

UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Finančna matematika – 1. stopnja

Martin Praček

Skriti markovski modeli v časovnih vrstah

Delo diplomskega seminarja

Mentor: izr. prof. dr. Damjan Škulj

Ljubljana, 2019

KAZALO

1. Uvod	4
Slovar strokovnih izrazov	4
2. Markovski modeli	5
3. Skriti markovski modeli	6
3.1. Zgodovina modela	7
3.2. Zahteve za model	8
3.3. Priprava in trening modela	9
3.4. Natančna določitev začetnih vrednosti	11
3.5. Trenutno stanje	12
3.6. Generiranje poti v skitem markovskem modelu	12
4. Uporaba	14
4.1. Procesiranje govora	14
4.2. Uporaba v biologiji in biokemiji	14
4.3. Napovedovanje prevoza	14
4.4. Prepoznavanje lastnoročne pisave	14
4.5. Kriptoanaliza	14
4.6. Prepoznavanje akcij	15
5. Časovne vrste	15
5.1. Finančna časovna vrsta	16
6. Uporaba skritih markovskih modelov v finančnih časovnih vrstah	17
7. Praktični primer	18
7.1. Splošna analiza kakovosti modela	19
7.2. Ekstremni dogodki	19
7.3. Napoved	20
Literatura	20

Skriti markovski modeli v časovnih vrstah

POVZETEK

V povzetku na kratko opiši vsebinske rezultate dela. Sem ne sodi razlaga organizacije dela – v katerem poglavju/razdelku je kaj, pač pa le opis vsebine.

Hidden Markov Models in Time Series

ABSTRACT

Prevod zgornjega povzetka v angleščino.

Math. Subj. Class. (2010): navedi vsaj eno klasifikacijsko oznako – dostopne so na www.ams.org/mathscinet/msc/msc2010.html

Ključne besede: skriti markovski modeli, časovne vrste, slučajni proces

Keywords: hidden markov models, time series, stochastic process

1. UVOD

Skriti markovski modeli so modelacijsko orodje, ki nam omogoča zelo široko uporabo. V mojem diplomskem seminarju se bom najbolj posvetil uporabi v finančni analizi. V prvem poglavju se bom posvetil različnim načinom uporabe skritih markovskih modelov, v drugem pa bom natančno opisal matematično ozadje le teh. Na koncu bom natančno predstavil še lasten primer uporabe, kjer si bom pomagal z različnimi programskimi jeziki in okolji, od Mathematice do R in MatLaba.

SLOVAR STROKOVNIH IZRAZOV

Skriti markovski model Skriti markovski model je statistični model, prek katerega določimo stanja, v katerih se nahajamo, verjetnost nahajanja v tem stanju, verjetnost prehoda med stanji za nek nabor podatkov.

Gaussova mešanica Gaussova mešanica je porazdelitvena funkcija z gostoto, ki jo lahko zapišemo kot tehtano povprečje normalnih gostot;

$$f(x) = \sum_{j=1}^M c_j N(x; \mu_j, \sigma_j^2)$$

. Veljati mora še, da je

$$\sum_{j=1}^M c_j = 1$$

in da je $N(x; \mu_j, \sigma_j^2)$ gostota normalne porazdelitvene funkcije.

Verjetnostni grafični model PGM Verjetnostni grafični model je model, za katere graf prikaže pogojno odvisnost med slučajnimi spremenljivkami.

Akaikejev informacijski kriterij Akaikejev informacijski kriterij oziroma AIC je merilo relativne kakovosti modela glede na preostale modele za iste podatke. Naj velja, da imamo podatke s k ocenjenimi parametri in \hat{L} maksimalna vrednost funkcije verjetja. AIC izračunamo kot

$$AIC = 2k - 2\ln(\hat{L})$$

AIC primerja več različnih modelov po kakovosti, kjer je najboljši tisti z najmanjšo vrednostjo.

Baum-Welchov algoritem Je algoritem, s katerim poiščemo neznane parametre skritih markovskih modelov.

Dinamična bayesianska mreža Je bayesianska mreža, v kateri so povezane spremenljivke, ki ležijo v dveh sosednjih časovnih korakih.

