



UNIVERZITET U NOVOM SADU
PRIRODNO-MATEMATIČKI
FAKULTET
DEPARTMAN ZA MATEMATIKU I
INFORMATIKU



Dejan Vučenović

Primena metoda distribuirane optimizacije u autoregresivnim modelima

-MASTER RAD-

Mentor

dr Nataša Krklec Jerinkić

Novi Sad, 2020.

Sadržaj

Predgovor.....	5
1 Osnovni pojmovi teorije verovatnoće i stohastički procesi	6
2 Analiza vremenskih serija	10
2.1 Transformacije vremenskih serija.....	11
2.2 Test normalnosti	12
2.3 Pojam stacionarnosti i autokorelacije.....	13
2.4 Autoregresivni modeli (AR)	15
<i>2.4.1 AR(1) model.....</i>	<i>16</i>
<i>2.4.2 AR(2) model.....</i>	<i>17</i>
<i>2.4.3 AR(p) model.....</i>	<i>18</i>
<i>2.4.4 Određivanje reda modela</i>	<i>18</i>
<i>2.4.5 Ocena parametara</i>	<i>19</i>
<i>2.4.6 Adekvatnost AR modela</i>	<i>20</i>
<i>2.4.7 Predikcije</i>	<i>21</i>
2.5 Distribuirani najmanji kvadrati	23
<i>2.5.1 Model i algoritam</i>	<i>23</i>
<i>2.5.2 Algoritam 1.</i>	<i>25</i>
<i>2.5.3 Algoritam 2.</i>	<i>26</i>
<i>2.5.4 Algoritam 3.</i>	<i>27</i>
3 Implemetacija	31
4. Rezultati testiranja.....	32
Zaključak	36
Dodatak – Matlab kodovi	37

Implementacija Algoritma 1	37
Implementacija Algoritma 2	37
Implementacija Algoritma 3	39
Implementacija algoritma za automatsku analizu vremenske serije	43
Implementacija pomoćnih algoritama	45
Literatura.....	49
Biografija	51

Predgovor

”Nikad ne razmišljam o
budućnosti, ona dolazi uskoro.”

Albert Einstein

Kako je u današnje vreme sve lakše doći do podataka koji su svuda oko nas, razvile su se oblasti matematike koje na osnovu podataka dolaze do zaključaka koji imaju uticaj na naše buduće odluke. Jedna od tih oblasti je analiza vremenskih serija.

Jedan od glavnih zadataka analize vremenskih serija je formiranje predikcija. Parametri modela koji se koriste za formiranje pomenutih predikcija dobijaju se na osnovu istorijskih podataka i najčešće se za njihovu ocenu koriste optimizacione metode. Problem optimizacije koji se javlja može se posmatrati kao problem distribuirane optimizacije i često je u formi najmanjih kvadrata.

Cilj ovog rada je da metode specificirane za rešavanje problema distribuirane optimizacije primeni na autoregresivne modele koji se koriste u analizi vremenskih serija. Time bi se ubrzao proces nalaženja parametara, a samim tim i formiranja predikcija, što je veoma značajno u primenama koje zahtevaju odgovore u „realnom vremenu“.

Rad ćemo započeti navođenjem osnovnih pojmova vezanih za vremenske serije, što uključuje i osnovne rezultate teorije verovatnoće i stohastičke analize. Zatim ćemo objasniti šta predstavlja autoregresivni model i kako dolazimo do problema distribuirane optimizacije. Nakon toga ćemo razmotriti metod za rešavanje pomenutog problema, kao i njegovu implementaciju. Na kraju ćemo prikazati implementaciju pomenutih metoda u programskom paketu Matlab, čiji je rezultat program koji će na osnovu date vremenske serije automatski izvršiti analizu serije, odrediti predikcije i uporediti modele na osnovu dobijenih rezultata.

Ovom prilikom bih se zahvalio svim profesorima i asistentima sa kojima sam sarađivao tokom svojih studija, na ukazanom strpljenju i znanju. Posebno bih želeo da se zahvalim svojoj profesorici i mentoru, *Dr Nataši Krklec Jerinkić*, za sve sugestije i svu pomoć koju mi je pružila pri izradi ovog rada, kao i za svo znanje koje sam stekao pored nje.

1 Osnovni pojmovi teorije verovatnoće i stohastički procesi

Teorija verovatnoće je polazna osnova za statistiku, jednu od oblasti matematike sa najviše primena, koja uključuje kvantitativnu analizu velikih skupova podataka. Ona se bavi izučavanjem zakonitosti raznih slučajnih procesa i događaja. Ovde ćemo objasniti neke osnovne pojmove iz teorije verovatnoće koji će nam služiti za dalji rad [10].

Definicija 1.1 *Slučajan događaj A (ili samo događaj A) je podskup skupa elementarnih događaja Ω . On se sastoji od onih elementarnih događaja ω koji imaju svojstvo kojim se događaj A definiše.*

Aksioma 1.1 (Aksioma σ -polja)

Podskup F partitivnog skupa $P(\Omega)$ je σ -polje (σ -algebra) nad Ω ako važe uslovi:

1. $\Omega \in F$,
2. ako $A \in F$, onda $\bar{A} \in F$, gde je \bar{A} komplement skupa A ,
3. ako $\{A_i\}_{i \in \mathbb{N}} \subseteq F$, onda $\bigcup_{i=1}^{\infty} A_i \in F$

Definicija 1.2 *Borelovo σ -polje $B = B(\mathbb{R})$ je σ -polje definisano nad skupom realnih brojeva. Formira se pomoću familije poluotvorenih intervala $[a,b)$, $a, b \in \mathbb{R}$ i sadrži sve skupove koji se dobijaju kao konačne ili prebrojive unije ili preseki te familije, kao i skupove koji se dobijaju uzimanjem komplementa. Može se pokazati da $B(\mathbb{R})$ sadrži sve otvorene, zatvorene, poluotvorene intervale i unije tih skupova.*

Aksioma 1.2 (Aksioma verovatnoće)

Neka je Ω skup elementarnih događaja i F σ -polje nad Ω . Funkcija $P: F \rightarrow [0,1]$ se zove verovatnoća na prostoru (Ω, F) ako zadovoljava sledeće uslove:

1. $P(\Omega) = 1$,
2. Ako $\{A_i\}_{i \in \mathbb{N}} \subseteq F$, $A_i \cap A_j = \emptyset$, $i \neq j$, $i, j = 1, 2, \dots$, onda $P(\sum_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$.

Prostor verovatnoća je uređena trojka (Ω, F, P) , gde je Ω skup svih elementarnih događaja, F je σ -polje nad Ω , a P je verovatnoća na (Ω, F) .

Definicija 1.3 *Prostor verovatnoća je trojka (Ω, F, P) , određena slučajnim eksperimentom, gde je Ω skup svih mogućih ishoda slučajnog eksperimenta, F je σ -algebra, a P je funkcija koja svakom skupu $A \in F$ dodeljuje broj $P(A)$, koji se naziva verovatnoća da se desi događaj A .*

Definicija 1.4 Neka je (Ω, \mathcal{F}, P) prostor verovatnoća i $A, B \in \mathcal{F}$, pri čemu je $P(B) > 0$. Uslovna verovatnoća $P(A|B)$ (verovatnoća događaja A pod uslovom da se realizovao događaj B) je

$$P(A|B) = \frac{P(AB)}{P(B)}.$$

Definicija 1.5 Preslikavanje $X: \Omega \rightarrow \mathbb{R}$ je slučajna promenljiva nad prostorom verovatnoća (Ω, \mathcal{F}, P) , ako $X^{-1}(S) \in \mathcal{F}$, za svaki Borelov skup S . Ekvivalentno, kažemo da je $X \mathcal{F}$ merljivo.

Definicija 1.6 Slučajna promenljiva X je diskretna (diskretnog tipa) ako postoji prebrojiv skup brojeva R_X takav da je $P\{X \in \overline{R_X}\} = 0$, odnosno ako je skup slika od X najviše prebrojiv skup.

Definicija 1.7 Funkcija $F_X(x) = \mathbb{R} \rightarrow [0,1]$ definisana sa

$$F_X(x) = P_X((-\infty, x)) = P\{\omega \in \Omega \mid X(\omega) < x\},$$

naziva se funkcija raspodele slučajne promenljive X .

Definicija 1.8 Slučajna promenljiva X je apsolutno neprekidna ako postoji nenegativna integrabilna funkcija $\varphi_X(x)$, $x \in \mathbb{R}$, takva da

$$P(X \in S) = \int_S \varphi_X(x), \text{ za } S \in \mathcal{B},$$

gde je \mathcal{B} Borelova σ – algebra.

Specijalno, za $S = [-\infty, x)$ sledi da je

$$F_X(x) = \int_{-\infty}^x (t) dt, \text{ za svako } x \in \mathbb{R},$$

što odgovara funkciji raspodele apsolutno neprekidne slučajne promenljive.

Definicija 1.9 Funkcija

$$F_X(x) = F_{(X_1, \dots, X_n)}(x_1, \dots, x_n) = P(X_1 < x_1, \dots, X_n < x_n), \text{ za } (x_1, \dots, x_n) \in \mathbb{R}^n,$$

zove se zajednička funkcija raspodele slučajnih promenljivih (X_1, \dots, X_n) .

Definicija 1.10 Slučajna promenljiva $X = (X_1, \dots, X_n)$ je diskretna (diskretnog tipa) ako postoji prebrojiv skup tačaka u \mathbb{R}^n :

$$R_X = \{(x_1^{(k_1)}, \dots, x_n^{(k_n)}); k_1, \dots, k_n = 1, 2, \dots\}$$

takav da je $P\{X \in \overline{R_X}\} = 0$. Skup R_X je skup svih mogućih vrednosti za X .

Definicija 1.11 Slučajna promenljiva $X = (X_1, \dots, X_n)$ je n -dimenzionalna slučajna promenljiva apsolutno neprekidnog tipa ako postoji integrabilna funkcija $f_X(x_1, \dots, x_n) \geq 0$, $-\infty < x_1, \dots, x_n < \infty$, takva da je, za svaki Borelov skup $S \in B_n$,

$$P\{(X_1, \dots, X_n) \in S\} = \int \dots \int_S f_X(x_1, \dots, x_n) dx_1 \dots dx_n.$$

Funkcija $f_X(x)$ zove se gustina raspodele verovatnoća (ili samo gustina raspodele) slučajne promenljive X . Specijalno, ako izaberemo $S = \{(u_1, \dots, u_n); u_k < x_k, k = \{1, \dots, n\}\}$ dobijamo

$$F_X(x_1, \dots, x_n) = \int_{-\infty}^{x_1} \dots \int_{-\infty}^{x_n} f_X(u_1, \dots, u_n) du_n \dots du_1$$

za svako $(x_1, \dots, x_n) \in R^n$.

U nastavku definisani pojmovi su dati za jednodimenzionalne slučajne promenljive ($n = 1$).

Definicija 1.12 Kvantil reda p je ona vrednost x_p za koju važi da je

$$F_X(x_p) = P(X < x_p) = p, \quad p \in (0,1),$$

gde $F_X(x) = P(X < x)$ predstavlja funkciju raspodele za jednodimenzionalnu apsolutno neprekidnu slučajnu promenljivu.

Definicija 1.13 Neka je X apsolutno neprekidna slučajna promenljiva. Očekivanje slučajne promenljive X dato je sa,

$$E(X) = \int_{-\infty}^{\infty} x \varphi_X(x) dx,$$

i ono postoji ako gornji integral apsolutno konvergira, odnosno, ako

$$\int_{-\infty}^{\infty} |x| \varphi_X(x) dx < \infty.$$

Definicija 1.14 Momenat reda k , $k \in \mathbb{N}$, slučajne promenljive X je $E(X^k)$. Centralni momenat reda k , $k \in \mathbb{N}$, slučajne promenljive X je

$$E((X - E(X))^k).$$

Dakle, matematičko očekivanje je momenat reda 1.

Definicija 1.15 Centralni momenat reda 2 slučajne promenljive X zove se disperzija (varijansa) slučajne promenljive X i označava se sa $D(X)$ ili $\sigma^2(X)$.

Dakle,

$$D(X) = E((X - E(X))^2).$$

Za izračunavanje disperzije često se koristi i sledeći izraz:

$$D(X) = E(X^2) - E^2(X).$$

Definicija 1.16 Standardna devijacija (standardno odstupanje, prosečno odstupanje) slučajne promenljive X se definiše kao

$$\sigma(X) = \sqrt{D(X)}.$$

Numeričke karakteristike koje opisuju zavisnost između slučajnih promenljivih X i Y su kovarijansa i koeficijent korelacije.

Definicija 1.17 Kovarijansa slučajne promenljive (X, Y) je

$$\text{cov}(X, Y) = E[(X - E(X))(Y - E(Y))] = E(XY) - E(X)E(Y).$$

Definicija 1.18 Koeficijent korelacije slučajne promenljive (X, Y) je

$$\rho_{XY} = \frac{E(XY) - E(X)E(Y)}{\sqrt{D(X)D(Y)}} = \frac{\text{cov}(X, Y)}{\sqrt{D(X)D(Y)}}.$$

Slučajne promenljive X i Y su nekorelisane ako je $\rho_{XY} = 0$, pozitivno korelisane za $\rho_{XY} > 0$ i negativno korelisane za $\rho_{XY} < 0$.

Posmatrajući familiju slučajnih promenljivih koja zavisi od vremena, dolazimo do pojma stohastičkog procesa.

