

# Modelovanie časových radov z monitorovacieho systému Hawkular

Bc. Pavol Loffay<sup>1</sup>

26. novembra 2015

**Abstrakt:** Práca spracováva predikciu časových radov prevzatých z monitorovacieho systému Hawkular<sup>a</sup>. Tento systém dokáže monitorovať Java aplikácie, alebo fyzický hardvér na ktorom je spustený. Z množiny pozorovaných metrík boli vybraté nasledujúce: vyťaženie Java hromady (heap), miesto na disku a počet voľných databázových spojení. Tieto časové rady som analyzoval a cieľom bolo zostaviť model, ktorý najlepšie popisoval priebeh danej časovej rady. V práci som postupoval podľa Box – Jenkinsovej metodológie.

**Kľúčové slová:** Časová rada; ARIMA; Hawkular; ACF

**JEL klasifikácia:** C53

---

<sup>a</sup>Dostupné na <http://www.hawkular.org>

## 1 Úvod

Monitorovanie dôležitých business aplikácií, ako sú napríklad bankové systémy alebo rôzne servery na ktorých bežia služby ktoré sú využívané 24/7 je veľmi dôležité. Väčšina monitorovacích systémov dokáže upozorniť administrátora na vyťaženie pri prekročení hraničnej hodnoty. V monitorovacom systéme Hawkular je možné zapnúť predikciu, takže administrátor bude upozornený vopred ak by náhodou malo dôjsť k prekročeniu spomenutej hraničnej hodnoty. Používané modely časových rád v systéme Hawkular sú varianty exponenciálneho vyrovnania. Modely *ARIMA* nemohli byť použité z dôvodu nestacionarity dát a náročnosti na výpočet – systém je schopný analyzovať aj niekoľko tisíc model naraz.

V tejto práci sa zameriam na konštrukciu optimálneho *ARIMA* modelu, ktorý následne porovnáam s exponenciálnym vyrovnaním konkrétne Holtovou metódou s lineárnym trendom.

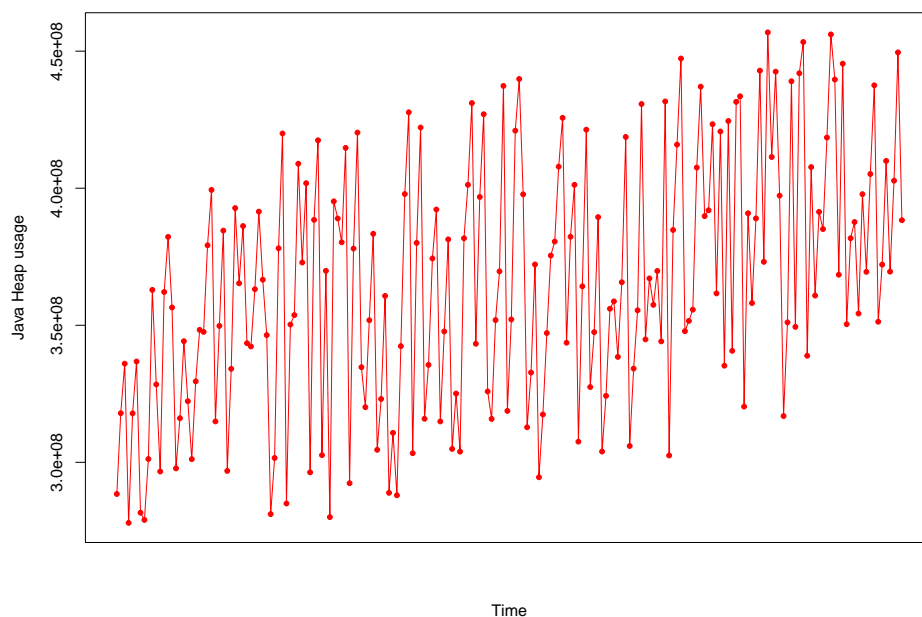
## 2 Identifikácia ARIMA modelu

V tejto kapitole budeme analyticky identifikovať najvhodnejší *ARIMA* model, ktorý následne porovnáme s výsledkom funkcie *auto.arima*.