Pričakovan primankljaj Naj bo X donos portfelja nekoč v prihodnosti in naj velja da je $0 < \alpha < 1$, kjer je α stopnja zaupanja. Pričakovan primankljaj ES_α je definiran kot

$$ES_\alpha = -\frac{1}{\alpha} \int_0^\alpha VaR_\gamma(x) d\gamma$$

kjer je VaR_γ **Value at risk**.

2. MARKOVSKI MODELI

Skriti markovski modeli so ena izmed vrst bolj splošne skupine modelov, ki jo imenujemo markovski modeli. Gre za modele, kjer se stanja slučajno spreminjajo. Za modele velja, da je njihovo stanje v času t odvisno le od stanja v času $t - 1$. To imenujemo tudi markovska lastnost, po kateri so ti modeli tudi poimenovani.

Definicija 2.1. Markovska lastnost Naj bo $(\Omega, \mathcal{F}, \mathbb{P}, \mathcal{F}_s)$ verjetnostni prostor s filtracijo za neko urejeno množico I . Naj bo (S, \mathcal{S}) merljiv prostor. (S, \mathcal{S}) merljiv slučajni proces $X = \{X_t : \Omega \rightarrow S\}_{t \in I}$, ki je prilagojen na filtracijo, ima markovo lastnost, če za vsak $A \in \mathcal{S}$ in vsak $s, t \in I$, kjer velja s, t velja

$$\mathbb{P}(X_t \in A \mid \mathcal{F}_s) = \mathbb{P}(X_t \in A \mid X_s)$$

To definicijo lahko v diskretnem primeru še razširimo, če velja še, da je $I = \mathbb{N}$. Potem velja

$$\mathbb{P}(X_n = x_n \mid X_{n-1} = x_{n-1}, \dots, X_0 = x_0) = \mathbb{P}(X_n = x_n \mid X_{n-1} = x_{n-1})$$

V posebnem lahko za markovske modele rečemo še, da so homogeni, če velja

$$P(X_n \mid X_{n-1}) = P(X_1 \mid X_0)$$

Markovska lastnost je nam pove, da je slučajni proces "brez spomina". To pomeni, da pot do stanja ni pomembna, pomembno je le stanje, v katerem se nahajamo. Naravno se nam tudi postavi vprašanje, zakaj je ta lastnost koristna. Izkaže se, da nam omogoča reševanje problemov, ki jih drugače v primernem času ne bi mogli rešiti.

Zaradi svoje široke uporabnosti markovske modele lahko ločimo na:

- prediktivno modeliranje, kjer želimo iz statističnih podatkov izračunati najbolj verjeten dogodek glede nanašne podatke. Ta model se lahko uporablja ne glede na čas v katerem se je dogodek zgodil, saj lahko modeliramo prihodnje dogodke ravno tako kot pretekle dogodke, policija npr. uporablja to vrsto modela za določitev osumljencev za storjeni zločin.
- verjetnostno napoved, kjer ne napovemo le najbolj verjetnega dogodka, ampak določimo verjetnosti vseh dogodkov kot popoln sistem dogodkov. Najbolj očiten primer tega so športne stavnice, kjer lahko npr. stavimo za eno izmed ekip stavimo na zmago, remi ali poraz.

Glede na to, kako je postavljen markovski model pa ločimo: Modeliranje markovskih

	V celoti opazovan	Le delno opazovan
Avtonomen	Markovska veriga	Skriti markovski model
Kontorliran	Markovski proces odločanja	Delno opazovan proces

modelov spada pod slučajno programiranje. Gre za sistem, ki nam pomaga pri modeliziranju optimizacije za modele, ki vključujejo negotovost. Ta negotovost je navadno predstavljena kot slučajna spremenljivka, ki jo v praksi aproksimiramo.

Primer 2.2. Primer žabe v ribniku Zamislimo si, da imamo ribnik z lokvanji, v katerem so se naselile žabe. Vsaka izmed žab je lahko v vsakem trenutku v vodi, na

lokvanju ali pa zunaj vode. To imenujemo stanja markovskega modela S . Poimenujmo vektor stanj

$$S = (Voda, Lokvanj, Zemlja).$$

Na ribnik pogledamo le v nekih diskretnih časih, z enakim razmikom, na primer ob koncu vsake minute. Odločimo se za opazovanje le ene izmed žab, za katero nas zanima, kje se bo nahajala v naslednjem trenutku.