Definicija 1.19 Stohastički (slučajan) proces $\{X(t), t \in T\} := \{X_t, t \in T\}$ je familija slučajnih promenljivih definisanih na istom prostoru verovatnoća (Ω, F, P) .

Skup T je parametarski skup. Parametarski skup T je najčešće skup koji predstavlja vremenski interval. Ako je parametarski skup konačan, dobićemo konačno mnogo slučajnih promenljivih. Ako je prebrojiv, onda govorimo o nizu slučajnih promenljivih. Specijalno, ako je neprebrojiv, onda govorimo o „pravom” stohastičkom procesu.

2 Analiza vremenskih serija

Podaci o pojavama u ekonomiji i drugim sferama istraživanja najčešće se prikupljaju hronološki kao vremenske serije. Dok su kod klasične statističke analize ti podaci međusobno nezavisni, kod analize vremenskih serija opservacije u uzorku nisu međusobno nezavisne. Upravo ovu zavisnost koristimo u analizi vremenskih serija u cilju formiranja modela. Model zatim koristimo da na osnovu prošlih vrednosti predviđamo buduće.

Definicija 1.20 *Vremenski niz ili vremenska serija je skup hronološki uređenih vrednosti promenljive, koja predstavlja pojavu ili stohastički proces u vremenu.*

Vremenska serija može da se posmatra kao jedna realizacija stohastičkog procesa ili kao uzorak iz kolekcije svih mogućih realizacija stohastičkog procesa. Cilj analize vremenskih serija je da se na osnovu konkretne vremenske serije izvedu zaključci o karakteristikama celog procesa.

Neka $\{x_t\}_{t \in T}$ predstavlja jednu realizaciju stohastičkog procesa odnosno uzorak iz kolekcije na konačnom vremenskom intervalu dužine T [11]. Tada je:

- Uzoračko očekivanje

$$\hat{\mu}_x = \frac{1}{T} \sum_{t=1}^T x_t$$

- Uzoračka disperzija

$$\hat{\sigma}^2_x = \frac{1}{T-1} \sum_{t=1}^T (x_t - \hat{\mu}_x)^2$$

- Standardna devijacija uzorka

$$\hat{\sigma}_x = \sqrt{\hat{\sigma}^2_x}$$

- Uzorački koeficijent korelacije

$$\hat{\rho}_{x,y} = \frac{\sum_{t=1}^T (x_t - \hat{\mu}_x)(y_t - \hat{\mu}_y)}{\sum_{t=1}^T (x_t - \hat{\mu}_x)^2 (y_t - \hat{\mu}_y)^2}$$

- Uzorački koeficijent asimetrije

$$\hat{S}(x) = \frac{1}{(T-1)\hat{\sigma}_x^3} \sum_{t=1}^T (x_t - \hat{\mu}_x)^3$$

Ovaj koeficijent meri simetričnost raspodele u odnosu na očekivanu vrednost i ukazuje nam da li je većina vrednosti iz uzorka manja (negativan predznak) ili veća od očekivane vrednosti (pozitivan predznak). U zavisnosti od vrednosti koeficijenta određuje se jačina asimetrije.

1. $|\hat{S}_x| \leq 0.25$ mala asimetrija,
 2. $0.25 < |\hat{S}_x| \leq 0.50$ srednja asimetrija,
 3. $|\hat{S}_x| > 0.50$ jaka asimetrija.
- Uzorački koeficijent spljoštenosti

$$\hat{K}(x) = \frac{1}{(T-1)\hat{\sigma}_x^4} \sum_{t=1}^T (x_t - \hat{\mu}_x)^4$$

Ovaj koeficijent meri "debljinu repova" raspodele i izražava u kojoj je meri neka raspodela spljoštena u odnosu na normalnu.

1. $\hat{K}(x) < 3$ raspodela je više spljoštena u odnosu na normalnu raspodelu,
2. $\hat{K}(x) = 3$ raspodela je normalno spljoštena,
3. $\hat{K}(x) > 3$ raspodela je više izdužena u odnosu na normalnu, raspodela ima "debele repove".

$\hat{S}(x)$ i $\hat{K}(x) - 3$ asimptotski prate normalnu raspodelu sa očekivanjem 0 i disperzijama $6/T$ i $24/T$, redom.

2.1 Transformacije vremenskih serija

Postoje bar dva razloga zašto se za modeliranje cena koristi serija njenih prinosa. Prvi razlog je zbog lakše uporedivosti, jer za prinose nije bitno u kojoj su valuti izraženi. Drugi razlog je što serija prinosa često ima lepše statističke osobine kao što su stabilnija varijansa i stacionarnost. Neka je P_t cena u trenutku t [1].

Neto prinos [1] za period od $t - 1$ do t je

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}},$$

dok je neto prinos za k perioda

$$R_t[k] = \frac{P_t - P_{t-k}}{P_{t-k}}.$$

Bruto prinos [1] za *jedan* period i za k perioda predstavljeni su sledećim formulama:

$$G_t = 1 + R_t = \frac{P_t}{P_{t-1}},$$

$$G_t[k] = \frac{P_t}{P_{t-k}} = \frac{P_t}{P_{t-1}} \frac{P_{t-1}}{P_{t-2}} \dots \frac{P_{t-k+1}}{P_{t-k}} = \prod_{j=0}^{k-1} G_{t-j}.$$

Logaritamski prinos [1] predstavlja prirodni logaritam bruto prinosa i za *jedan* period iznosi

$$r_t = \ln(1 + R_t) = \ln \frac{P_t}{P_{t-1}} = L_t - L_{t-1}$$

gde je $L_t = \ln(P_t)$.

Vrednost logaritamskog prinosa za k perioda je

$$r_t[k] = \ln G_t[k] = \ln \frac{P_t}{P_{t-1}} + \dots + \ln \frac{P_{t-k+1}}{P_{t-k}} = r_t + r_{t-1} + \dots + r_{t-k+1}$$

i njegova prednost se ogleda u činjenici da je prinos za k perioda jednak sumi prinosa za jedan period, što je mnogo lakše za izračunati nego proizvod. Takođe logaritamski prinosi imaju pogodnije statističke osobine [1].

2.2 Test normalnosti

Zbog svojih lepih statističkih osobina, u analizama česta je pretpostavka da serija prinosa prati normalnu raspodelu. Normalna raspodela je u potpunosti određena sa parametrima μ i σ^2 , odnosno sa njenim očekivanjem i varijansom.

Neka je data serija prinosa $\{r_1, \dots, r_T\}$. Test koji se najčešće koristi za ispitivanje normalnosti je Jarque-Bera [11]. Ovaj test ispituje da li koeficijenti asimetrije i spljoštenosti odgovaraju normalnoj raspodeli. Dakle nulta hipoteza koja se testira je da ovi koeficijenti odgovaraju normalnoj raspodeli, odnosno da su koeficijenti asimetrije i spljoštenosti jednaki nula i tri, redom. Test statistika koja se koristi je

$$JB = \frac{S(r)^2}{6/T} + \frac{(K(r) - 3)^2}{24/T}$$

i ona asimptotski prati χ^2 raspodelu sa dva stepena slobode. Nulta hipoteza se odbacuje ukoliko je vrednost test statistike veća od kvantila χ^2 raspodele sa dva stepena slobode, ili ukoliko je registrovana p vrednost manja od zadatog nivoa značajnosti. U suprotnom nulta hipoteza se prihvata.

2.3 Pojam stacionarnosti i autokorelacije

Ispitivanje stacionarnosti vremenskih serija je jedan od prvih koraka koji se radi u analizi nekog skupa podataka. Stacionarnost je pretpostavka koja je osnova mnogih statističkih procedura korišćenih u analizi vremenskih serija. U nastavku rada kada se kaže stacionarna serija misliće se na slabo stacionarnu seriju.

Definicija 1.21 *Vremenska serija $\{r_t\}_{t \in \mathbb{N}}$ je striktno stacionarna ako $(r_{t_1}, \dots, r_{t_k})$ i $(r_{t_1+1}, \dots, r_{t_k+t})$ imaju istu raspodelu za svako t , pri čemu $t_1, \dots, t_k \in \mathbb{N}$.*

Drugim rečima, vremenska serija je striktno stacionarna ako je raspodela $(r_{t_1}, \dots, r_{t_k})$ invarijantna u odnosu na vreme. Ovo je dosta jak uslov koji je teško proveriti i zato se uvodi pojam slabe stacionarnosti.

Definicija 1.22 *Vremenska serija $\{r_t\}_{t \in \mathbb{N}}$ je slabo stacionarna ako su $E(r_t)$ i $Cov(r_t, r_{t-l})$ nezavisne od vremenskog trenutka t za svako $l \in \mathbb{Z}$.*

Dakle, vremenska serija je slabo stacionarna ako se neke od njenih najznačajnijih numeričkih karakteristika ne menjaju tokom vremena, odnosno ako važi:

- $E(r_t) = \mu$ (očekivanje je konstanta)
- $Cov(r_t, r_{t-l}) = y_l$ (kovarijansa zavisi samo od l)

$Cov(r_t, r_{t-l}) = y_l$ se naziva kovarijansa kašnjenja l i važi da je $y_0 = D(r_t)$ i $y_l = y_{-l}$.

Implicitna pretpostavka slaboj stacionarnosti je da su prva dva momenta konačna, odnosno $E(r_t) < \infty$ i $E(r_t^2) < \infty$.

Definicija 1.23 *Autokorelacija kašnjenja l (ACF) slabo stacionarne serije $\{r_t\}_{t \in \mathbb{N}}$ je*

$$\rho_l = \frac{Cov(r_t, r_{t-l})}{\sqrt{Var(r_t)Var(r_{t-l})}} = \frac{Cov(r_t, r_{t-l})}{Var(r_t)} = \frac{y_l}{y_0},$$

gde zbog slabe stacionarnosti važi da je $Var(r_t) = Var(r_{t-l})$.

ACF zavisi samo od l i iz definicije se lako zaključuje da je $\rho_0 = 1$, $\rho_l = \rho_{-l}$ i $-1 \leq \rho_l \leq 1$.

Definicija 1.24 *Uzoračka autokorelacija kašnjenja l (ACF) slabo stacionarne vremenske serije $\{r_t / t = 1, \dots, T\}$ je*

$$\hat{\rho}_l = \frac{\sum_{t=l+1}^T (r_t - \bar{r})(r_{t-l} - \bar{r})}{\sum_{t=1}^T (r_t - \bar{r})^2}, \quad 0 \leq l < T - 1$$

gde je \bar{r} uzoračka srednja vrednost vremenske serije.

Ljung-Box statistika

Ljung i Box (1978) su definisali statistički test koji ispituje odsustvo autokorelacije među članovima vremenske serije i on je prikazan sledećom formulom:

$$Q(m) = T(T + 2) \sum_{l=1}^m \frac{\hat{\rho}_l^2}{T-l}$$

Nulta hipoteza koja se testira je $H_0: \rho_1 = \dots = \rho_m = 0$ odnosno ne postoji autokorelacija među članovima vremenske serije protiv alternativne $H_a: \rho_i \neq 0$ za neko $i \in \{1, \dots, m\}$. Nulta hipoteza se odbacuje na nivou značajnosti α ako je $Q(m) > \chi_\alpha^2$, gde je χ_α^2 kvantil hi-kvadrat raspodele sa m stepeni slobode. Analogno, nulta hipoteza se odbacuje ako je registrovana p vrednost manja od nivoa značajnosti α .

U stvarnosti mnoge vremenske serije, kao što su na primer serije cena i prinosa akcija, su nestacionarne. Posmatrajući nestacionarnu seriju kroz vreme uočava se rastući ili opadajući trend, što znači da se njena statistička svojstva menjaju tokom vremena. Ukoliko je uočen rastući trend, očekivana vrednost će se tokom vremena povećavati. Korišćenje nestacionarnih vremenskih serija u regresionoj analizi može da dovede do pogrešnih zaključaka. Na primer, može da se javi "lažna regresija" među podacima. Dalje dobijeni ocenjeni koeficijenti nisu najefikasniji mogući i model može da ima dobre statističke osobine ali da ima jako lošu prediktivnu moć. Jedan od načina da se eliminišu trendovi u vremenskim serijama jeste da se one diferenciraju i svedu na stacionarnu seriju. Najpre ćemo dati definiciju linearne vremenske serije, jer ćemo stacionarnost ispitati pri izgradnji linearnih modela.

Definicija 2.6.1 *Vremenska serija a_t se naziva beli šum ako je $\{a_t\}$ niz nezavisnih i jednako raspodeljenih slučajnih promenljivih sa konačnim očekivanjem i varijansom. Ako a_t prati normalnu raspodelu sa očekivanjem nula i disperzijom σ^2 ta serija se naziva Gausovski beli šum.*

Definicija 1.25 *Vremenska serija $\{r_t\}$, $t = 1, \dots, T$ je linearna ako se može prikazati u obliku*

$$r_t = \mu + \sum_{i=0}^{\infty} \psi_i a_{t-i},$$

gde je μ aritmetička sredina niza $\{r_t\}$, $\psi_0 = 1$, a $\{a_t\}$, $t = 1, \dots, T$ je proces belog šuma sa očekivanjem nula.

