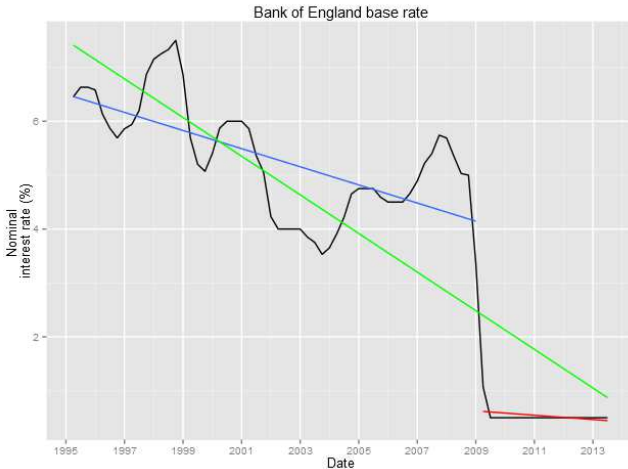
**Пример** (Sullivan, Timmermann, White, 1999, 2001): к ряду промышленного индекса Доу-Джонса с 1 января 1897 по 30 июня 1998 (27447 отсчётов) было применено большое количество моделей — 7846 моделей технического анализа и 9452 календарных. В качестве эталона рассматривалась стратегия долгосрочного инвестирования. Критерий качества — средний ожидаемый доход по всем инвестициям.

Критерий Уайта показал, что лучший метод технического анализа выигрывает у эталона, в то время как лучший календарный метод, по всей видимости, переобучен — он существенно лучше эталона по критерию Диболда-Мариано, но не лучше по критерию Уайта.

# Модификация Романо-Вольфа

Построив на основе критерия Уайта нисходящую процедуру, можно найти все методы, дающие прогноз лучше эталона, асимптотически контролируя при этом FWER (Romano, Wolf, 2005).

# Структурное изменение модели



Как проверить, не нужно ли настраивать разные модели на разных участках ряда?

# Критерий Чоу

Пусть  $\hat{\varepsilon}_t, k$  — остатки и число параметров общей модели,  $\hat{\varepsilon}_{1t}, k_1$  — модели на первом участке,  $\hat{\varepsilon}_{2t}, k_2$  — на втором.

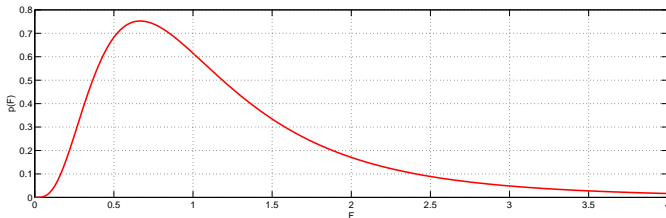
нулевая гипотеза:  $H_0$ : структура модели стабильна

альтернатива:  $H_1$ :  $H_0$  неверна

статистика:  $RSS = \sum_{t=T_1}^{T_3} \hat{\varepsilon}_t^2, \quad RSS_1 = \sum_{t=T_1}^{T_2} \hat{\varepsilon}_{1t}^2, \quad RSS_2 = \sum_{t=T_2}^{T_3} \hat{\varepsilon}_{2t}^2,$

$$F = \frac{(RSS - RSS_1 - RSS_2) / (k_1 + k_2 - k)}{(RSS_1 + RSS_2) / (n - k_1 - k_2)}, \quad n = T_3 - T_1 + 1$$

нулевое распределение:  $F(k_1 + k_2 - k, n - k_1 - k_2)$



# Критерий Чоу

Применение в R:

```
n <- length(y)
m <- ets(y)
k <- length(m$par)
rss <- sum(m$residuals^2)

m1 <- ets(y[1:24])
k1 <- length(m1$par)
rss1 <- sum(m1$residuals^2)

m2 <- ets(y[25:n])
k2 <- (m2$aic + 2*m2$loglik)/2
rss2 <- sum(m2$residuals^2)

f <- ((rss-rss1-rss2)/(k1+k2-k)) / ((rss1-rss2)/(n-k1-k2))
pf(f, k1+k2-k, n-k1-k2)
```

# Требования к решению ДЗ прогнозирования временных рядов

- визуализация данных, анализ распределения признака (оценка необходимости трансформации), оценка наличия выбросов, преобразования для снятия календарных эффектов;
- настройка модели ARIMA: выбор модели руками на основании анализа ACF/PACF, поиск наилучшей модели в окрестности, анализ остатков; автоматический подбор модели, проверка её соответствия особенностям ряда, анализ остатков;
- настройка модели экспоненциального сглаживания: автоматический подбор модели, проверка её соответствия особенностям ряда, корректировка, анализ остатков;
- визуальный анализ, при необходимости — формальная проверка наличия структурных изменений в моделях;
- сравнение и выбор лучшей модели по критерию Диболда-Мариано;
- выводы.