Do sedaj smo opazovali splošen primer za vse tipe markovskih modelov. Na tem mestu pa moramo vedeti, ali je naša opazovana žaba dresirana, in se bo odzvala na naše opazovanje. Če to velja, potem imamo kontroliran markovski model, ki je lahko delno opazovan ali markovski proces odločanja.

Predstavljajmo si, da lahko vsakič ko pogledamo, našo žabo poljubno premaknemo ali pa pustimo na mestu. Žaba lahko nato poljubno skače naprej. Če poznamo prehodno matriko med stanji bomo v tem primeru imeli markovski proces odločanj. V primeru ko pa to ne velja, imamo delno opazovan proces odločanja.

Na drugi strani pa imamo avtonomen model, kjer lahko ločimo če je proces v celoti opazen ali so neka stanja skrita. Če stanja niso skrita, tako poznamo prehodno matriko A , ki nam pove, s kakšno verjetnostjo se premikamo med posamičnimi stanji. V tem primeru imamo markovsko verigo, ki je le poseben primer markovskih procesov odločanja, kjer žabe nikoli ne prestavimo. Preprost primer markovskih verig predstavljajo slučajni sprehodi.

Recimo, da v našem primeru velja, da je prehodna matrika oblike:

$$\begin{bmatrix} 0.5 & 0.3 & 0.2 \\ 0.1 & 0.5 & 0.4 \\ 0.4 & 0.2 & 0.4 \end{bmatrix}$$

Matrika je torej znana, in navadno predpostavimo, da se verjetnosti prehoda iz enega stanja v drugega ne spreminjajo. Vsota po vsakem stolpcu in vrstici je vedno enaka ena, to je

$$\sum_{i=1}^n a_{ij} = \sum_{j=1}^n a_{ij} = 1$$

saj predstavlja verjetnosti prehoda iz enega stanja v drugo. Tako je a_{ij} verjetnost prehoda iz stanja j v stanje i . Velja še, da je n število stanj in je v našem primeru enako 3, očitno pa velja tudi, da je matrika velikost $n \times n$ in da so vsi elementi a_{ij} iz intervala $[0, 1]$.

V tem primeru lahko torej, v vsakem trenutku predvidimo, kam bo najverjetneje skočila žaba v naslednjem trenutku. Na drugi strani pa imamo še skrite markovske modele.

Za skrite markovske modele pa velja, da same prehodne matrike načeloma ne poznamo, jo pa lahko določimo iz podatkov, ki jih imamo za naš problem. Tako bi za našo žabo dovolj dolgo opazovali, in si zapisovali njene pozicije glede v vsakem trenutku opazovanja. Tako bi dobili vektor opazovanj O . Vektor O je naš osnovni podatek, iz katerega nato izpeljemo celoten model. Iz tega bi najprej pogledali, kakšna so možna stanja sistema. Iz tega bi nato poizkušili ugotoviti, kako izgleda prehodna matrika. Kako to naredimo, si bomo pogledali v nadaljevanju. \diamond

3. SKRITI MARKOVSKI MODELI

Skriti markovski model je statistični markovski model, kjer predpostavljamo, da je modelirani sistem markovski proces z skritimi stanji. Gre torej za tip modela,

kjer lahko razberemo rezultat, ne moremo pa ugotoviti, kakšna je bila funkcija, ki nam ga je dala.

Gre za najbolj preprosto vrsto dinamične bayesijske mreže. Tu gre za vrsto bayesijske mreže, v kateri so povezane spremenljivke, ki ležijo v dveh sosednjih časovnih korakih. Tako lahko vrednost v času t izračunamo prek vrednosti v času $t - 1$ ter notranih **regresorjev**. Bayesijske mreže so verjetnostni grafični model, ki je idealen za to, da za nek dogodek za vsak možen vzrok določimo verjetnost, da ga je ta povzročil.