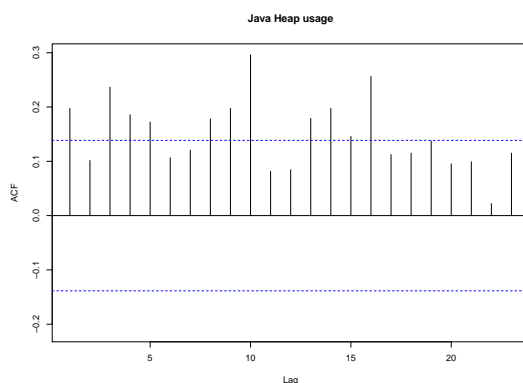
Začneme vykreslením časovej rady. Na obrázku 1 je možné vidieť, že časová rada obsahuje mierny lineárny trend, čo indikuje nestacionaritu. Vykresíme autokorelačnú (zkrátene *ACF*) a parciálnu autokorelačnú funkciu (zkrátene *PACF*).

---

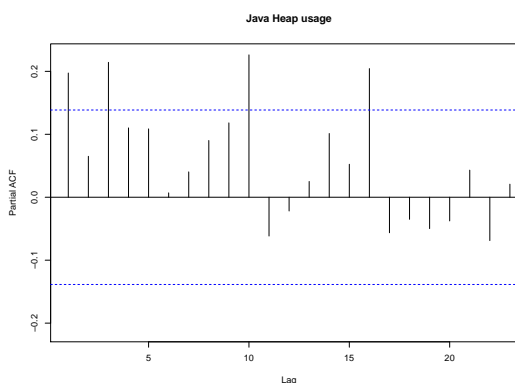
<sup>1</sup>Masarykova univerzita, Fakulta informatiky, obor: Service Science Management Engineering, p.loffay@mail.muni.cz



Obr. 1: Vyťaženosť Java Heap-u v čase.



(a) Autokorelačná funkcia.



(b) Parciálna autokorelačná funkcia.

Z grafu autokorelačnej funkcie 2a je vidieť že hodnoty sú kladné ale relatívne blízke nuly, stým že významne neklesajú. Na grafe *ACF* taktiež nie sú prítomné periodicky posunuté vysoké hodnoty, ktoré by indikovali sezónnosť. Keďže hodnoty *ACF* so zvyšujúcim spozdením neklesajú k nule rozhodli sme sa urobiť *ADF* test na testovanie stacionarity.

```
> adfTest(as.numeric(b$avg), lags = 0, type="nc")
Title:
Augmented Dickey-Fuller Test
Test Results:
PARAMETER:
Lag Order: 0
STATISTIC:
Dickey-Fuller: -0.976
P VALUE:
0.3042
```

Nulovú hypotézu *ADF* testu o prítomnosti jednotkového koreňa, na hladine významnosti 0.05 nezamietame. Takže naša časová rada je nestacionárna. Pre overenie skúsime *KPSS* test, ktorého nulová hypotéza je, že časová rada je stacionárna.

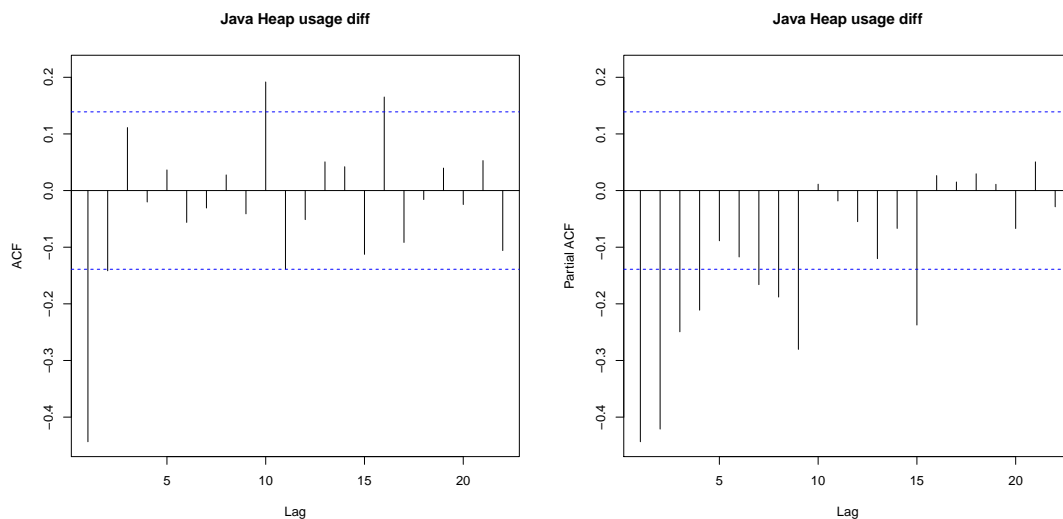
```
> kpss.test(as.numeric(b$avg))
KPSS Test for Level Stationarity
data: as.numeric(b$avg)
KPSS Level = 2.5916, Truncation lag parameter = 3, p-value = 0.01
```