Za ispitivanje stacionarnosti linearne vremenske serije koriste se razni testovi jediničnog korena. Mi smo izabrali Dicky – Fuller test (DF test) jediničnog korena [1]. Prvi model koji razmatramo je autoregresivni model prvog reda – AR(1) model, koja ima sledeći oblik:

$$r_t = \phi_0 + \phi_1 r_{t-1} + a_t,$$

gde ϕ_0 i ϕ_1 predstavljaju konstante, a $\{a_t\}$, $t=1, \dots, T$ je proces belog šuma sa očekivanjem nula.

Pod pretpostavkom da važi stacionarnost vremenske serije, dobijamo da

$$E(r_t) = \mu = \frac{\phi_0}{1-\phi_1} \text{ i } D(r_t) = \sigma^2 = \frac{\sigma_a^2}{1-\phi_1^2},$$

takođe, iz stacionarnosti sledi da $|\phi_1| < 1$.

Definicija 1.26 Niz $\{p_t\}$ se zove slučajan hod ako važi da $p_t = p_{t-1} + a_t$, gde je $p_0 \in \mathbb{R}$, $\{a_t\}$, $t = 1, \dots, T$ je beli šum sa očekivanjem nula.

Posmatranjem slučajanog hoda kao AR(1) model, dobijamo da $\phi_1 = 1$, a znamo da nam onda propada uslov slabe stacionarnosti. Ovo je osnovna motivacija testova jediničnih korena.

Da bi testirali da li neka linearna vremenska serija prati slučajan hod, posmatrajmo sledeće modele

$$p_t = \phi_1 p_{t-1} + a_t$$

$$p_t = p_0 + \phi_1 p_{t-1} + a_t$$

$$p_t = p_0 + \delta t + \phi_1 p_{t-1} + a_t$$

gde a_t predstavljaju greške modela, $p_0, \delta \in \mathbb{R}$ su konstante, a t je vremenska komponenta.

Prvo se ocenjuje koeficijent $\hat{\phi}_1$ za svaku od posmatranih jednačina, a zatim se testira hipoteza $H_0 : \hat{\phi}_1 = 1$ protiv $H_1 : \hat{\phi}_1 \neq 1$ DF testom jediničnog korena, koja ima sledeći oblik testa statistike

$$DF = \frac{\hat{\phi}_1 - 1}{SE(\hat{\phi}_1)}$$

gde je $SE(\hat{\phi}_1)$ predstavlja standardno odstupanje ocenjenog parametara $\hat{\phi}_1$. Ako je apsolutna vrednost DF testa statistike veća od zadate kritične vrednosti, odbacujemo nultu hipotezu.

2.4 Autoregresivni modeli (AR)

Autoregresivni model reda p , u oznaci AR(p) model, definiše se na sledeći način:

$$r_t = \phi_0 + \phi_1 r_{t-1} + \phi_2 r_{t-2} + \dots + \phi_p r_{t-p} + a_t \quad (2.1)$$

gde je $p \in \mathbb{N}_0$, $\phi_0, \phi_1, \phi_2, \dots, \phi_p$ su autoregresivni parametri, a_t je beli šum sa očekivanjem nula i dobro definisanom disperzijom. Naziv modela odgovara prethodnom izrazu (2.1), u pitanju je regresioni model u kojem je zavisna promenljiva predstavljena članom vremenske serije u trenutku t , dok skup objašnjavajućih promenljivih čine članovi iste vremenske serije, ali u

prethodnim trenutcima u odnosu na zavisnu promenljivu, $t-1, \dots, t-p$. Drugim rečima, data promenljiva opisuje se u funkciji od sopstvenih prethodnih vrednosti.

2.4.1 AR(1) model

Autoregresivni model prvog reda, AR(1), predstavlja se na sledeći način:

$$r_t = \phi_0 + \phi_1 r_{t-1} + a_t. \quad (2.2)$$

gde je, kao što je već rečeno a_t beli šum sa očekivanjem nula i disperzijom σ_a^2 , a ϕ_1 autoregresivni koeficijent. Navešćemo dovoljan i potreban uslov za slabu stacionarnost AR(1) modela u vidu sledeće dve teoreme.

Teorema 2.1 Neka $\{r_t\}$ prati AR(1) model, tj.

$$r_t = \phi_0 + \phi_1 r_{t-1} + a_t,$$

gde je a_t beli šum sa $E(a_t) = 0$, $D(a_t) = \sigma_a^2$ i a_t su nezavisne i jednako raspodeljene. Ako je $\{r_t\}$ slabo stacionarna serija, onda je $|\phi_1| < 1$.

Teorema 2.2 Neka $\{r_t\}$ prati AR(1) model, tj.

$$r_t = \phi_0 + \phi_1 r_{t-1} + a_t,$$

gde je a_t beli šum sa $E(a_t) = 0$, $D(a_t) = \sigma_a^2$ i a_t su nezavisne i jednako raspodeljene i pretpostavimo da $|E(r_t)| \leq S < \infty$, za svako t . Tada je $\{r_t\}$ slabo stacionarna ako je $|\phi_1| < 1$.

Očekivanje AR(1) modela je dato sa

$$\mu = E(r_t) = \frac{\phi_0}{1-\phi_1}.$$

Disperzija modela AR(1) je predstavljena formulom

$$\sigma^2 = D(r_t) = \frac{\sigma_a^2}{1-\phi_1^2}.$$

Da bi očekivanje modela postojalo mora da važi $\phi_1^2 \neq 1$. Očekivanje modela je nula ako i samo ako je $\phi_0 = 0$. Disperzija treba da bude ograničena i nenegativna, tako da važi $\phi_1^2 < 1$. Slaba stacionarnost AR(1) modela implicira da $\phi_1 \in (-1, 1)$.

Autokovarijansna i autokorelaciona funkcija kašnjenja l su date sa

$$\gamma_l = \begin{cases} \phi_1 \gamma_{l-1} + \sigma_a^2, & l = 0, \\ \phi_1 \gamma_{l-1}, & l > 0, \end{cases}$$

$$\rho_l = \phi_1^l,$$

respektivno. Autokorelaciona funkcija ρ_l kašnjenja l , slabo stacionarnog modela AR(1) eksponencijalno opada sa stopom ϕ_1 .

2.4.2 AR(2) model

Autoregresivni model drugog reda, AR(2), predstavlja se na sledeći način:

$$r_t = \phi_0 + \phi_1 r_{t-1} + \phi_2 r_{t-2} + a_t. \quad (2.3)$$

gde je a_t beli šum sa očekivanjem nula i disperzijom σ_a^2 , a ϕ_1 i ϕ_2 autoregresivni koeficijenti. Očekivanje modela AR(2) dato je sa

$$\mu = E(r_t) = \frac{\phi_0}{1 - \phi_1 - \phi_2}.$$

Da bi očekivanje postojalo mora da važi $\phi_1 + \phi_2 \neq 1$.

Autokovarijansna funkcija kašnjenja l :

$$\gamma_l = \begin{cases} \phi_1 \gamma_{l-1} + \phi_2 \gamma_{l-2} + \sigma_a^2, & l = 0, \\ \phi_1 \gamma_{l-1} + \phi_2 \gamma_{l-2}, & l > 0, \end{cases}$$

Disperzija modela AR(2) data je sa

$$\sigma^2 = D(r_t) = \gamma_0 = \frac{\sigma_a^2 + 2\phi_0\phi_1\gamma_1}{1 - \phi_1^2 - \phi_2^2}.$$

Kako disperzija treba da bude nenegativna i ograničena treba da važi $\phi_1^2 + \phi_2^2 < 1$.

Autokorelaciona funkcija kašnjenja l :

$$\rho_l = \frac{\gamma_l}{\gamma_0} = \frac{\phi_1 \gamma_{l-1} + \phi_2 \gamma_{l-2}}{\gamma_0} = \phi_1 \frac{\gamma_{l-1}}{\gamma_0} + \phi_2 \frac{\gamma_{l-2}}{\gamma_0} = \phi_1 \rho_{l-1} + \phi_2 \rho_{l-2}, \quad (2.4)$$

$$\rho_0 = \frac{\gamma_0}{\gamma_0} = 1,$$

$$\rho_1 = \phi_1 \rho_0 + \phi_2 \rho_{-1} = \phi_1 + \phi_2 \rho_{-1} \Rightarrow \rho_{-1} = \frac{\phi_1}{1 - \phi_2},$$

Jednačina (2.4) govori da autokorelaciona funkcija AR(2) modela zadovoljava diferencnu jednačinu drugog reda

$$(1 - \phi_1 B - \phi_2 B^2)\rho_l = 0,$$

gde B nazivamo operator unazad (engl. back-shift) i važi $B\rho_t = \rho_{t-1}$. Prikazana diferencna jednačina određuje osobine autokorelacione funkcije i ponašanje prediktivnih vrednosti. Odgovarajuća polinomna jednačina za datu diferencnu jednačinu je

$$1 - \phi_1 x - \phi_2 x^2 = 0.$$

Rešenje ove jednačine je dato sa

$$x_{1,2} = \frac{\phi_1 \pm \sqrt{\phi_1^2 + 4\phi_2}}{-2\phi_2}.$$

Inverzna vrednost rešenja $x_{1,2}$ predstavlja karakteristične korene modela AR(2) u oznaci $\omega_{1,2}$. Uslov slabe stacionarnosti postaje $|\omega_{1,2}| < 1$.

2.4.3 AR(p) model

AR(p) model je uopštenje modela AR(1) i AR(2). Sve osobine se izvode na sličan način, jedino što dodatno komplikuje račun jeste veći broj koeficijenata.

Očekivanje je dato sa

$$\mu = E(r_t) = \frac{\phi_0}{1 - \phi_1 - \phi_2 - \dots - \phi_p}.$$

Uvođenjem operatora unazad B za AR(p) model dobijamo diferencnu jednačinu reda p

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p)\rho_t = 0.$$

Analogno, kao u modelu AR(2) imamo polinomnu jednačinu

$$1 - \phi_1 x - \phi_2 x^2 - \dots - \phi_p x^p = 0.$$

Karakteristični koreni modela AR(p) su inverzne vrednosti rešenja ove polinomne jednačine i uslov slabe stacionarnosti je $|\omega_i| < 1$, gde je $i \in \{1, 2, \dots, p\}$.

2.4.4 Određivanje reda modela

Da bismo razlikovali AR procese različitog reda uvodimo parcijalnu autokorelacionu funkciju (eng. Partial Autocorrelation Function, PACF). U statistici je čest slučaj da je korelacija između dve promenljive ustvari rezultat njihove korelisanosti sa trećom promenljivom. Zato koristimo koeficijent parcijalne korelacije kao meru „neto” zavisnosti između dve promenljive nakon eliminisanja uticaja treće promenljive. U analizi vremenskih serija deo korelacije između X_t i X_{t-k}

može se pripisati njihovoj korelaciji sa vrednostima serije na kašnjenju između vremenskog trenutka t i $t-k$. Korelaciju između r_t i r_{t-k} , nakon što je njihova uzajamna zavisnost od tih međufaznih vrednosti odstranjena, iskazujemo na osnovu parcijalne autokorelacione funkcije. Parcijalna autokorelacija na docnji k je parcijalni regresioni koeficijent u autoregresiji k -tog reda:

$$\begin{aligned} r_t &= \phi_{01} + \phi_{11}r_{t-1} + a_{1t} \\ r_t &= \phi_{02} + \phi_{12}r_{t-1} + \phi_{22}r_{t-2} + a_{2t} \\ &\vdots \\ r_t &= \phi_{0p} + \phi_{1p}r_{t-1} + \phi_{2p}r_{t-2} + \dots + \phi_{pp}r_{t-p} + a_{pt} \end{aligned}$$

$PACF(p) = \hat{\phi}_{pp}$ predstavlja dodatni doprinos uključivanja r_{t-p} na $AR(p-1)$ model. Ako je $AR(p)$ dobar model, tj. ako je p dobar red modela, onda $PACF(p)$ treba da bude statistički različito od nule, a $PACF(j)$ za $j > p$ treba da bude statistički jednako nuli.

2.4.5 Ocena parametara

Parametre AR modela ocenjujemo pomoću metoda najmanjih kvadrata. Za $AR(p)$ model:

$$r_t = \phi_0 + \phi_1 r_{t-1} + \phi_2 r_{t-2} + \dots + \phi_p r_{t-p} + a_t$$

na osnovu vremenske serije $\{r_t\}_{t=1, \dots, T}$ koja predstavlja poznatu istoriju podataka, ocenjujemo nepoznate parametre $\phi_0, \phi_1, \phi_2, \dots, \phi_p$.

$$\hat{r}_t = E(r_t | r_1 \dots r_{t-1}) = \hat{\phi}_0 + \hat{\phi}_1 r_{t-1} + \hat{\phi}_2 r_{t-2} + \dots + \hat{\phi}_p r_{t-p} := h_t(\vec{\phi})$$

pri čemu je $r_1 \dots r_{t-1}$ zapravo cela istorija do trenutka $t-1$, $\vec{\phi} = [\phi_p, \dots, \phi_0]^T$, $h_t(\vec{\phi})$ je model funkcija, a \hat{r}_t je predikcija za vrednost u trenutku t , tj. aproksimacija buduće vrednosti koju daje model funkcija (2.6).

Greška u trenutku t je: $h_t(\vec{\phi}) - r_t$ pa problem najmanjih kvadrata možemo zapisati na sledeći način:

$$\min_{\phi \in \mathbb{R}^{p+1}} f(\phi) = \frac{1}{2} \sum_{t=p+1}^T (h_t(\vec{\phi}) - r_t)^2. \quad (2.5)$$

Ekvivalentan zapis:

$$\min_{\phi \in \mathbb{R}^{p+1}} f(\phi) = \frac{1}{2} \|F(\phi) - y\|^2,$$

gde su $F(\phi) = [h_{p+1}(\phi), \dots, h_T(\phi)]^T$, $y = [r_{p+1}, \dots, r_T]^T$.