Ideja za delovanje skritih markovskih modelov izhaja iz dela Andrewa Viterbija na področju komunikacije. V komunikacijskih storitvah šumi namreč predstavljajo veliko težavo, zato se je ameriško-italijanski električni inženir domislil algoritma, s katerimi bi iz seta podatkov izločili šume. To si lahko predstavljamo kot da bi si na primer želeli na fiksnem mestu v električnem tokokrogu določiti naboj $Q(t)$ glede na naboj ob času s , kjer $s < t$. Zaradi napake pri merjenju $Q(s)$ ne moremo točno določiti, saj zaznamo neko zmoteno različico $\widehat{Q}(s)$. Želimo si torej način, ki bi čim bolj učinkovito izločil te napake.

Skrite markovske modele navadno predstavimo na diskretnih množicah in na diskretnem času, a lahko definicijo tako razširimo, da velja tudi za zvezne množice in čas. V tem primeru dobimo tri plasten slučajni proces, katerega največja težava je velika računska zahtevnost, poleg tega pa nam lahko veliko število parametrov prinese nestabilnost. Vendar se s tem v moji diplomski nalogi ne bom ukvarjal.

Prav tako je možno s skritimi markovskimi modeli predstaviti procese, odvisne od več slučajnih spremenljivk. Vendar tak model s seboj prinese več težav. Prve se pokaže v računski zahtevnosti, saj moramo v vsakem koraku določiti inverz in determinanto $N \cdot M$ matrik Σ_{ij} velikosti NA , kjer je NA število različnih spremenljivk, na primer število vrednostnih papirjev katerih proces aproksimiramo. Če je ta NA velik, je lahko izračun inverza in determinante računsko in časovno zahteven. Drugi problem se pokaže v samem računanju, ki je podrobneje razložen v 3.3. To velja, ker so determinante Σ_{ij} zelo majhne, kar posledično privede do velik verjetnosti $P(O|\lambda)$. Ob vsakem novem koraku iteracije tako pridobimo nov λ in posledično novo pogojno verjetnost. Ta verjetnost lahko v tem primeru zelo hitro naraste, in tako se iteracija prehitro konča. Če je NA prevelik, pa se včasih ne da izvesti niti prvega koraka. Modeliranje skritih markovskih modelov z več spremenljivkami je tako v teoriji možno, a ni nujno dobro v praksi.

3.1. Zgodovina modela. Ideja skritih markovskih modelov se v svoji prvi fazi začne z ruskim matematikom Andrejem Andrejevičem Markovom, ki je za časa svojega življenja med 1865 – 1922 razvil idejo markovskega procesa in verige. Prve teoretične rezultate markovskih verig je svetu predstavil 1906, 1913 pa je izračunal zaporedje črk ruščine. Slednji izračun je opravil na besedilu Jevgenij Onjegin ruskega pesnika Aleksandra Sergejeviča Puškina. Markov je želel s tem dokazati veljavnost šibkega zakona velikih števil zaradi spora z drugim ruskim matematikom Pavlom Nekrasom, a je na ta način odprl novo vprašanje v matematiki.

Skriti markovski modeli zahtevajo veliko numeričnega računanja, in zato ne preseneča, da se je nadaljni razvoj začel šele z širšo uporabo računalnikov. Po pomembnem delu, ki so ga opravili von Neumann, Turing in Conrad Zuse so se znanstveniki začeli pospešeno ukvarjati z implementacijo primernih algoritmov. Velik korak k temu je pripomogel Claude Shannon s svojim delom Matematična teorija komunikacije.

Za implementacijo skritih markovskih modelov je bilo v začetku potrebnih kar nekaj razvojev algoritmov. Prvi algoritem je bil algoritem maksimizacije pričakovanega (Expectation-maximization), ali, kot ga med drugim poznamo, EM-algoritem so uporabljali med drugim že Laplace, Gauss in drugi, vendar je do dokončnega poimenovanja prišlo šele leta 1977. Demster, Laird in Rubin so algoritem poimenovali in razložili splošno teorijo, ki deluje v ozadju procesa. Andrew Viterbi je drugi pomemben del skritih markovskih modelov implementiral leta 1967 kot algoritem za dekodiranje konvolucij čez šume v komunikaciji. Pri tem algoritmu deluje princip dinamičnega programiranja, ki najde najbolj verjetno zaporedje stanj, glede na dano opazovano zaporedje dogodkov. Viterbijev algoritem je podrobneje opisan pod 1. Celotnih modelov pa prav gotovo ne bi bilo brez še enega Rusa, in sicer Ruslana Leontijeviča Stratonoviča, ki je prvi opisal rekurzijo naprej - nazaj leta 1960. To je delal na primeru optimalnega nelinearnega problema filtracije.