# Адаптивная селективная модель

Пусть имеется  $k$  моделей прогнозирования,  
 $\hat{y}_{j,t+d}$  — прогноз  $j$ -й модели на момент  $t + d$ ,  
 $\varepsilon_{jt} = y_t - \hat{y}_{jt}$  — ошибка прогноза в момент  $t$ ,  
 $\tilde{\varepsilon}_{jt} = \sum_{l=R}^{t-d} \delta^{t-d-l} |\varepsilon_{jl}|$  — экспоненциально сглаженная ошибка,  
 $\delta$  — параметр сглаживания ошибки.

Лучшая модель в момент времени  $t$ :

$$j_t^* = \operatorname{argmin}_{j=1, \dots, k} \tilde{\varepsilon}_{jt}.$$

Адаптивный селективный прогноз:

$$\hat{y}_{j,t+d} := \hat{y}_{j_t^*, t+d}.$$

# Адаптивная композиция моделей

Пусть имеется  $k$  моделей прогнозирования,  
 $\hat{y}_{j,t+d}$  — прогноз  $j$ -й модели на момент  $t+d$ ,  
 $\varepsilon_{jt} = y_t - \hat{y}_{jt}$  — ошибка прогноза в момент  $t$ ,  
 $\tilde{\varepsilon}_{jt} = \sum_{l=R}^{t-d} \gamma^{t-d-l} |\varepsilon_{jl}|$  — экспоненциально сглаженная ошибка.

Линейная комбинация моделей:

$$\hat{y}_{t+d} = \sum_{j=1}^k w_{jt} \hat{y}_{j,t+d}, \quad \sum_{j=1}^k w_{jt} = 1, \quad \forall t.$$

Адаптивный подбор весов:

$$w_{jt} = \frac{(\tilde{\varepsilon}_{jt})^{-1}}{\sum_{s=1}^k (\tilde{\varepsilon}_{st})^{-1}}.$$

## Другие способы комбинирования

- регрессия ряда на несколько первых главных компонент прогнозов в скользящем окне;
- МНК с байесовским регуляризатором на веса (априорное распределение — равномерное);
- экспоненциально сглаживаются не модули, а квадраты ошибок;
- вместо экспоненциального сглаживания используется усреднение в окне;
- ...

# Online Learning

## Definition

**Game  $G$**  comprises  $\langle \Omega, \Gamma, \lambda \rangle$  where  $\Omega$  is a set of outcomes,  $\Gamma$  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  $\hat{y}_{t+1} \in \Gamma$ ;
- 2 obtain outcome  $y_{t+1} \in \Omega$ ;
- 3 calculate loss  $\lambda(y_{t+1}, \hat{y}_{t+1})$ .

## Definition (loss process)

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

# Online Learning

Простейшие примеры игр:

- двоичные (бинарные) игры  $\Omega = \{0, 1\}$ ,  $\Gamma = [0, 1]$ ;
- квадратичная игра  $\lambda(y, \hat{y}) = (y - \hat{y})^2$ ;
- абсолютная игра  $\lambda(y, \hat{y}) = |y - \hat{y}|$ ;
- логарифмическая игра

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

- простая предсказательная игра  $\Omega = \Gamma = \{0, 1\}$ ,

$$\lambda(y, \hat{y}) = \begin{cases} 0, & y = \hat{y}; \\ 1, & x \neq \hat{y}. \end{cases}$$

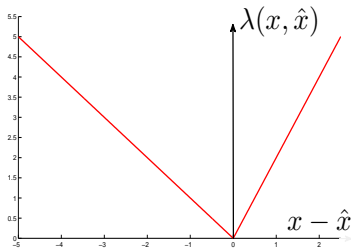
# Online Learning

При прогнозировании временных рядов:

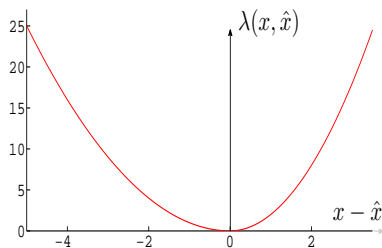
● Game  $G = \langle [Y_1, Y_2], [Y_1, Y_2], \lambda \rangle$  where

$$\lambda(x, \hat{x}) = \begin{cases} k_1 \cdot |x - \hat{x}|, & x - \hat{x} < 0, \\ k_2 \cdot |x - \hat{x}|, & x - \hat{x} \geq 0, \end{cases} \quad \left| \quad \begin{array}{l} \text{where } k_1 > 0, k_2 > 0 \end{array} \right.$$

$$\lambda(x, \hat{x}) = \begin{cases} k_1 \cdot (x - \hat{x})^2, & x - \hat{x} < 0, \\ k_2 \cdot (x - \hat{x})^2, & x - \hat{x} \geq 0, \end{cases} \quad \left| \quad \begin{array}{l} \text{where } k_1 > 0, k_2 > 0 \end{array} \right.$$



linear loss function



square loss function

# Потери важнее прогнозов

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

❶ Задача 1

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

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

- *Ответ: ???*

❷ Задача 2

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

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

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

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



Идея агрегирующего алгоритма В. Вовка

- 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

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

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

# Идея агрегирующего алгоритма В. Вовка

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 алгоритма В. Вовка

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

Итоговая композиция  $AA$  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)

Ответ композиции — это решающее правило над функцией смешивания:

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

$$S_2(g) = \frac{g(Y_1) - g(Y_2)}{2(Y_2 - Y_1)} + \frac{Y_1 + Y_2}{2}$$

# Оценки процесса потерь

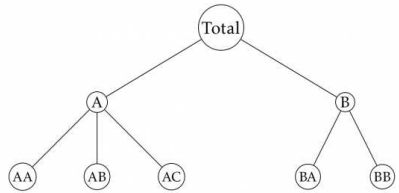
- Базовые алгоритмы  $\{A^1, \dots, A^N\}$ .
- Возьмём подходящие  $\beta$  and  $S(g)$
- Получаем композицию  $AA$ .
- алгоритмическая сложность  $O(NT)$ .

Если правильно подобрать  $\beta$  и  $S(g)$ , тогда

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

# Иерархия рядов

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



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

# Подходы к прогнозированию

**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:

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

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

# Подходы к прогнозированию

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

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

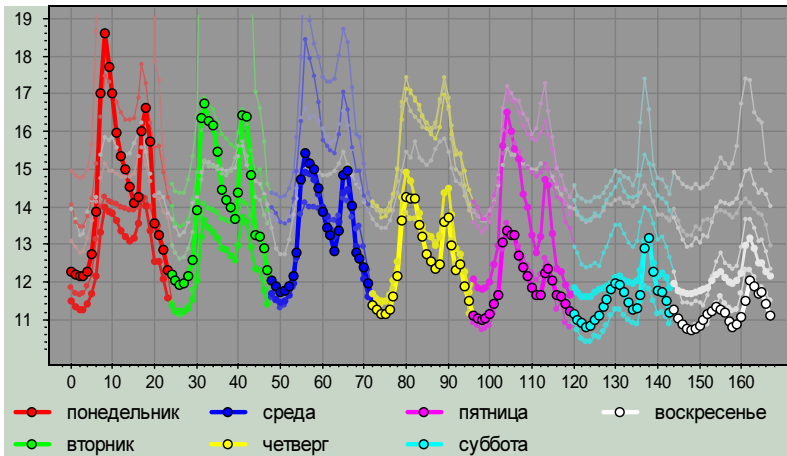
$$\varepsilon_h \approx S\varepsilon_{K,h} \Rightarrow \tilde{\mathbf{y}}_h = S \left( S^T S \right)^{-1} \hat{\mathbf{y}}_h.$$

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.

# Прогнозирование объёмов потребления электроэнергии

Почасовые цены электроэнергии на бирже NordPool, 2000г.