Na hladine významnosti by sme nulovú hypotézu zamietli. Oba testy nám ukázali, že časová rada nie je stacionárna. Následne môžeme pomocou *KPSS* testu testovať, či je daná časová rada trend stacionárna.

```
> kpss.test(as.numeric(b$avg), null='Trend')
KPSS Test for Trend Stationarity
data: as.numeric(b$avg)
KPSS Trend = 0.073128, Truncation lag parameter = 3, p-value = 0.1
> ndiffs(as.numeric(b$avg))
[1] 1
> diff = diff(as.numeric(b$avg))
```

Z výstupu je jasné, že rada je trend stacionárna takže nulovú hypotézu na hladine 0.05 nezamietame.

Časovú radu stacionarizujeme diferenciováním a pokračujeme vykreslením *ACF* a *PAC* upravenej časovej rady. Z grafu parciálnej autokorelačnej funkcie 3 môžeme usúdiť, že do úvahy by spadal model *AR(15)* alebo *MA(1)*. Keďže *ACF* je možné obmedziť krivkou U, tak je lepšie vybrať model *MA(1)* [2].



Obr. 3: ACF a PACF diferencovanej časovej rady.

Alternatívny spôsob voľby modelu je pomocou informačných kritérií. Tento spôsob je vhodný pre plne automatizované spracovanie [2] napríklad v ekonometrických softvéroch. K identifikácii modelu *ARMA(p, q)* sa priktupuje ako k minimalizácii funkcie 2.1

$$(\hat{p}, \hat{q}) = \arg \min_{(k, l)} A(k, l) \quad (2.1)$$

*A* je vhodné kritérium pre ktorého konštrukciu musíme odhadnúť model *ARMA(k, l)*. Pri minimalizácii postupujeme postupne inkrementujeme obidva parametre *k, l*. V tejto práci sme zvolili Akaikeho informačné kritérium:

$$A(k, l) = AIC(k, l) = \ln \hat{\sigma}_{k, l}^2 + \frac{2(k + l + 1)}{n} \quad (2.2)$$

Z rovnice 2.2 je zrejmé, že kritérium penalizuje veľké rády  $k$  a  $l$ .  $\hat{\sigma}_{k,l}^2$  je smerodajná odchylka reziduí modelu. Poďme si vypísať niekoľko kandidátov *ARIMA* modelov pomocou príkazu `auto.arima()`.

```
> auto.arima(as.numeric(df$avg), approximation=FALSE, trace=TRUE, ic='aic', allowdrift=FALSE)
ARIMA(2,1,2) : 7556.055
ARIMA(0,1,0) : 7696.058
ARIMA(1,1,0) : 7651.407
ARIMA(0,1,1) : 7557.045
ARIMA(1,1,2) : 7560.654
ARIMA(3,1,2) : 7557.714
ARIMA(2,1,1) : 7553.937
ARIMA(1,1,1) : 7557.937
ARIMA(3,1,1) : 7555.879
ARIMA(2,1,0) : 7613.902

Best model: ARIMA(2,1,1)
Series: as.numeric(df$avg)
ARIMA(2,1,1)
Coefficients:
      ar1      ar2      ma1
s.e. -0.0985 -0.1774 -0.9441
      0.0716  0.0716  0.0199
```

Ako je vidieť funkcia zvolila model *ARIMA(2,1,1)* ktorého *AIC* kritérium bolo najnižšie. Odhadnutý model môžeme zapísať v tvare:

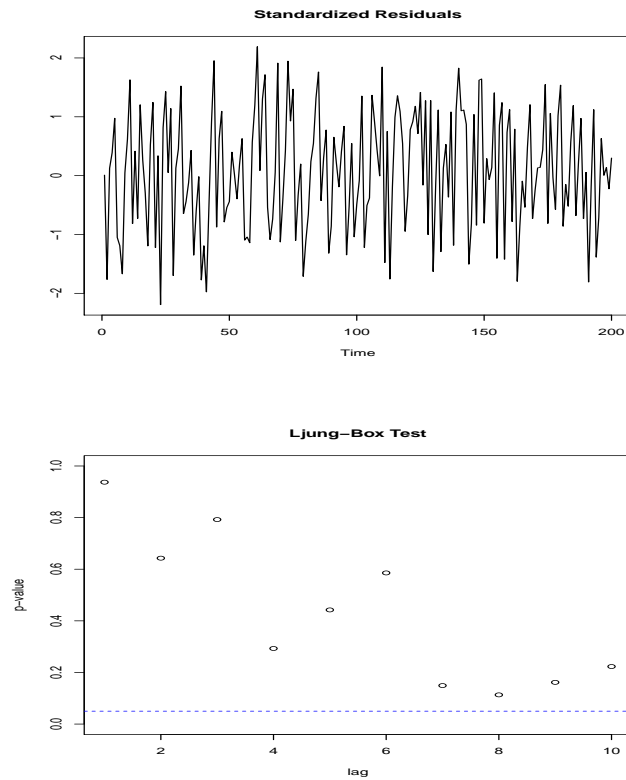
$$Y_t = -0.985Y_{t-1} - 0.1774Y_{t-2} - 0.9441\varepsilon_{t-1} + \varepsilon_t \quad (2.3)$$

Ukázali sme, že je možné konštruovať *ARIMA* model aj analytickým spôsobom. Chcel by som avšak poznamenať, že voľbu modelu je lepšie prenechať overenému štatistickému softvéru.

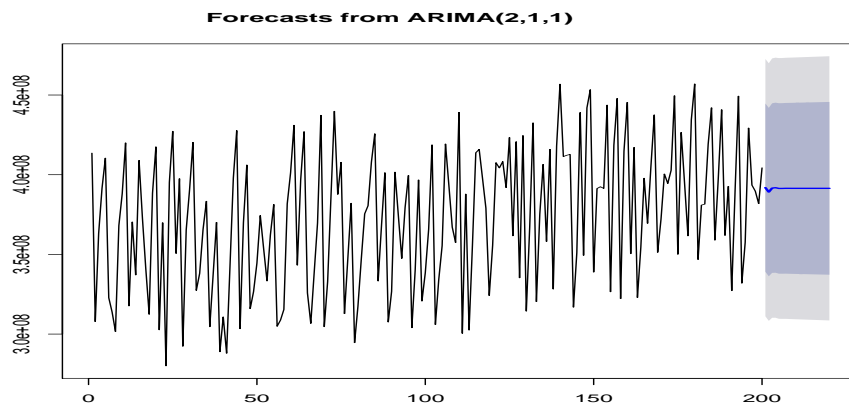
Po úspešnom odhade rádu modelu je by sme chceli zmieniť ako sa počítajú jednotlivé koeficienty. Pre *AR* model platí, že sa dajú vypočítať pomocou *OLS* alebo Yule – Walkerových rovníc [1]. Výpočet koeficientov *MA* modelu je zložitejší a je možný pomocou rekurzívnej Levison-Durbin metódy. Odhadom presných parametrov modelu sa v tejto práci ďalej nebudeme zaoberať.

Na záver sa pozrieme na rezíduá odhadnutého modelu. Ak sme postupovali správne rezíduá by mali pripomínať biely šum a nemali by byť korelované (inakšie by sme ich vedeli modelovať). Toto tvrdenie si overíme Ljung – Box testom, ktorého nulová hypotéza hovorí o tom, že dáta sú nezávislé distribuované. Z grafu 4 môžeme prehlásiť, že nulovú hypotézu na hladine významnosti 0.05 nezamietame.