Za glavnega avtorja pa velja Leonard E. Baum. Skupaj z Lloyd R. Welchom sta okrog leta 1970 razvila Baum-Welchov algoritem, kjer gre za poseben primer posplošenega EM-algoritma. Prav Baum-Welchov algoritem je tisti korak, pri katerem velja, da je bila dokončno izpeljana teorija za skritimi markovskimi modeli. Algoritem uporablja EM-algoritem da najde cenilko po metodi največjega verjetja glede na dano zaporedje podatkov. Delovanje algoritma je podrobno razloženo v 3.3, njegove zahteve pa v 3.2.

Uporabe skritih markovskih modelov na področju financ in ekonomije pa ne bilo brez dela ameriškega ekonometrista Jamesa Hamiltona, ki je predlagal, da bi model, katerega stanj in prehodov med njimi ne poznamo opišemo z markovskim modelom.

3.2. Zahteve za model. Kot vsak matematični model, ima tudi skriti markovski model svoje zahteve.

- (1) Prva zahteva je, da lahko rezultate, ki jih model producira, opazujemo v ekvidistančnih časih, torej je razlika med dvema poljubnima zaporednima časoma opazovanja $t - 1$ in t vedno enaka, na primer d . Rezultate imenujemo signali ali opazovanja.
- (2) V vsakem izmed časov t je sistem lahko v enem izmed N stanj. Ta stanja so S_1, S_2, \dots, S_N . Vsako izmed stanj S_i je slučajna spremenljivka, ki je lahko zvezna ali diskretna, a njenega porazdelitvenega zakona ne poznamo. Ta stanja so v času t neznana.
- (3) V vsakem izmed časov t ne vemo, v kakšnem stanju se nahajamo, vemo le, kakšni so rezultati našega procesa v tem času. Zato potrebujemo dodatni slučajni proces $Q = (Q_t)_{t=1,2,\dots}$, ki nam pove, v kakšnem stanju je sistem v času t . Tako velja, da je signal O_t dan s stanjem slučajnega procesa Q , v odvisnosti od gostote verjetnostne porazdelitve Gaussove mešanice. Signali našega procesa so torej rezultati slučajnega procesa Q .
- (4) Vektor verjetnosti začetnih stanj je označen z Π , in vsota njegovih elementov je enaka 1.
- (5) V vsakem času velja, da bodisi sistem spremeni svoje stanje, bodisi ostane isto. Verjetnost prehoda v vsako stanje je določena s prehodnimi verjetnostmi, podanimi v matriki A^t , kjer verjetnost prehoda iz stanja i v času t v stanje j v času $t + 1$ simbolizira element a_{ij}^t .
- (6) Vsota vsakega stolpca vsake matrike A^t je enaka 1.
- (7) Ker govorimo o markovskem modelu, bo stanje v času $t + 1$ odvisno le od stanja v času t , ne pa od celotne zgodovine.

S temi predpostavkami bi lahko zmodelirali, a jih imamo še nekaj, ki nam sam model še precej olajšajo.

- (1) Predpostavimo lahko, da so vse prehodne matrike A^t enake, torej neodvisne od časa t . Enolično prehodno matriko torej lahko označimo A .
- (2) Signal v času t je odvisen le od stanja modela v tem času; torej so slučajne spremenljivke v času t odvisne le od tega.

Naši signali so torej odvisni od stanja S . Ta stanja nam vrnejo rezultat v odvisnosti od verjetnostnih porazdelitev podanih z različnimi parametri. Le te predstavimo kot mešanice normalnih porazdelitev in jih imenujemo tudi Gaussove mešanice. Gaussovo mešanico definiramo kot tehtano povprečje normalnih porazdelitvenih funkcij, glej slovar strokovnih izrazov. Vsako stanje i iz nabora vseh stanj je torej podano z porazdelitveno gostoto

$$b_i = \sum_{j=1}^M c_{ij} N(x; \mu_j, \sigma_j^2)$$

Gaussove mešanice so primerne, ker lahko zelo dobro aproksimirajo vsako končno zvezno porazdelitev, poleg tega pa se znebimo tveganja, ki nam ga v tem modelu predstavlja normalna porazdelitev. To tveganje predstavlja dejstvo, da je normalna porazdelitev simetrična; to pa pogosto ne drži za procese v realnem svetu, na primer za donose v financah. Za te večkrat drži, da so "left-skewed", torej da je večji del manjši od povprečja.