Standardna grška za ocenjene parametre $\hat{\phi}_i$:

$$SE(\hat{\phi}_i) = \sqrt{[COV(\hat{\phi})]_{n-i, n-i}}, \text{ za } i = 0, \dots, p \text{ i } n = p+1.$$

Matrica kovarijanse je data sa:

$$COV(\hat{\phi}) = (tm^T tm)^{-1} \sigma_a^2,$$

gde su σ_a^2 i matrica tm dati sa:

$$\sigma_a^2 = \frac{\sum_{t=p+1}^T (r_t - \hat{r}_t)^2}{T-2p-1} = \frac{2f(\hat{\phi})}{T-2p-1},$$

$$tm = \begin{bmatrix} r_1 & \cdots & r_p & 1 \\ \vdots & \ddots & \vdots & \\ r_{T-p} & \cdots & r_{T-1} & 1 \end{bmatrix}.$$

Testiranje značajnosti ocenjenih koeficijenata

Nulta hipoteza je $H_0(\hat{\phi}_i = 0)$, tj. koeficijent je statistički jednak nuli i nije značajan za model.

Test:

$$\left| \frac{\hat{\phi}_i}{SE(\hat{\phi}_i)} \right| > \left| Z_{\frac{\alpha}{2}} \right|,$$

ako odbacimo H_0 , trebali bismo da uključimo r_{t-i} u model i $\hat{\phi}_i$ je statistički različito od nule.

Ako je $\hat{\phi}_0 \neq 0$ onda je $E(r_t) \neq 0$, jer

$$E(r_t) = \frac{\phi_0}{1 - \phi_1 - \phi_2 - \dots - \phi_p} := \hat{\mu}.$$

2.4.6 Adekvatnost AR modela

Ako je model adekvatan, trebalo bi da dovede do toga, da se reziduali ponašaju kao beli šum, u smislu da je korelacija statistički jednaka nuli. Ako su neki koeficijenti statistički jednaki nuli, izbacujemo ih i uprošćavamo model.

Ako je g broj koeficijenata statistički različitih od nule, tada $Q(m)$ prati χ_{m-g}^2 raspodelu, za m se često uzima vrednost $m = \lceil \ln(T) \rceil$. Ako vazi $H_0(\rho_1 = \dots = \rho_m = 0)$, model je adekvatan.

Stoga ćemo u daljem radu proveravati da li postoji linearna zavisnost u $\{\hat{a}_t\}$, preciznije, da li postoji autokorelacija reziduala.

2.4.7 Predikcije

Formiranje predikcija je jedan od glavnih zadataka analize vremenskih serija. Pretpostavimo da se nalazimo u vremenskom trenutku t , zanimaju nas buduće vrednosti serije. Posmatramo AR(p) model

$$r_t = \phi_0 + \phi_1 r_{t-1} + \phi_2 r_{t-2} + \dots + \phi_p r_{t-p} + a_t,$$

zanimaju nas buduća (prava) vrednost vremenske serije r_{t+1} dobijena pomoću AR(p) modela

$$r_{t+1} = \phi_0 + \phi_1 r_t + \phi_2 r_{t-1} + \dots + \phi_p r_{t+1-p} + a_{t+1}.$$

Predikcija za 1 korak unapred, odnosno predviđena vrednost za trenutak $t+1$ dobija se na sledeći način:

$$\bar{r}_{t+1} \approx \hat{r}_t(1) = E(r_{t+1} | r_1 \dots r_t) = \hat{\phi}_0 + \hat{\phi}_1 r_t + \hat{\phi}_2 r_{t-1} + \dots + \hat{\phi}_p r_{t+1-p}. \quad (2.6)$$

Greška predikcije je:

$$e_t(1) = r_{t+1} - \hat{r}_t(1) = a_{t+1},$$

disperzija greške je:

$$D(e_t(1)) = D(a_{t+1}) = \sigma_a^2.$$

Ako a_t prati Gausovski beli šum, onda je interval poverenja za r_{t+1}

$$\hat{r}_t(1) \pm |Z_{\frac{\alpha}{2}}| \sigma_a$$

jer je

$$E(r_{t+1} | r_1 \dots r_t) = \hat{r}_t(1),$$

$$D(r_{t+1} | r_1 \dots r_t) = D(\hat{r}_t(1) + a_{t+1} | r_1 \dots r_t) = D(a_{t+1}) = \sigma_a^2.$$

Predikcija za 2 korak unapred, odnosno predviđena vrednost za trenutak $t+2$ dobija se na sledeći način:

$$r_{t+2} = \phi_0 + \phi_1 r_{t+1} + \phi_2 r_t + \dots + \phi_p r_{t+2-p} + a_{t+2}.$$

Ocena predikcije za trenutak $t+2$:

$$\begin{aligned}\bar{r}_{t+2} &\approx \hat{r}_t(2) = E(r_{t+2}|r_1 \dots r_t) = \hat{\phi}_0 + \hat{\phi}_1 E(r_{t+1}|r_1 \dots r_t) + \hat{\phi}_2 r_t + \dots + \hat{\phi}_p r_{t+2-p} \\ &= \hat{\phi}_0 + \hat{\phi}_1 \hat{r}_t(1) + \hat{\phi}_2 r_t + \dots + \hat{\phi}_p r_{t+2-p}.\end{aligned}$$

Kako nam vrednost predikcija za trenutak $t+1$ utiče na vrednost predikcije za trenutak $t+2$, sledi da greška predikcije u trenutku $t+1$ utiče na grešu predikcije za trenutak $t+2$.

Greška predikcije je:

$$\begin{aligned}e_t(2) &= r_{t+2} - \hat{r}_t(2) = \phi_0 + \phi_1 r_{t+1} + \phi_2 r_t + \dots + \phi_p r_{t+2-p} + a_{t+2} \\ &\quad - (\hat{\phi}_0 + \hat{\phi}_1 \hat{r}_t(1) + \hat{\phi}_2 r_t + \dots + \hat{\phi}_p r_{t+2-p}) \\ &= \phi_1 (r_{t+1} - \hat{r}_t(1)) + a_{t+2} \\ &= \phi_1 a_{t+1} + a_{t+2}\end{aligned}$$

disperzija greške je:

$$D(e_t(2)) = \phi_1^2 \sigma_a^2 + \sigma_a^2 = (\phi_1^2 + 1) \sigma_a^2 \geq D(e_t(1)).$$

Disperzija se uvećava sa porastom broja predikcija.

Ako a_t prati Gausovski beli šum, onda je interval poverenja za r_{t+2}

$$\hat{r}_t(2) \pm |Z_{\frac{\alpha}{2}}| \sqrt{1 + \phi_1^2} \sigma_a.$$

Predikcija za k koraka unapred, odnosno predviđena vrednost za trenutak $t+k$ dobija se na sledeći način:

Prava vrednost serije u trenutku $t+k$:

$$r_{t+k} = \phi_0 + \phi_1 r_{t+k-1} + \phi_2 r_{t+k-2} + \dots + \phi_p r_{t+k-p} + a_{t+k}.$$

Ocena predikcije za trenutak $t+k$:

$$\bar{r}_{t+k} \approx \hat{r}_t(k) = E(r_{t+k}|r_1 \dots r_t) = \hat{\phi}_0 + \sum_{i=1}^p \hat{\phi}_i \hat{r}_t(k-i),$$

$$e_t(k) = r_{t+k} - \hat{r}_t(k).$$

Za slabo stacionarne vremenske serije koje prate AR(p) model važi:

$$\lim_{k \rightarrow \infty} \hat{r}_t(k) = E(r_t).$$

Predikcije se vraćaju srednjoj vrednosti.

$$\lim_{k \rightarrow \infty} D(e_t(k)) \rightarrow D(r_t).$$

2.5 Distribuirani najmanji kvadrati

Problem najmanjih kvadrata koji smo definisali u (2.5) možemo posmatrati kao problem distribuirane optimizacije tako što ćemo sumu najmanjih kvadrata podeliti na N podskupova, koji sadrže samo te parcijalne sisteme jednačina na osnovu kojih je formiran sistem najmanjih kvadrata, pri čemu je N konačan broj. Dakle rešavamo sledeći problem:

$$\min_{\phi \in \mathbb{R}^{p+1}} f(\phi) = \frac{1}{2} \sum_{i=1}^N f_i(\phi), \quad (2.7)$$

gde je svaki

$$f_i(\phi) = \sum_{t \in \mu_i} (h_t(\vec{\phi}) - r_t)^2, \quad \mu_i \subseteq \{1, \dots, N\}, \quad (2.8)$$

pri čemu se skupovi μ_i koje smo definisali u (2.8) uglavnom kreiraju na sledeći način:

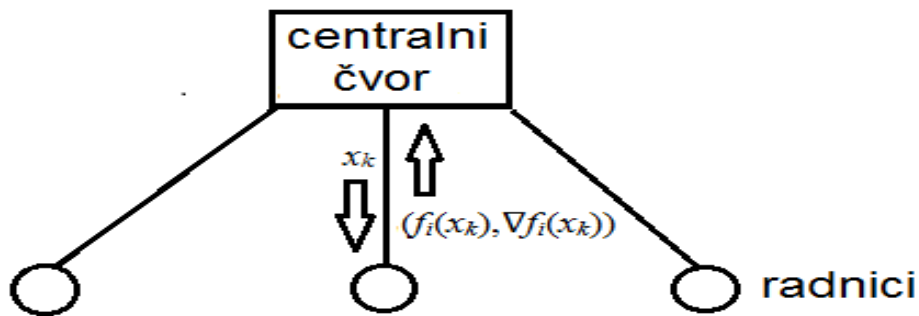
$$\mu_i = \left\{ (i-1) * \left\lfloor \frac{T}{N} \right\rfloor + 1, \dots, i * \left\lfloor \frac{T}{N} \right\rfloor \right\},$$

pri čemu $\frac{T}{N}$ ne mora da bude ceo broj, u tom slučaju se ostatak sume raspoređuje ili se stavlja u jedan skup.

2.5.1 Model i algoritam

Koristimo centralni čvor - radnik model izračunavanja (Slika 1), koji rešava problem (2.7) iterativnim algoritmom, kao što sledi. Algoritam se sastoji od spoljašnjih iteracija k i unutrašnjih iteracija s . U spoljašnjim (glavnim) iteracijama računa se niz aproksimacija rešenja $\{x_k\} \in \mathbb{R}^n$. U svakoj iteraciji k , centralni čvor šalje svim radnicima vrednosti x_k i kontrolni parametar $p_k \in [0, 1]$, a svaki aktivan radnik računa par $(f_i(x_k), \nabla f_i(x_k))$ i šalje ga nazad centralnom čvoru. U svakoj iteraciji k , svaki radnik može da bude u jednom od dva moguća stanja: aktivno i neaktivno. Kao što ćemo videti kasnije, stanja svakog radnika se ne menjaju kroz unutrašnje iteracije s .

Pretpostavljena apstraktna arhitektura centralni čvor - radnik je pogodna za situacije gde su dimenzije promenljive n i broj čvorova N umerene do srednje veličine, dok svaka f_i može da odgovara velikim skupovima podataka za trening modela, pa je ekonomično da svaki radnik drži lokalno trening podatke, računa par $(f_i(x_k), \nabla f_i(x_k))$ i šalje centralnom čvoru izračunate vrednosti [9]. Odnosno u našem slučaju, kako smo sumu najmanjih kvadrata podelili na onoliko podskupova koliko imamo radnika, svaki radnik drži lokalno jedan od N podskupova sume najmanjih kvadrata.



Slika 1: Ilustracija računarskog modela centralni čvor - radnik. U spoljnjim iteracijama k , centralni čvor šalje radnicima x_k , dok aktivni radnici šalju nazad par $(f_i(x_k), \nabla f_i(x_k))$. U unutrašnjim iteracijama s koje se dešavaju na početku k -te spoljašnje iteracije, centralni čvor šalje radnicima probnu tačku $x_k + \beta^s d_k$ dok aktivni radnici šalju centralnom čvoru nazad $f_i(x_k + \beta^s d_k)$ [9].

Nakon što smo svakom radniku dodeliti jedan od N podskupova, aktivni radnici će pristupiti računanju lokalno, svaki nad svojim podskupom podataka, a zatim će samo izračunate vrednosti prosleđivati centralnom čvoru koji će vršiti sumarnu kalkulaciju pomoću dostavljenih rezultata od strane čvorova radnika.

Kada su skupovi podataka za trening modela veliki, može se primenjivati gradijentni metod u kojem je, radi smanjenja troškova, pun gradijent (gradijent funkcije f) zamenjen, po pitanju troškova izračunavanja, jeftinijom aproksimacijom u svakoj iteraciji k . Aproksimacija $\nabla f(x_k)$ ($x_k \in \mathbb{R}^n$ je trenutna iteracija), je oblika $\frac{1}{|N_k|} \sum_{i \in N_k} \nabla f_i(x_k)$, gde je N_k podskup skupa $N = \{1, \dots, N\}$.