Na nasledujúcich grafoch si vykresíme rezíduá modelu, ich autokorelačnú funkciu a p-hodnoty pre rôzne opozdenia Ljung – Box testu.



Obr. 4: Rezíduá odhadnutého *ARIMA* modelu.



Obr. 5: *ARIMA*, predikcia na 20 krokov do predu.

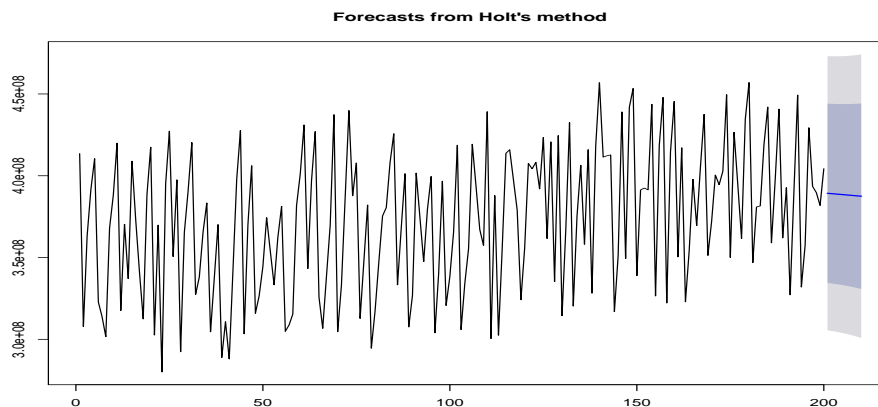
### 3 Exponenciálne vyrovňavanie – Holtova metóda

Zobecnením dvojitého exponenciálneho vyrovňavania je takzvaná Holtova metóda. Táto metóda používa dve vyrovňavacie konštanty:  $\alpha$  pre vyrovnanie úrovne  $l_t$  a  $\beta$  pre vyrovnanie smernice  $b_t$  (lineárneho trendu). Výhoda tejto metódy je možnosť použitia v prúdovom spracovaní, kde nie je možné získať staré hodnoty časovej rady. Rovnice majú tvar 3.1. Parametre  $\alpha, \beta$  patria do

intervalu  $\alpha, \beta \in (0, 1)$ . Pre exponenciálne vyrovnanie platí, čím je hodnota parametra  $\alpha$  menšia, tak väčšia váha je daná pozorovaniam z vzdialenej minulosti.

$$\begin{aligned}\hat{y}_{t+h} &= l_t + hb_t \\ l_t &= \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1}) \\ b_t &= \beta(l_t - l_{t-1}) + (1 - \beta)b_{t-1}\end{aligned}\tag{3.1}$$

V jazyku *R* použijeme funkciu *holt*, ktorá počíta predikciu na  $k$  krokov do predu. Súčasťou funkcie je aj výpočet parametrov  $\alpha$  a  $\beta$ .



Obr. 6: Holt, predikcia na 20 krokov do predu.

## 4 Záver

Na záver by som porovnal kvadratickú sumu chýb (*SSE*) oboch modelov. Pre model *ARIMA* chyba vyšla  $3.377059e + 17$  pre Holtovu metódu  $3.657135e + 17$ . Vidíme, že rozdiel nieje významne iný. Keď opticky porovnáme grafy 5, 6 vidíme, že exponenciálne vyrovnanie strmšie klesá

## Podakovanie

Na záver by som chcel poďakovať Ing. Danielovi Němcovi, Ph.D. za návrh na vypracovanie tejto témy a za veľmi príjemné a užitočné konzultácie. Ďalej by som chcel poďakovať Ester Železnáčkovej za gramatickú korektúru textu.

## Literatúra

- [1] Brockwell, P.; Davis, R.: *Time Series: Theory and Methods*. Praha: Springer-Verlag New York, 2009, ISBN 978-0-387-97429-3.

[2] Tomáš, C.: *Finanční ekonometrie*. Ekopress, 2008, ISBN 978-80-86929-43-9.

## A Prílohy

- Skript v jazyku R
- Dáta analyzovanej časovej rady
- Zdrojový text tejto správy v  $\text{\LaTeX}$  – e