Tako potrebujemo za sestavo modela še:

- število mešanic M
- matriko C , ki predstavlja koeficiente c_{ij} , ki so faktorji v Gaussovi mešanici. To matriko imenujemo tudi Izhodno matriko.
- matrika Γ , kjer μ_{ij} predstavlja pričakovano vrednost mešanice j v stanju i , ter
- matriko Σ , kjer σ_{ij} predstavlja varianco mešanice j v stanju i .

3.3. Priprava in trening modela. Ko vemo kaj potrebujemo, lahko začnemo s pripravo naših modelov. Preden začnemo s določanjem parametrov našega modela, moramo najprej dobro preučiti naš problem, saj je to znanje ključno za dobro reševanje problema. Kot prvo stvar moramo najprej izbrati set podatkov, na katerem se bo potekal tako imenovan trening sistema. Ti podatki morajo biti zbrani na na podobni enoti; če na bo na primer zanimale cena delnice podjetja, ki se ukvarja z predelavo pomarančnega soka, za trening ne bomo vzeli cene delnic metalurškega podjetja.

Podatke, ki smo jih zbrali moramo nato urediti; določimo časovne trenutke z enakim razponom, kot ga želimo imeti v našem modelu, in v teh trenutkih $t \in (1, \dots, T)$ določimo opazovanja O .

Ko imamo podatke zbrane, moramo najprej določiti število stanj N ter število mešanic M . Tu gre tudi za ključna problema priprave skritega markovskega modela. V praksi število stanj včasih določimo v skladu s potrebno aplikacijo, kot na primer

v 4.2, oziroma z vizualnim ogledom točk grafa iz zgodovinskih podatkov.

Včasih pa lahko to določimo preko Akaikejevega informacijskega kriterija, ki primerja več različnih modelov, ki lahko določijo z različnim številom stanj N in izbere najboljšega. AIC ocenjuje relativno kakovost modela. Podobno deluje tudi BIC, to je Bayesianski informacijski kriterij.

Za pravilno določitev števila mešanic M pa navadno izberemo razvrščanjem v skupine (k -mean clustering). Gre za metodo vektorske kvantitizacije, kjer želimo n opazovanj razvrstiti v k skupin, kjer vsako opazovanje razvrstimo v skupino, z najbližjim povprečjem.

Ostale parametre navadno označimo z $\lambda = (\Pi, A, C, \Gamma, \Sigma)$. Cilj treninga modela je, da parametre nastavimo tako, da bo verjetnost, da so bila vsa opazovanja določena tudi s strani modela največja, $P(O|\lambda)$. Pri tem mora veljati, da sta število mešanic M ter število stanj N že znana.

Naravno se pojavi vprašanje, kako to določimo. Izkaže se, da je učinkovit sistem *naprej – nazaj*. Določimo spremenljivki *naprej* $\alpha_t(i)$, ki predstavlja verjetnost, da so se zgodila opazovanja od o do t in stanje i , ter *nazaj* $\beta_t(i)$, ki predstavlja verjetnost, da se bodo zgodila opazovanja od t do T pri stanju i . Definirati pa rabimo le eno izmed njiju, saj gre pri $\beta_t(i)$ le alternativno metodo za $\alpha_t(i)$. $\alpha_t(i)$ definiramo kot:

$$\alpha_1(i) = \pi_i b_i(O_1) = \pi_i b_i = \sum_{k=1}^M c_{ik} N(O_1; \mu_j, \sigma_j^2)$$

Naslednje $\alpha_{t+1}(j)$ lahko nato induktivno izračunamo kot

$$\alpha_{t+1}(j) = b_j(O_{t+1}) \sum_{i=1}^N \alpha_t(i) a_{ij}$$

Iz tega sledi, da je $P(O|\lambda) = \sum_{i=1}^N \alpha_T(i)$. Na podoben način izračunamo tudi $\beta_t(i)$. β je inicializirana z

$$\beta_T(i) \forall i \in 1, \dots, N$$

indukcija pa poteka prek

$$\beta_t(i) = \sum_{j=1}^N \beta_{t+1}(j) a_{ij} b_j(O_{t+1})$$

$\beta_t(i)$ sicer ni nujen za izračun $P(O|\lambda)$, a ga nujno potrebujemo za trening našega modela.