Koristimo metod nemonotonog linijskog pretraživanja, sa adaptivnim izborom skupova N_k i centralni čvor - radnik model računarskih mašina u klasteru, sa jednim centralnim čvorom i N čvorova radnika. Ovaj model je adekvatan za distribuiranu optimizaciju velikih razmera u klaster ili cloud okruženju, gde je skup podataka kod svakog radnika i (koji parametrizuje funkciju f_i) toliko velik da je nemoguće procesirati ili čuvati podatke svih radnika na centralnom čvoru, ili radnici ne mogu slati njihove podatke centralnom čvoru zbog ograničenja koja su posledica zaštite privatnosti podataka.

Centralni čvor organizuje izračunavanje i ažurira ocenu rešenja $x_k \in \mathbb{R}^n$. Svaki radnik, u svakoj iteraciji k , može biti u jednom od dva moguća stanja: aktivnom ili neaktivnom. Svaki radnik i sadrži jednu funkciju $f_i(x)$ definisanu u (2.8) i može da izračuna njenu vrednost i vrednost njenog gradijenta. Aktivni radnici i računaju par $(f_i(x_k), \nabla f_i(x_k))$ i šalju ga centralnom čvoru, dok neaktivni radnici ostaju neaktivni po pitanju komunikacija, ali i izračunavanja vrednosti lokalnih funkcija i njihovih izvoda. Broj aktivnih radnika kontrolisan je od strane centralnog čvora i povećava se ili se smanjuje u zavisnosti od procenjenog smanjenja vrednosti

funkcije cilja. Dakle, centralni čvor šalje informaciju radnicima o tome kolika (prosečna) veličina uzorka je potrebna za efikasan progres u sledećoj iteraciji, pomoću parametra p_k koji predstavlja verovatnoću uključivanja svakog čvora u iteraciji k .

Pre nego što navedemo glavni algoritam, definisaćemo dva pomoćna algoritma, koje koristi glavni algoritam. Ažuriranje donje granice p_k^{min} u centralnom čvoru je definisano sa Algoritmom 1. Pre nego što ga navedemo predstavimo parametre θ_k i γ_k koji određuju dovoljan pad u aproksimaciji funkcije cilja i povećanje donje granice. Oba parametra su proizvoljna, pretpostavljamo samo da su pozitivni i uniformno ograničeni od dole pozitivnom konstantom. Primetimo da je niz $\{p_k^{min}\}_{k \in N}$ neopadajući i ograničen od gore sa 1.

2.5.2 Algoritam 1.

Ažuriranje donje granice p_k^{min} u centralnom čvoru.

S0 Ulazni parametri: $p_k, p_{k+1}, f_{N_{k+1}}(x_{k+1}), f_k^{min}, \theta_k, \gamma_k$.

Neka je j takvo da $\pi_j = p_{k+1}$.

S1 Ažuriranje donje granice.

Ako je $p_k < p_{k+1}$ i ako je $[F_k]_j - f_{N_{k+1}}(x_{k+1}) < \theta_k$ postavlja se

$$p_{k+1}^{min} = \min\{1, p_k^{min} + \gamma_k\}.$$

U suprotnom, $p_{k+1}^{min} = p_k^{min}$.

S2 Ažuriranje vektora F_k

Ako je $p_k < p_{k+1}$, postavlja se

$$[F_{k+1}]_j = \min\{f_{N_{k+1}}(x_{k+1}), [F_k]_j\}.$$

U suprotnom $F_{k+1} = F_k$.

Glavna ideja je da se poveća donja granica i da se time poveća očekivana preciznost aproksimacije funkcije cilja ako nemamo dovoljno velik pad vrednosti relevantne aproksimativne funkcije. Preciznije, pratimo pad očekivanog nivoa preciznosti određenog sa p_{k+1} koji će se koristiti u sledećoj iteraciji. Primetimo da ako je p_{k+1} korišćeno po prvi put, tada $[F_{k+1}]_j = +\infty$ i donja granica ostaje nepromenjena. Podsetimo se i da broj aktivnih čvorova $|N_{k+1}|$ ne mora da bude jednak N_{k+1} [9].

Sada ćemo definisati algoritam za ažuriranje kontrolnog parametra p_k i glavni algoritam. Ovi algoritmi su predstavljeni sa generičkim pravcima pretraživanja, generičkom funkcijom ε_k i generičkim parametrima θ_k, γ_k . Algoritmi daju dosta slobode u njihovom izboru.

2.5.3 Algoritam 2.

Ažuriranje kontrolnog parametra p_k u centralnom čvoru.

S0 Ulazni parametri: $dm_k, p_k^{min}, p_k, \varepsilon_k(p_k), v_1 \in (0,1)$.

S1 Ako je $p_k < p_k^{min}$ onda je $p_{k+1} = \min\{\pi_j \in \Pi: \pi_j \geq p_k^{min}\}$, u suprotnom prelazi se na korak S2.

S2 1) Ako je $dm_k = \varepsilon_k(p_k)$ onda $p_{k+1} = p_k$.

2) Ako je $dm_k > \varepsilon_k(p_k)$ onda $p_{k+1} = \max\{\pi_j \in \Pi: p_k^{min} \leq \pi_j \leq p_k, dm_k \leq \varepsilon_k(\pi_j)\}$. Ako takvo p_{k+1} ne postoji, onda je $p_{k+1} = \min\{\pi_j \in \Pi: \pi_j \geq p_k^{min}\}$.

3) Ako je $dm_k < \varepsilon_k(p_k)$ i

i) $dm_k \geq v_1 \varepsilon_k(p_k)$ onda $p_{k+1} = \min\{\pi_j \in \Pi: \pi_j \geq p_k, dm_k \leq \varepsilon_k(\pi_j)\}$. Ako takvo p_{k+1} ne postoji, onda je $p_{k+1} = 1$.

ii) $dm_k < v_1 \varepsilon_k(p_k)$, onda je $p_{k+1} = 1$.

Glavna ideja Algoritma 2, je da drži dve mere preciznosti: dm_k i ε_k bliskim međusobno prilagođavajući očekivani nivo preciznosti. Ako je $dm_k > \varepsilon_k(p_k)$, preciznost je previsoka i zato je smanjujemo smanjivanjem kontrolnog parametra p_k , ako donja granica p_k^{min} to dozvoljava. Grubo rečeno, rešenje je još daleko i želimo da mu se približimo sa što manjim troškovima, tj. sa što manjim brojem aktivnih radnika. Sa druge strane, ako je mera napretka u funkciji cilja dm_k relativno mala, povećavamo očekivanu preciznost povećavanjem kontrolnog parametra. Navedeni algoritmi imaju za cilj da svi radnici budu aktivni, ali tek kada se približimo rešenju [9].

U nastavku navodimo glavni algoritam. Primetimo, da Algoritam 3 funkcioniše na način da svaki radnik ažurira dužinu koraka u koordinaciji sa ostalim radnicima, a centralni čvor je zadužen za računanje pravca pretraživanja i ažuriranje dužine koraka u svim unutrašnjim iteracijama linijskog pretraživanja.

2.5.4 Algoritam 3.

Glavni algoritam.

S0 Ulazni parametri kod centralnog čvora: $p_0^{min} \in (0,1)$, $x_0 \in R^n$, $\beta, v_1, \eta \in (0,1)$.

S1 Centralni čvor postavlja p_0 na p_0^{min} i šalje p_0 i x_0 svim radnicima.

S2 Svaki radnik se aktivira sa verovatnoćom p_0 , nezavisno od drugih radnika. Svaki aktivan radnik i šalje par $(f_i(x_0), \nabla f_i(x_0))$ centralnom čvoru. Centralni čvor postavlja $k = 0$.

S3 Centralni čvor računa pravac pretraživanja d_k takav da $d_k^T \nabla f_{N_k}(x_k) < 0$.

S4 Nemonotono linijsko pretraživanje kroz unutrašnje iteracije: nalazi najmanje nenegativno s_k takvo da $\alpha_k = \beta^{s_k}$ zadovoljava

$$f_{N_k}(x_k + \alpha_k d_k) \leq f_{N_k}(x_k) + \eta \alpha_k d_k^T \nabla f_{N_k}(x_k) + e_k.$$

Za svako $s = 0, 1, \dots, s_k$ centralni čvor šalje $x_k + \beta^{s_k} d_k$ svim aktivnim radnicima, i svaki aktivan radnik šalje $f_i(x_k + \beta^{s_k} d_k)$ centralnom čvoru.

S5 Centralni čvor postavlja $x_{k+1} = x_k + \alpha_k d_k$ i $dm_k = -\alpha_k d_k^T \nabla f_{N_k}(x_k)$.

S6 Centralni čvor određuje p_{k+1} pomoću Algoritma 2 i šalje x_{k+1} i p_{k+1} svim radnicima.

S7 Ažuriranje stanja radnika: ako je $p_{k+1} = p_k$, svaki radnik ostaje u svom prethodnom stanju, tj. $N_{k+1} = N_k$. U suprotnom, svaki radnik postaje aktivan sa verovatnoćom p_{k+1} , nezavisno od drugih radnika i nezavisno od svog prethodnog stanja i prethodnih stanja ostalih radnika. Skup N_{k+1} se sastoji od trenutno aktivnih čvorova.

S8 Svaki radnik i šalje par $(f_i(x_{k+1}), \nabla f_i(x_{k+1}))$ centralnom čvoru.

S9 Centralni čvor određuje p_k^{min} pomoću Algoritma 1.

S10 Centralni čvor postavlja $k = k+1$. Ako $N_k = N$ i ako je $\nabla f_{N_k}(x_k) = 0$, algoritam se završava, u suprotnom, ide se na korak S3.

Primitimo iz S10 da smo pretpostavili da je centralni čvor svestan broja radnika N_k .

Pomoću parametra p_k centralni čvor kontroliše očekivani broj aktivnih radnika u iteraciji k . Nakon primanja parametra p_k , svaki radnik odlučuje, nezavisno od drugih radnika, da li će biti

aktivan ili neaktivan u iteraciji k . On postaje aktivan sa verovatnoćom p_k ili neaktivan sa verovatnoćom $1 - p_k$. Preciznije, promena stanja svakog od radnika je predstavljena na sledeći način, u zavisnosti od trenutne i prethodne vrednosti kontrolnog parametra, p_k i p_{k-1} . (Primetimo da svaki radnik i čuva vrednosti p_k i p_{k-1}). Ako je $p_k = p_{k-1}$, tada svaki radnik zadržava isto stanje kao u prethodnoj iteraciji $k-1$. U suprotnom, ako je $p_k \neq p_{k-1}$, svaki radnik postavlja svoje stanje u k -toj iteraciji u aktivno sa verovatnoćom p_k (ili u neaktivno sa verovatnoćom $1 - p_k$), nezavisno od svih drugih radnika i nezavisno od njegovog i tuđih prethodnih stanja. Odnosno, sa svakom promenom parametra dolazi do rekonfiguracije sistema [9]. Kako je N_k zapravo skup indeksa aktivnih čvorova u iteraciji k , a $\tilde{N}_k := |N_k|$ je kardinalnost tog skupa, sa \tilde{N}_k ćemo označiti broj aktivnih radnika u iteraciji k . Primetimo da ako je $p_k \neq p_{k-1}$, \tilde{N}_k je slučajna promenljiva koja zavisi od parametra p_k i ima Binomnu raspodelu $\tilde{N}_k: B(N, p_k)$. Označimo sa $N_k := p_k N$ prosečan broj aktivnih radnika, koji zavisi od trenutne vrednosti parametra p_k .

Centralni čvor ažurira ocenu rešenja x_k na osnovu primljenih informacija od aktivnih radnika ($f_i(x_k), \nabla f_i(x_k)$), $i \in N_k$, na osnovu funkcije

$$f_{N_k}(x) := \frac{1}{|N_k|} \sum_{i \in N_k} f_i(x).$$

Da bismo osigurali globalnu konvergenciju metoda, koristimo linijsko pretraživanje [15]. Klasično linijsko pretraživanje je dato pomoću Armijo pravila:

$$f_{N_k}(x_k + \alpha_k d_k) \leq f_{N_k}(x_k) + \eta \alpha_k d_k^T \nabla f_{N_k}(x_k).$$

Ako je pravac opadajući, onda je Armijo pravilo dobro definisano za funkcije ograničene od dole. Međutim, postoje modifikacije linijskog pretraživanja koje mogu da prihvate i pravce koji nisu opadajući. Konkretno, mi ćemo ocenu rešenja ažurirati pomoću nemonotonog linijskog pretraživanja sa generičkim opadajućim pravcima, kao što sledi:

$$x_{k+1} = x_k + \alpha_k d_k,$$

gde je d_k pravac pretraživanja i α_k dužina koraka. Dozvoljavamo generički pravac pretraživanja d_k za funkciju f_{N_k} koji zadovoljava

$$d_k^T \nabla f_{N_k}(x_k) < 0.$$

Razmotrićemo neke praktične mogućnosti za odabira pravca d_k . Očigledan izbor bi bio $d_k = -\nabla f_{N_k}(x_k)$. Međutim, uključivanje nekih drugih informacija može poboljšati algoritam značajno, pogotovo ako je nemonotono linijsko pretraživanje primenjeno na način kao što sledi u nastavku. Na primer, možemo koristiti spektralni gradijentni pravac u formi $d_k = -z_k \nabla f_{N_k}(x_k)$, gde je z_k pozitivna konstanta koja se dobija pomoću $x_k - x_{k-1}$ i $\nabla f_{N_k}(x_k) - \nabla f_{N_{k-1}}(x_{k-1})$, za više detalja pogledajte [11]. U tom slučaju, centralni čvor treba da čuva prethodnu procenu rešenja x_{k-1} i vrednost prethodnog gradijenta $\nabla f_{N_{k-1}}(x_{k-1})$ za računanje novog pravca pretraživanja, što predstavlja dodatno čuvanje dva vektora dužine n . Alternativno, mogu se koristiti Kvazi-

Njutnovi tipovi pravca pretraživanja kao što je BFGS [15].