Особенности задачи: три вложенные сезонности, скачки



# Линейная модель авторегрессии

В роли признаков —  $n$  предыдущих наблюдений ряда:

$$\hat{y}_{t+1}(w) = \sum_{j=1}^n w_j y_{t-j+1}, \quad w \in \mathbf{R}^n$$

В роли объектов  $\ell = t - n + 1$  моментов в истории ряда:

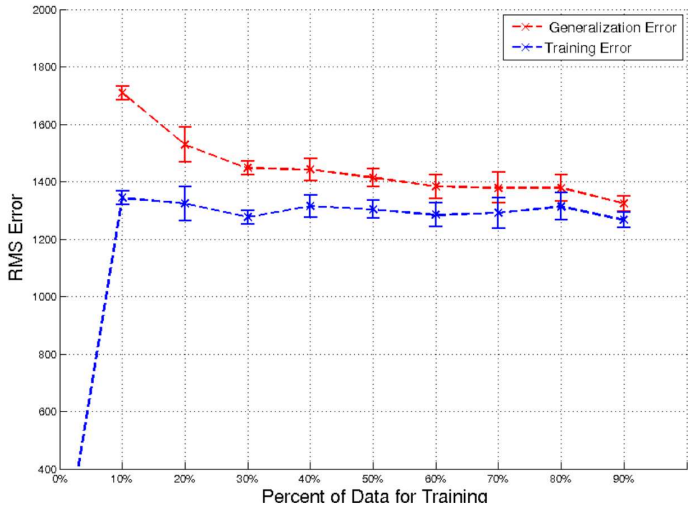
$$F_{\ell \times n} = \begin{pmatrix} y_t & y_{t-1} & y_{t-2} & \dots & y_{t-n+1} \\ y_{t-1} & y_{t-2} & y_{t-3} & \dots & y_{t-n} \\ y_{t-2} & y_{t-3} & y_{t-4} & \dots & y_{t-n-1} \\ \dots & \dots & \dots & \dots & \dots \\ y_n & y_{n-1} & y_{n-2} & \dots & y_1 \end{pmatrix}, \quad y_{\ell \times 1}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

Функционал квадрата ошибки:

$$Q_t(w, X^\ell) = \sum_{i=n+1}^{t+1} (\hat{y}_i(w) - y_i)^2 = \|Fw - y\|^2 \rightarrow \min_w$$

# Когда нужно что-то ещё более сложное?

Чем больше обучающая выборка, тем точнее результат



# Точность предсказаний объёмов электроэнергии

Результаты для различных моделей

Таблица: Результаты прогнозирования для различных моделей обучения

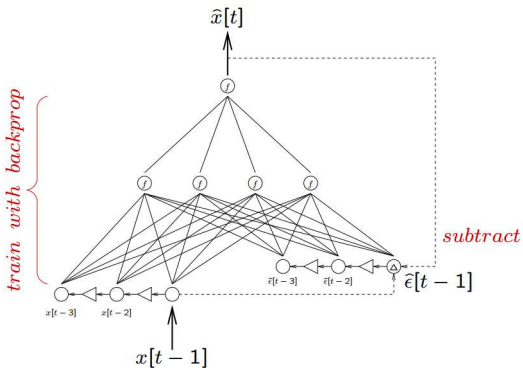
Метод обучения	RMSE	% RMSE
Kernelized Regression	1 540	8.3%
NN	1250	6.7%
Deep Forward NN	1130	5.9%
Deep Recurent NN	<b>530</b>	2.8%

---

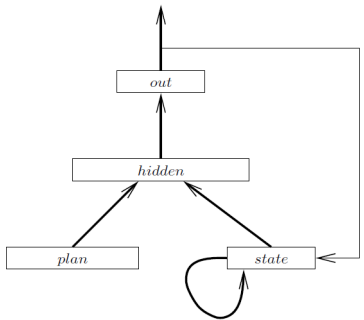
*Enzo Buseti et al.* Deep Learning for Time Series Modeling. CS 229 final Project Report, 2012.