Skozi celotno točko 3.3 smo se obnašali, kot da lahko vse podatke iz λ kar izračunamo. Vendar to ne drži v popolnosti. Res, ko je enkrat *naprej – nazaj* vzpostavljen, se nam dozdeva, da gre le še za numerično operacijo. Vendar težave nastopijo preden naš postopek začne z delom. Problem nastopi ko moramo v *naprej* oceniti začetne parametre λ .

Ne poznamo analitičnega načina, kako bi lahko ta problem rešili, vendar lahko najdemo tak λ , da lahko našo verjetnost $P(O|\lambda)$ lokalno maksimiziramo. Tu lahko uporabimo različne algoritme, najbolj pa je uporabljen Baum-Welchov.

Pri tem moramo vsak paramater λ najprej pogledati posebej. Izkaže se, da začetni vrednosti za prehodno matriko A in začetni vektor Π nista pomembni, če nista ravno ničelna. Za Π tako lahko vzamemo vektor, kjer so vse vrednosti enake $1/N$, kjer je N število stanj.

Več problemov nam povzročajo tri postavke z zvezno porazdelitvijo, to je C , Σ in

Γ . Dobra začetna ocena le teh je nujna za kakovost modela. Tudi tu se izkaže, da lahko, podobno kot pri oceni števila mešanic M ozremo na razvrščanjem v skupine. Ta postopek nam sicer ne da globalnega minimuma, ki bi ga mogoče lahko dobili na drugačen način, a se v praksi izkaže za dobrega. Za C to pomeni, da bo element $c_{ij} = 1/k$ za vsak par ij , kjer je k število šopov povprečij. Pričakovane vrednosti in variance nato pridobimo iz vrednosti šopov povprečij.

Z znanimi začetnimi vrednostmi se lahko spustimo v maksimiziranje $P(O|\lambda)$. Osnovati moramo zaporedje $\lambda = \lambda_t, t \in 0, \dots$, da bo veljalo

$$P(O|\lambda_{i+1}) \geq P(O|\lambda_i)$$

Za to zaporedje velja, da konvergira k lokalnemu maksimumu.

3.4. Natančna določitev začetnih vrednosti. Zanima nas torej, kako bomo osnovali zaporedje λ , ki ga potrebujemo za maksimizacijo $P(O|\lambda)$. Izkaže se, da za maksimizacijo le tega potrebujemo še nekaj dodatnih formul, ki so rezultat Baum-Welchovega algortma. Z njimi definiramo večfazni iterativni proces, pri katerem popravljamo vrednosti parametrov do konvergence.

Kot prvo potrebujemo $\xi_t(i, j)$, ki nam pove verjetnost, da smo v času t v stanju i in v času $t + 1$ v stanju j .

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(O_t)\beta_{t+1}(j)}{P(O|\lambda)}.$$

Vsota $\sum_{t=1}^N \xi_t$ nam pove pričakovano število prehodov iz stanja i v stanje j . Druga je $\gamma_t(i)$, ki nam pove verjetnost, da smo v času t v stanju i .

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)}$$

Vsota $\sum_{t=1}^N \gamma_t(i)$ nam pove pričakovano število prehodov iz stanja i .

Zadnja, tretja pa je $\gamma_t(j, k)$, ki nam pove verjetnost, da smo v času t v stanju j , in k ta mešanica predstavlja O_t .

$$\gamma_t(j, k) = \gamma_t(j) \frac{c_{jk}N(x; \mu_{jk}, \sigma_{jk}^2)}{\sum_{m=1}^M c_{jm}N(x; \mu_{jm}, \sigma_{jm}^2)}$$

Iz teh podatkov lahko nato popravimo izračun elementov:

$$\begin{aligned} \overline{\Pi}_i &= \gamma_1(i) \\ \overline{a_{ij}} &= \frac{\sum_{t=1}^{T-1} \xi_t}{\sum_{t=1}^{T-1} \gamma_t(i)} \\ \overline{c_{jk}} &= \frac{\sum_{t=1}^T \gamma_t(j, k)}{\sum_{t=1}^T \sum_{m=1}^M \gamma_t(j, m)} \\ \overline{\mu_{jk}} &= \frac{\sum_{t=1}^T \gamma_t(j, k) O_t}{\sum_{t=1}^T \gamma_t(j, k)} \end{aligned}$$

Ko je to določeno, nam zaporedje λ , določenih s temi parametri da lokalni maksimum.