Pomoću unutrašnjih iteracija s određuje se dužinu koraka α_k pomoću Armijo tipa pravila. Posebno, kada centralni čvor izračuna d_k , to inicira unutrašnje iteracije $s = 0, 1, \dots$. U svakoj iteraciji s centralni čvor procenjuje probnu tačku $x_k + \beta^s d_k$ i šalje je svim radnicima. Radnici ne menjaju njihova stanja kroz unutrašnje iteracije, tj. u svim unutrašnjim iteracijama oni ostaju u trenutnom stanju. Nakon prijema probne tačke $x_k + \beta^s d_k$, svaki aktivni radnik $i \in N_k$, računa vrednost njegove funkcije $f_i(x_k + \beta^s d_k)$ i šalje tu vrednost nazad centralnom čvoru. Proces se ponavlja dok sledeći uslov nije zadovoljen za neku određenu unutrašnju iteraciju s (ovde je $\alpha_k := \beta^s$):

$$f_{N_k}(x_k + \alpha_k d_k) \leq f_{N_k}(x_k) + \eta \alpha_k d_k^T \nabla f_{N_k}(x_k) + e_k. \quad (2.9)$$

U (2.9), $\{e_k\}_{k \in N_0}$, je pozitivan predefinisani konačno prebrojiv niz,

$$e_k > 0, \sum_{k=0}^{\infty} e_k < \tilde{e} < \infty. \quad (2.10)$$

Generalno, ovo nemonotono linijsko pretraživanje dozvoljava duži korak u poređenju sa Armijo pravilom:

$$f_{N_k}(x_k + \alpha_k d_k) \leq f_{N_k}(x_k) + \eta \alpha_k d_k^T \nabla f_{N_k}(x_k),$$

(koje se od gore navedenog nemonotonog razlikuje samo u tome što je $e_k = 0$), tj. manje probnih tačaka je potrebno i manje računanja $f_i(x_k + \alpha_k d_k)$.

Sada ćemo objasniti kako centralni čvor ažurira kontrolni parametar i postavlja novu vrednost p_{k+1} . Na ažuriranje parametra p_k utiču dve mere napretka algoritma: dm_k i ε_k . Prvu meru napretka u smanjivanju vrednosti funkcije koju minimiziramo definišemo sa:

$$dm_k = -\alpha_k d_k^T \nabla f_{N_k}(x_k). \quad (2.11)$$

Svrha druge mere napretka ε_k , je da proceni grešku koja nastaje zbog korišćenja prosečne uzoračke funkcije f_{N_k} umesto f . Vrednost $\varepsilon_k = \varepsilon_k(p)$ je funkcija koja zavisi od vrednosti sledećeg kontrolnog parametra $p = p_{k+1}$ koja se utvrđuje. Vrednost $\varepsilon_k(p)$ može da bude zavisno od podataka koji su trenutno dostupni kod centralnog čvora, tj. od vrednosti $f_{N_k}(x_k)$ – ovo je označeno u indeksu k od $\varepsilon_k(p)$. Funkcija $p \rightarrow \varepsilon_k(p)$ može biti generička. Kao što se može videti u dokazu [9], jedini tehnički zahtev je da $\varepsilon_k(p) \geq 0$, za svako $p \in [0, 1]$ i uniformno je ograničena od dole sa pozitivnom konstantom, za sve $p \in [0, 1)$. Poželjna svojstvo je da funkcija $\varepsilon_k(p)$ opada sa rastom parametra p .

Sada ćemo razmotriti specifične izbore za p i $\varepsilon_k(p)$. Pretpostavimo da kontrolni parametar p_k uzima vrednosti iz diskretnog skupa $\Pi = \{\pi_1, \pi_2, \dots, \pi_m\}$, gde su $0 < \pi_1 < \pi_2 < \dots < \pi_m = 1$. Broj mogućih vrednosti (m) može biti proizvoljno veliki. Testovi su pokazali da $m \leq N$ daje dovoljno dobre performance [9]. Što se tiče $\varepsilon_k(p)$, jedna od mogućnosti je da se uzme $\varepsilon_k(p) = 1 - p$, što je dopustivo jer smo pretpostavili da p može uzeti samo konačno mnogo diskretnih

vrednosti. Ovo je jednostavan izbor koji se lako računa, ali ne uzima u obzir uticaj trenutne ocene rešenja x_k . Da bismo inkorporirali poslednje informacije, možemo je definisati na sledeći način:

$$\varepsilon_k(p) = \mu \frac{\sigma_k}{\sqrt{pN}}, \quad (2.12)$$

gde je μ pozitivna konstanta i σ_k^2 je varijansa skupa vrednosti $f_i(x_k)$, $i \in N_k$ i definisana je sa

$$\sigma_k^2 = \frac{1}{|N_k|} \sum_{i \in N_k} (f_i(x_k) - \frac{1}{|N_k|} f_{N_k}(x_k))^2 \quad (2.13)$$

Što se tiče ažuriranja kontrolnog parametra, glavna ideja je da se napravi algoritam koji nam omogućava da radimo sa što manje čvorova možemo, posebno na početku optimizacionog procesa. Međutim, da bi se osigurala skoro sigurna konvergencija ka stacionarnoj tački funkcije cilja f , moramo na kraju dostići ceo skup čvorova. To se postiže pomoću pomoćnog niza definisanog sa $\{p_k^{min}\}$. Ove vrednosti će biti donje granice za vrednosti pravog kontrolnog parametra p_k , preciznije, $p_{k+1} \geq p_k^{min}$ za sve k . Ažuriranje pomoćnog niza će biti opisano kasnije [9].

Neka najniža dozvoljena verovatnoća π_1 definiše početnu vrednost zaštite $p_0^{min} := \pi_1$ i uzmimo ovu vrednost za početnu vrednost parametra $p_0 := p_0^{min}$. Da bi ažurirao zaštitu p_k^{min} , centralni čvor treba da čuva i ažurira vektor dužin m koji označavamo sa $F_k = ([F_k]_1, \dots, [F_k]_m)$. Uloga ovog vektora je da prati vrednosti aproksimacije funkcije cilja, za različite vrednosti p_k , tj. za različite nivoe preciznosti. Komponenta $[F_k]_j$ odgovara nivou preciznosti π_j i predstavlja najnižu vrednost aproksimacije funkcije cilja koja je dobijena sa preciznošću π_j u prvih k iteracija. Podsetimo se da kontrolni parametar određuje samo očekivani broj aktivnih čvorova. Dakle, F_k prati različite nivoe očekivane preciznosti. Vrednost svake komponente ovog vektora se inicijalno postavlja na $+\infty$. Napomenimo da je dozvoljeno uzeti m nezavisno od N .

3 Implementacija

U poglavlju „Dodatak - Matlab kodovi” se nalazi implementacija programa koji vrši automatsku analizu zadate vremenske serije i kao rezultat daje zadatih (k) predikcija date serije. Kako se problem najmanjih kvadrata može posmatrati kao problem distribuiranih najmanjih kvadrata (2.7), za ocenu parametara modela implementiran je algoritam za rešavanje problema distribuirane optimizacije koji je definisan u delu 2.5.1.

Nemonotono linijsko pretraživanje koje smo definisali u (2.9) je implementirano sa $e_0 = 0.1$ i $e_k = e_0 k^{-1.1}$. Ostali parametri koji se koriste u algoritmima su definisani na sledeći način: $\theta_k = (k+1)p_k$, $\gamma_k = (Ne^{\frac{1}{k}})^{-1}$, $\nu_1 = \frac{1}{\sqrt{N}}$, $p_0^{min} = 0.1$, $\eta = 10^{-4}$, $\beta = 0.5$. Za skup dopustivih vrednosti parametra p_k smo koristili skup $\{\frac{1}{N}, \frac{2}{N}, \dots, 1\}$ sa korakom $\frac{1}{N}$.

Za pravac pretraživanja smo koristili BFGS metod koji do pravca pretraživanja d_k u iteraciji k dolazi rešavanjem problema $B_k d_k = -\nabla f_{N_k}(x_k)$, gde je B_k aproksimacija hesijana koju ažuriramo u svakoj iteraciji k na sledeći način:

$$B_{k+1} = B_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{B_k s_k s_k^T B_k^T}{s_k^T B_k s_k}, \quad (2.14)$$

pri čemu smo za B_0 uzeli jediničnu matricu, a s_k i y_k su definisani na sledeći način $s_k = x_{k+1} - x_k$ i $y_k = -\nabla f_{N_k}(x_{k+1}) - \nabla f_{N_k}(x_k)$.

Ako primetimo da je B_k^{-1} simetrična matrica i da su $y_k^T B_k^{-1} y_k$ i $s_k^T y_k$ skalari, ažuriranje matrice B_k pomoću (2.14) se može efikasnije izračunavati na sledeći način:

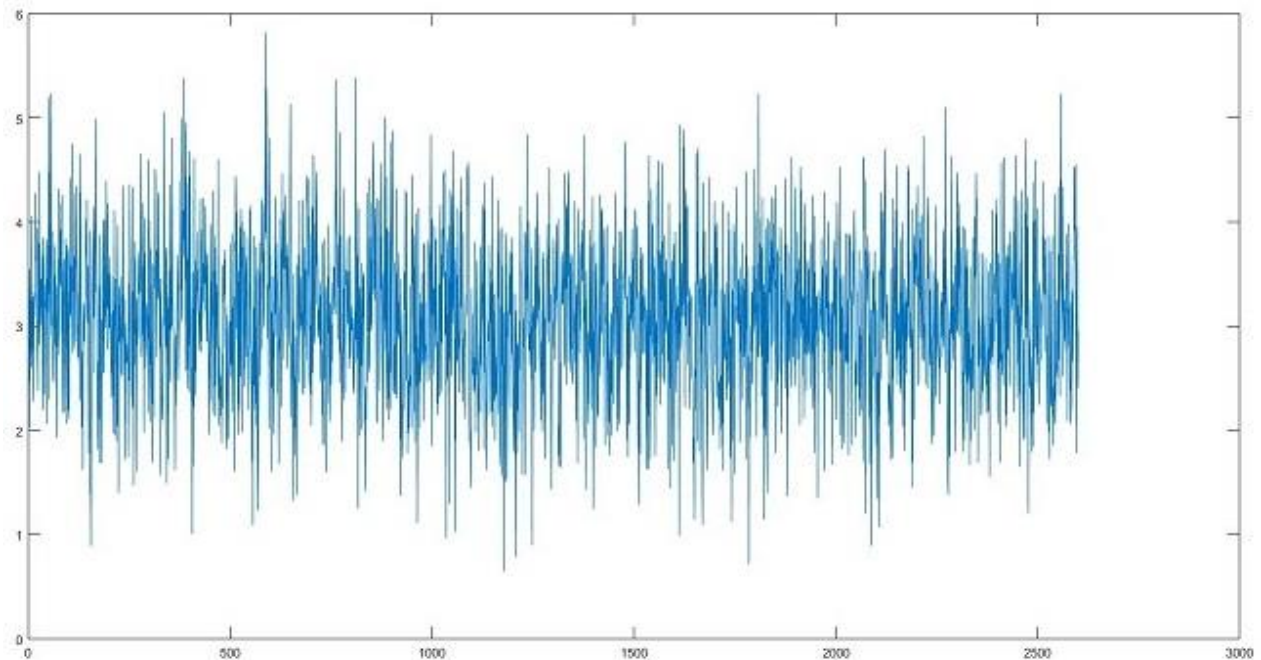
$$B_{k+1}^{-1} = B_k^{-1} + \frac{(s_k^T y_k + y_k^T B_k^{-1} y_k)(s_k s_k^T)}{(s_k^T y_k)^2} - \frac{B_k^{-1} y_k s_k^T + s_k y_k^T B_k^{-1}}{s_k^T y_k}.$$

4. Rezultati testiranja

Kako je glavni cilj ovog rada implementacija algoritma koji sa varijabilnim izborom parametra p_k postiže sličan kvalitet predikcija kao da su svi radnici aktivni u svakoj iteraciji tj. $p_k = 1$ u svim iteracijama, uz manje troškove izračunavanja, upoređićemo njihove troškove izračunavanja pomoću vrednosti FEV - broj evaluacija funkcija f_i , pri čemu se svaka komponenta ∇f_i računa kao jedna evaluacija funkcije f_i .