# Модель нелинейной ARMA в NN

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



# Memory term для NN



Элемент памяти:  $\bar{x}_i(t) = \sum_{\tau=1}^t c_{t-\tau} \cdot x_{\tau}$

Веса для элемента памяти

- дельта-функция  $c_t = \delta_d(t)$
- экспоненциальные:  $c_t = (1 - \alpha)\alpha^t$

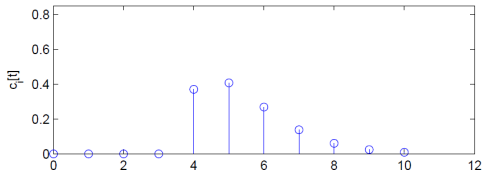
# Веса memory term

Варианты весов:

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

при  $d = 0$  получаются экспоненциальные веса;  
 при  $\alpha \rightarrow 0$  приходим к дельта функции  $\delta_d(t)$

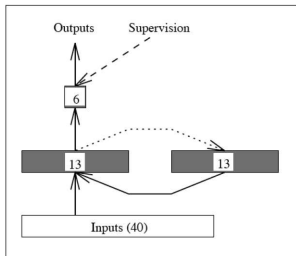
Пример для  $d = 4, \alpha = 0.2$





# Elman NET

Output:  $\{\hat{x}[t], \dots, \hat{x}[t + 5]\}$



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

Метод обучения	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



## Резюме по сложным временным рядам

- для временных рядов со сложной структурой используются
  - сложные регрессионные подходы (отбор признаков, фурье разложения, ядерные преобразования данных и др.);
  - (глубинные) нейронные сети;
- (глубинные) нейронные сети обобщают базовые подходы к прогнозированию ARMA, ARIMA и др.
- усложнение модели прогнозирования должно значимо улучшать точность предсказаний

## Резюме в конце лекции

- простые адаптивные алгоритмы моделируют различные особенности рядов
- адаптивные методы хорошо работают для большой доли рядов, в том числе на коротких
- простота адаптивных методов компенсируется комбинированием их прогнозов
- композиции алгоритмов прогнозирования часто не дают улучшения точности прогнозирования
- сложные алгоритмы (регрессионные, RNN) уместны для временных рядов со сложной структурой (несколько видов сезонностей, высоко модальные временные ряды)
- для обучения сложных алгоритмов требуется большая история

## Литература

- модели экспоненциального сглаживания (exponential smoothing) — Hyndman;
- критерий Диболда-Мариано (Diebold-Mariano) и его модификация для коротких рядов — Harvey;
- reality check Уайта (White) и нисходящая процедура на его основе — Romano;
- критерий Чоу (Chow test) — Chow;
- адаптивная селекция и композиция прогнозирующих алгоритмов — Воронцов, Genre;
- агрегирующий алгоритм Вовка — Калнишкан;
- иерархические ряды — Hyndman, 9.4;
- TBATS — De Livera.

Воронцов К.В., Егорова Е.В. (2006). *Динамически адаптируемые композиции алгоритмов прогнозирования*. Искусственный Интеллект, Донецк, 2, 277–280.

Калнишкан Ю.А. *Введение в методы конкурентного предсказания*.  
[http://www.clrc.rhul.ac.uk/people/yura/concurrent\\_rus.pdf](http://www.clrc.rhul.ac.uk/people/yura/concurrent_rus.pdf)

Стенина М.М., Стрижов В.В. (2015). *Согласование прогнозов при решении задач прогнозирования иерархических временных рядов*. Информатика и ее применения, 9(2), 77–89.

## Литература

Chow G.C. (1960). *Tests of Equality Between Sets of Coefficients in Two Linear Regressions*. *Econometrica*, 28(3), 591–605.

De Livera A.M., Hyndman R.J., Snyder R.D. (2011). *Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing*. *Journal of the American Statistical Association*, 106(496), 1513–1527.

Genre V., Kenny G., Meyler A., Timmermann A. (2013). *Combining expert forecasts: Can anything beat the simple average?* *International Journal of Forecasting*, 29(1), 108–121.

Harvey D., Leybourne S., Newbold P. (1997). *Testing the equality of prediction mean squared errors*. *International Journal of Forecasting*, 13, 281–291.

Hyndman R.J., Athanasopoulos G. *Forecasting: principles and practice*, 2016.  
<https://www.otexts.org/book/fpp>

Romano J., Wolf M. (2005). *Stepwise multiple testing as formalized data snooping*. *Econometrica*, 73(4), 1237–1282.

Sullivan R., Timmermann A., White H. (2003). *Forecast evaluation with shared data sets*. *International Journal of Forecasting*, 19(2), 217–227.