Serijska za testiranje je generisana pomoću sledećih komandi:

```
model = arima('Constant',2, 'AR',{0.1,0.25}, 'Variance',.5);  
y=simulate(model,2600);  
plot(y)
```



Rezultati algoritma za $p_k=1$ u svim iteracijama:

```
pred1=predictions(y,15)  
Kalibraciona serija je reda stacionarnosti i(0).  
Cela serija je reda stacionarnosti i(0).  
Max red modela koji ispitujemo je 2  
Vrednost FEV je 1053000  
Koef modela reda 1 su:  
    0.0998    2.7826
```


Postoji autokorelacija u rezidualima
Vrednost FEV je 16264000
Koeff modela reda 2 su:
0.2420 0.0761 2.1084

Reziduali su nezavisni
Odabrani red modela koji ispitujemo je 2
Vrednost FEV je 16264000
Koefficijenti ocenjenog modela su:
0.2420 0.0761 2.1084

pred1 =

2.9158
3.0454
3.0457
3.0771
3.0795
3.0873
3.0885
3.0905
3.0909
3.0914
3.0916
3.0917
3.0918
3.0918
3.0918

Rezultati algoritma za varijabilne vrednosti p_k :

pred2 =predictions(y,15)
Kalibraciona serija je reda stacionarnosti i(0).
Cela serija je reda stacionarnosti i(0).
Max red modela koji ispitujemo je 2
Vrednost FEV je 434157
Koeff modela reda 1 su:
0.1018 2.7759

Postoji autokorelacija u rezidualima
Vrednost FEV je 746888
Koeff modela reda 2 su:
0.2418 0.0763 2.1084

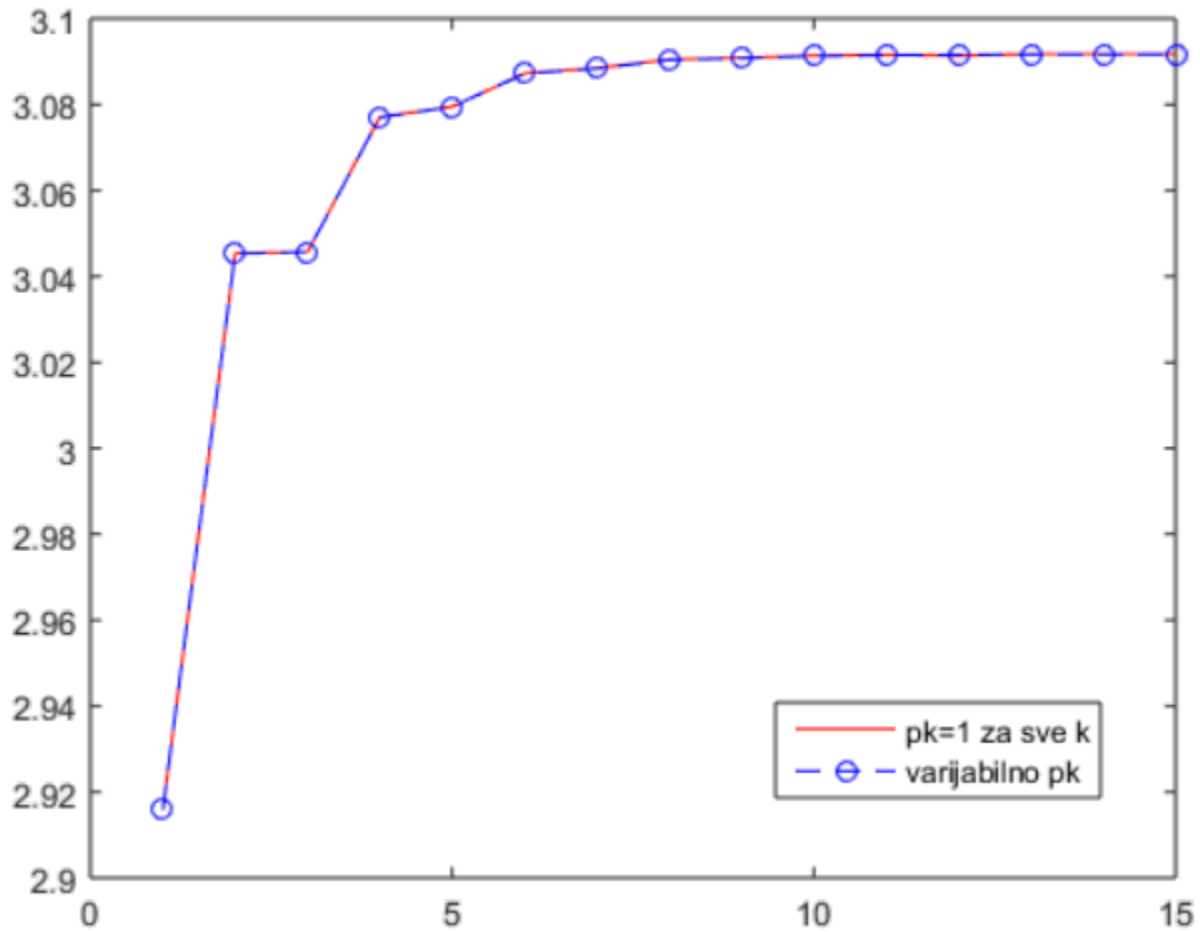
Odabrani red modela koji ispitujemo je 2
Vrednost FEV je 477932
Koefficienti ocenjenog modela su :
0.2417 0.0763 2.1084

pred2 =

2.9159
3.0453
3.0457
3.0770
3.0795
3.0872
3.0884
3.0904
3.0908
3.0913
3.0915
3.0916
3.0917
3.0917
3.0917

Na osnovu gore navedenih rezultata možemo videti da se ocenjeni parametri razlikuju tek na trećoj decimali, dok se vrednosti FEV znatno razlikuju, u slučaju gde je $p_k=1$ u svim iteracijama ukupan FEV = 33581000, što je više od 20 puta više od ukupne vrednosti FEV=1658977 algoritma sa varijabilnim p_k . Da bismo pokazali sličnost predikcija prikazaćemo ih na istom grafiku.

```
>> plot(pred1, 'red')  
>> hold on  
>> plot(pred2, 'b--o')
```



Slika 2: Grafički prikaz dobijenih predikcija. Pri čemu su pred1 predikcije dobijene ocenom parametara pomoću Algoritma 3 sa konstantnom vrednošću parametra $p_k = 1$ u svim iteracijama, a pred2 su predikcije dobijene ocenom parametara pomoću Algoritma 3 sa varijabilnim vrednostima parametra p_k . Ovim smo hteli da pokažemo da se sa varijabilnim vrednostima parametra p_k dobija rešenje skoro istog kvaliteta kao i sa konstantnim $p_k = 1$ u svim iteracijama, uz znatno manje troškove izračunavanja.

Zaključak

Cilj ovog rada je implementacija algoritma koji metode specifične za rešavanje problema distribuirane optimizacije primenjuje na autoregresivne modele koji se koriste u analizi vremenskih serija. Kako smo na osnovu testiranja došli do rezultata koji nam pokazuju da se varijabilnim izborom parametra p_k vrednost FEV smanjuje, dolazimo do zaključka da implementirani algoritam ostvaruje uštedu u vremenu u primerima gde je izračunavanje funkcija i izvoda dominantan trošak. Time se ubrzao proces ocene parametara, a time i formiranja predikcija što je veoma značajno u situacijama koje zahtevaju odgovore u „realnom vremenu“.

Kao rezultat rada, implementiran je algoritam u programskom paketu Matlab, koji vrši automatsku analizu zadate vremenske serije, što podrazumeva testiranje stacionarnosti zadate serije, diferenciranje ukoliko je potrebno, ispitivanje autokorelacije, određivanje reda modela na osnovu „in-sample and out-of-sample” test strategije, upoređuje relativne greške svih modela koji su zadovoljili testne kriterijume i bira model sa najmanjom relativnom greškom i računa zadati broj predikcija ukoliko serija prati AR model.

Kako se pokazalo na seriji za testiranje performansi, uz znatno niže troškove izračunavanja implementirani algoritam daje gotovo iste rezultate kao i algoritam koji konstantno radi sa punim gradijentom u svim iteracijama.

Dodatak – Matlab kodovi

Implementacija Algoritma 1

Algoritam 1 koji je definisan u 2.5.2 za ažuriranje donje granice p_k^{min} i vrednosti vektora $[F_k]_j$:

```
function [pm,Fknew]=ALGO1(pk,pk1,fNk1,pkmin,tetak,gamak,Fk)

%-----s1-Ažuriranje-pk-min-----

    if(pk<pk1 && Fk-fNk1<tetak);
        pm=min(1,pkmin+gamak);
    else
        pm=pkmin;
    end
    if pk<pk1;

%-----s2-Ažuriranje-[Fk]_j-----

        Fknew=min(fNk1,Fk);
    else
        Fknew=Fk;
    end
end
```

Implementacija Algoritma 2

Algoritam 2 koji je definisan u 2.5.3 ažuriranje kontrolnog parametra p_k :

```
function res=ALGO2(dmk,pkmin,pk,ekpk,v1,mi,vark,N)
%pk uzima vrednosti iz diskretnog skupa Pskup=[p11,....,pim] gde je
%0<p11<...<pim=1

Pskup=1/1000:1/1000:1;

% m<=N uzeli smo da je m=1000
%uslov pk+1>=pkmin
%-----s1-----
```

```

if(pk<pkmin)
    podskup=[];
    for j=1:1000
        if Pskup(j)>=pkmin
            podskup=[podskup,Pskup(j)];
        end
    end
    pknew=min(podskup); %min elemnt vektora mogucih verovatnoca
else

%-----s2-----

%1)

if dm k==ekpk
    pknew=pk;
else

%2)

if dm k>ekpk
    podskup1=[];
    for j=1:1000
        if Pskup(j)<=pk && Pskup(j)>=pkmin &&
dm k<=mi*(sqrt(vark))/sqrt(Pskup(j)*N)
            podskup1=[podskup1,Pskup(j)];
        end
    end
    if length(podskup1)>0
        pknew=max(podskup1);
    else
        podskup2=[];
        for j=1:1000
            if Pskup(j)>=pkmin
                podskup2=[podskup2,Pskup(j)];
            end
        end
        pknew=min(podskup2); %min elemnt vektora mogucih verovatnoca
    end
end

%3)

if dm k<ekpk %i)
    if dm k>=v1*ekpk
        podskup3=[];

```

```

for j=1:1000
    if Pskup(j)>=pk && dmk>=mi*(sqrt(vark))/sqrt(Pskup(j)*N)
        podskup3=[podskup3,Pskup(j)];
    end
end
if length(podskup3)>0
    pknew=min(podskup3); %min elemnt vektora mogucih verovatnoca
else
    pknew=1;
end
else %ii)
    pknew=1;
end
end
end
end
end
res=pknew;

```

Implementacija Algoritma 3

Implementacija simulacije odlučivanja radnika o njihovoj aktivnosti koju ćemo koristiti u *Algoritmu 3*:

```

function res=actinact(N,p)
r=[];
for i=1:N
    r=[r;binornd(1,p)];
%simulacija odlucivanja radnika, da li ce biti aktivan ili ne
end
res=r;
end

```

Algoritam 3 koji je definisan u 2.5.4 - glavni algoritam za ocenu parametara AR(p) modela.

```

function res=Algo3(rniz,p0min,x0,beta,psi)

%-----s1-----

p0=p0min; %p0min=0.1
x=x0; %pocetna vrednost resenja
n=size(x0,1);
p=n-1; %gledamo koji je red modela zadat-p
N=1000; %broj radnika 1000

```

```

L=size(rniz,1);
y=rniz(n:L);
t=[];
for i=n:L %pravimo tm matricu
    ti=[];
    for j=0:(p-1)
        ti=[ti rniz(i-p+j)];
    end
    t=[t;[ti 1]];
end
step= floor(size(t,1)/1000);
borders=1:step:step*1000+1;%size(t,1); %delimo seriju najmanjih kvadrata na
1000 delova
if borders(length(borders))~=size(t,1)+1
    borders(length(borders))=size(t,1)+1;
end

%-----s2-----

a=actinact(N,p0); FEV=sum(a)+sum(n*a);
braktivnihrad=[sum(a)];
%disp(['Br aktivnih radnika je', num2str(sum(a))])
pvalues=[p0];
pminvalues=[p0];
xmat=[];
for i=1:N %punimo matricu dostavljenim podacima od aktivnih radnika, svaki
radnik ima svoju vrstu
    if a(i)==1 %ako je radnik aktivan upisujemo njegove gradijente i
vrednosti funkcija
        j=borders(i):borders(i+1)-1;
        xmat=[xmat;[Fls(x,t(j,:),y(j)),t(j,:)]]
RADIMO UPDATE MATRICE SA GRADIJENTIMA, NEAKTIVNI IMAJU NULU
    end
end

%-----

B=eye(p+1); %pocetna aproksimacija hesijana
Fk=inf(1,1000);
k=0;
Nk=sum(a);

%-----vrtimo iteracije dok ne zadovoljimo uslove -----

while 1
%-----s3-----

```



```

%negativnigradijent
%dk=-xmat(:,2:size(xmat,2))'*xmat(:,1); %negativni gradijent
%-----BFGS-----

dk=-B*(xmat(:,2:size(xmat,2))'*xmat(:,1));

%-----s4 Step size with Armijo-type rule -----

m=0;
alfak=beta^m;
ek=0.1;
while
fnk(x+alfak*dk,N,borders,t,a,y)>(fnk(x,N,borders,t,a,y)+psi*alfak*(dk'*(xmat(
(:,2:size(xmat,2))'*xmat(:,1)))+ek)
    m=m+1;
    alfak=beta^m;
    ek=ek*m^(-1.1);
end
sk=m;
alfak=beta^sk;

%-----s5-----

xold=x; %cuvamo staro x za update B za BFGS
x=x+alfak*dk;
%dmk=-alfak*dk'*-dk; kad je negativni gradijent dk
dmk=-alfak*dk'*(xmat(:,2:size(xmat,2))'*xmat(:,1));

%-----s6 Algoritam2 -----

%-----ek(pk)-----

mi=3.2;
vark=(1/sum(a))*sum((xmat(:,1).^2-(1/sum(a))*sum(xmat(:,1).^2)).^2);
pk=pvalues(length(pvalues));%poslednje koricenso p
ekpk=mi*(sqrt(vark))/sqrt(p*N);
v1=1/sqrt(N); %ovo cemo proslediti
pkmin=pminvalues(length(pminvalues));%poslednje koricenso pkmin
pknew=ALGO2(dmk,pkmin,pk,ekpk,v1,mi,vark,N);%mi,vark,N zbog racunanja
epsilon
pvalues=[pvalues,pknew];%dodajemo novo p

%-----s7-----

if p0~=pknew;

```

```

        a=actinact(N,pknew);
        Nk=sum(a);
    End
    FEV=FEV+sum(a)+sum(n*a);
    braktivnihrad;
    %disp(['Br aktivnih radnika je ', num2str(sum(a))])

%-----s8-----

    xmatnew=[];
    for i=1:N
        if a(i)==1 %ako je radnik aktivan
            j=borders(i):borders(i+1)-1;
            xmatnew=[xmatnew;[Fls(x,t(j,:),y(j)),t(j,:)]]; %RADIMO UPDATE
MATRICE SA GRADIJENTIMA
        end
    end
    if ismember(pknew,pvalues(1:end-1)) %proveravamo da li se pk+1 koristi prvi
    put

%-----s9-----

    pkmin=pminvalues(length(pminvalues));
    tetak=(k+1)*pk;
    gamak=(N*exp(1/k))^(-1);
    fNk1=fnk(x,N,borders,t,a,y);
    [pkmin,Fknew]=ALGO1(pk,pknew,fNk1,pkmin,tetak,gamak,Fk(pknew*1000));
    pminvalues=[pminvalues,pkmin];
    Fk(pknew*1000)=Fknew;
end

%-----update B za BFGS-----

    q=x-xold; %|
    g=xmatnew(:,2:size(xmatnew,2))'*xmatnew(:,1)-
xmat(:,2:size(xmat,2))'*xmat(:,1); %|
    B=B+(q'*g+g'*B*g)*(q*q')/((q'*g)^2)-(B*g*q'+q*g'*B)/(q'*g);

%-----

    xmat=xmatnew;
    k=k+1;
    if Nk==N && norm(xmat(:,2:size(xmat,2))'*xmat(:,1))<0.5
        disp(['Vrednost FEV je ', num2str(sum(FEV))])
        break
    end
end

```

```
end
res=x;
```

Implementacija algoritma za automatsku analizu vremenske serije

```
function res=predictions(rniz,k)
original=rniz;
calibration=rniz(1:(floor(0.8*size(rniz,1)/1000))*1000);
validation=rniz((floor(0.8*size(rniz,1)/1000))*1000+1:size(rniz,1));

%----- adf test i diferenciranje nestacione serije -----

s=0; % brojac diferenciranja
slas=calibration;
for p=1:2
    if adftest(slas)==0 %H0 nestacionarna serija, test vraca 0 ako ne moze da
se odbaci H0, tj. 1 ako se odbacuje
        slas=deltas(slas,1);
        s=s+1;
    end
end

if s<2
    disp(['Kalibraciona serija je reda stacionarnosti i(', num2str(s),')'])
end
if s==2
    disp('Kalibraciona serija ima vise od jednog jedinicnog korena, program
je predvidjen za stacionarne serije ili serija sa jednim jedininim korenom')
    return
end

l=0;% brojac diferenciranja
slasorigin=original;
for o=1:2
    if adftest(slasorigin)==0 %H0 nestacionarna serija, test vraca 0 ako ne
moze da se odbaci H0, tj 1 ako se odbacuje
        slasorigin=deltas(slasorigin,1);
        l=l+1;
    end
end
if l<2
    disp(['Cela serija je reda stacionarnosti i(', num2str(l),')'])
end
if l==2
```

```

disp('Cela serija ima vise od jednog jedinicnog korena, program je
predvidjen za stacionarne serije ili serija sa jednim jedininim korenom')
return
end

%----- ispitivanje autokorelacije -----

m=ceil(log(size(slas,1))); %dokle najdalje proveravamo autokorelaciju
alfa=0.05;
autocorel=tratio(slas,m,alfa);
pmax=0;%p max poslednji stat znacajan koji treba ispitati.
for i=1:m
    if autocorel(5,i)==1
        pmax=i;
    end
end
disp(['Max red modela koji ispitujemo je ', num2str(pmax)])

%----- odredjivanje reda modela -----

%-----in sample and out off sample metodom -----

valsize=size(validation,1); % duzina validacionog uzorka,da znamo koliko nam
treba predikcija
predikcije=[];
adekvatni=[];
for p=1:pmax
    xzfinal=ARocenakoeff1(slas,p); %ocenjujemo koef za sve p od 1 do pmax
    disp(['Koef modela reda ', num2str(p), 'su:'])
    disp(xzfinal)
    [boxtest at]=adekvatnostAR1(slas,xzfinal); % za svaki ocenjeni model
proveravamo adekvatnost
    %ad=basicstat(at,0.05);
    if boxtest(3)<=boxtest(4) % reziduali su nekorelisani
        adekvatni=[adekvatni,p];
        predikcije=[predikcije,kpredikcijeAR(slas,xzfinal,p,valsize)];
        disp('Reziduali su nezavisni')
    else
        disp('Postoji autokorelacija u rezidualima')
    end
end
predikcijes=[];
if s>0;
    for u=1:size(predikcije,2)
        predikcijes=[predikcijes ,delta1(predikcije(:,u),calibration)]; %ako
smo radili diferencirnje odmotavamo predikcije

```

```

    end
else
    predikcijes=predikcije; %ako je serija inicijalno stacionarna predikcije
smo vec izracunali
end
if size(predikcijes)==[0,0]
    poruka = ['Ni jedan model reda <=', num2str(pmax), ' nije adekvatan.'];
    disp(poruka)
    return
end
p=poredjenje(predikcijes,validation); %biramo red modela koji daje najmanju
relativnu gresku na validacionom uzorku
p=adekvatni(p);
disp(['Odabrani red modela koji ispitujemo je ', num2str(p)])

%-----

xzfinal=ARocenakoef1(slas,p); %ponovo ocenjujemo koeficijente ali sada samo
za izabrani red modela p
disp(['Koeficijenti ocenjenog modela su :'])
disp(xzfinal)
knovihpredikcija=kpredikcijeAR(slasorigin,xzfinal,p,k);% pravimo predikcije
ovaj put pocevsi od poslednjeg poznatog podatka koji se nalazi u validacionom
uzorku
kkonacnepredikcije=[];
if l>0
    for u=1:size(knovihpredikcija,2)
        kkonacnepredikcije=[predikcijes
,delta1(knovihpredikcija(:,u),original)]; %ako smo radili diferencirnje
odmotavamo predikcije
    end
else
    kkonacnepredikcije=knovihpredikcija; %ako je serija inicijalno
stacionarna predikcije smo vec izracunali
end
res=kkonacnepredikcije;

```

Implementacija pomoćnih algoritama

U nastavku su navedeni algoritmi koji su korišćeni u implementaciji algoritma za automatsku analizu vremenske serije.

```

function res=acf(rniz,l)
r=mean(rniz);

```

```

T=size(rniz,1);
d=sum((rniz-r*ones(T,1)).^2);
r1=rniz((l+1):T,:);
r2=rniz(1:(T-l),:);
rb=r*ones(T-l,1);
g=(r1-rb)'*(r2-rb);
res=g/d;

function res=acfgraf(rniz,m)
y=[];
for l=1:m
    rl=acf(rniz,l);
    y=[y;rl];
end
%plot(y,'g*');
res=y;

function [res at]=adekvatnostAR1(rniz,x)
T=size(rniz,1);
p=size(x,1)-1;
t=tmatrix(rniz,p);
pred=Fm(x,t);
y=rniz(p+1:T,1);
at=y-pred;
m=ceil(log(T));
g=sum(abs(sign(x)));
res=ljungbox(at,m,0.05,g);
end

function res=ARocenal(rniz,p)
x0=zeros(p+1,1);
p0min=0.1;
beta=0.5;
psi=0.0001;
xz=Algo3(rniz,p0min,x0,beta,psi);
res=[xz'];
se=standls(rniz,xz);
res=[res;se'];
test=abs(xz./se);
hip=[];
for i=1:p+1
    if test(i)<=1.96
        hip=[hip;0];
    else
        hip=[hip;1];
    end
end
end

```

```

res=[res;hip'];
xzfinal=xz.*hip;
res=[res;xzfinal'];

function res=ARocenakoeff1(rniz,pz)
%Red modela
T=size(rniz,1);
%Npr.
m=ceil(log(T)); %zakucali
%pz=PACF1(rniz,alfa,lambda,eps,m);
%Odredjivanje koeficijenata
resm=ARocenal(rniz,pz);
res=resm(4,:)';

function res=delta1(predfinal,yniz)
k=size(predfinal,1);
T=size(yniz,1);
y=yniz(T);
res=[];
for i=1:k
    y=predfinal(i)+y; %iterativno u svakom koraku uzima poslednju vrednost i
    %lepi predikciju
    res=[res;y];
end

function res=deltas(rniz,s)
T=size(rniz,1);
res=[];
for i=s+1:T
    r=rniz(i)-rniz(i-s);
    res=[res;r];
end

function res=fi(x,ti)
res=ti*x;

function res=Fls(x,t,y)
m=size(t,1);
r=[];
for i=1:m
    r=[r;fi(x,t(i,:))-y(i)];
end
res=r;

function res=flsAR1(rniz,x)
T=size(rniz,1);
p=size(x,1)-1;

```

```

t=tmatrixa(rniz,p);
pred=Fm(x,t);
y=rniz(p+1:T,1);
res=0.5*norm(pred-y)^2;

function res=Fm(x,t)
res=t*x;

function res=fnk(x,N,borders,t,a,y)
niz=[];
    for i=1:N
        if a(i)==1 %ako je radnik aktivan
            for j=borders(i)+1:borders(i+1)
                niz=[niz;Fls(x,t(j,:),y(j))]; %RADIMO UPDATE MATRICE SA
GRADIJENTIMA, NEAKTIVNI IMAJU NULU
            end
        end
    end
res=0.5*sum(niz.^2);

```


Literatura

- [1] Tsay, R. S. Analysis of Financial Time Series. 2. izd. Njujork: A John Wiley & Sons, Inc., 2005.
- [2] Maddala, G. S., Introduction to Econometrics. 2. izd. Njujork: Macmillan Publishing Company, Inc., 1992.
- [3] Steven J. Janke, Frederick C. Tinsley Introduction to Linear Models and Statistical Inference, John Wiley & Sons, Inc., 2005.
- [4] CAMPBELL, John Y. The econometrics of financial markets, Princeton: University Press, 1997.
- [5] Mladenović, Zorica Primenjena analiza vremenskih serija, Beograd: Centar za izdavačku delatnost Ekonomskog fakulteta, 2012.
- [6] S. Boyd and L. Vandenberghe: Convex Optimization, Cambridge University Press, 2004.
- [7] D. Bertsekas, Nonlinear Programming, Athena Scientific, 2004.
- [8] D. Bertsekas and J. Tsitsiklis Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, 1989.
- [9] Dragana Bajović, Dušan Jakovetić, Nataša Krejić, Nataša Krklec Jerinkić Parallel stochastic line search methods with feedback for minimizing finite sums, Novi Sad: Faculty of Technical Sciences and Faculty of Sciences, University of Novi Sad, 2019., http://www.optimization-online.org/DB_FILE/2016/07/5548.pdf
- [10] Danijela Rajter-Ćirić Verovatnoća, Univerzitet u Novom Sadu, Prirodno matematički fakultet, 2009.
- [11] Lozanov Crvenković Z. Statistika, Univerzitet u Novom Sadu, Prirodno-matematički fakultet, 2012.

- [12] E. G. Birgin, J. M. Martínez, M. Raydan Nonmonotone Spectral Projected Gradient Methods on Convex Sets *SIAM J. Optim.* 10(4), 2006., pp. 1196-1211.
- [13] R. H. Byrd, G. M. Chin, W. Neveitt, J. Nocedal On the Use of Stochastic Hessian Information in Optimization Methods for Machine Learning, *SIAM J. Optim.*, 21(3), 2011., pp. 977-995.
- [14] R. H. Byrd, G. M. Chin, J. Nocedal, Y. Wu Sample size selection in optimization methods for machine learning, *Mathematical Programming*, 134(1), 2012., pp. 127-155.
- [15] J. Nocedal, S. J. Wright, *Numerical Optimization*, Springer, 1999.

Biografija



Dejan Vučenović rođen je 04.01.1991. godine u Vrbasu. U Srbobranu je završio osnovnu školu Vuk Karadžić. 2005. godine upisao je opšti smer gimnazije Svetozar Miletić, takođe u Srbobranu. Nakon toga, 2009. godine upisuje Prirodno-matematički Fakultet na Univerzitetu u Novom Sadu, smer primenjena matematika – matematika finansija. Master na istom smeru upisao je 2015. godine. Od marta 2017. godine do septembra 2018. godine bio je zaposlen u Uniqa sTech u Beogradu, na poziciji analitičar u okviru odeljenja za modelovanje. Od septembra 2018. godine do jula 2019. godine bio je zaposlen u Komercijalnoj banci u Beogradu na poziciji analitičara podataka u okviru odeljenja za modelovanje kreditnih rizika. Od jula 2019. godine zaposlen je na poziciji biznis analitičara za implementacije softvera u Adacta fintech u Beogradu.