3.5. Trenutno stanje. Zadnja stvar, ki jo moramo narediti, preden začnemo z določanjem prihodnjih stanj je določitev trenutnega stanja gospodarstva, torej stanja v zadnjem času, na katerem je naš model treniral. Za določitev le tega uporabimo tako imenovani Viterbijev algoritem. Le ta je oblikovan tako, da nam vrne zaporedje Q , ki maksimizira $P(Q|O, \lambda)$.

Za delo s tem algoritmom moramo definirati $\delta_t(i)$, ki za vsako stanje i vrne največjo

Algorithm 1: Viterbijev algoritem

Result: Najbolj verjetno zaporedje opazovanj $X = (x_1, x_2, \dots, x_N)$

Input:

- Prostor opazovanj $O = \{o_1, o_2, \dots, o_N\}$
- Prostor stanj $S = \{s_1, s_2, \dots, s_K\}$
- Nabor začetnih verjetnosti $\Pi = (\pi_1, \pi_2, \dots, \pi_K)$, kjer $\pi_i == s_i$
- Zaporedje opazovanj $Y = (y_1, y_2, \dots, y_T)$, tako da je $y_t == i$, če je opazovanje v času t o_i
- Prehodna matrika A velikosti $K \times K$, kjer stanje A_{ij} pove verjetnost prehoda iz stanja s_i v stanje s_j
- Izhodno matriko B velikosti $K \times N$, kjer stanje B_{ij} pove verjetnost, da opazimo o_j iz stanja s_i

```

for vsako stanje  $j \in \{1, 2, \dots, K\}$  do
  |  $T_1[j, 1] \leftarrow \pi_{B_{jy_1}}$ 
  |  $T_2[j, 1] \leftarrow 0$ 
end
for vsako opazovanje  $i = 2, 3, \dots, T$  do
  | for  $j \in \{1, 2, \dots, K\}$  do
  | |  $T_1[j, i] \leftarrow \max_k (T_1[k, i-1] \cdot A_{kj} \cdot B_{jy_i})$ 
  | |  $T_2[j, i] \leftarrow \arg \max_k (T_1[k, i-1] \cdot A_{kj} \cdot B_{jy_i})$ 
  | end
end
 $z_T \leftarrow \arg \max_k (T_1[k, T])$ 
 $x_T \leftarrow s_{z_T}$ 
for  $i \leftarrow T, T-1, \dots, 2$  do
  |  $z_{i-1} \leftarrow T_2[z_i, i]$ 
  |  $x_{i-1} \leftarrow s_{z_{i-1}}$ 
end
return  $X$ 

```

verjetnost vzdolž poti v času t . Prek $\delta_t(i)$ nato induktivno izvedemo algoritem.

Viterbijev algoritem nam vrne p^* , ki je največja verjetnost in q_T^* , ki nam pove stanje v času T , ki nam to verjetnost vrne.

3.6. Generiranje poti v skritem markovskem modelu. V celoti lahko sedaj predstavimo delovanje skritega markovskega modela na primeru generiranja ene poti. V tem primeru je pot na primer cenovni proces, ki ga bo glede na naš model opravil finančni instrument.

Glede na naše podatke, ki jih označimo kot $O = (O_1, \dots, O_T)$ ocenimo parametre našega modela $\lambda = (\Pi, A, C, \Gamma, \Sigma)$. Začetne vrednosti elementov Π, A in C predpostavimo kot enake, za Γ in Σ pa ocenimo prek razvrščanja v skupine. Te parametre

rekurzivno izboljšujemo, dokler nismo zadovoljni z določeno $P(O|\lambda)$. Parametre, ki smo jih dobili na ta način lahko za isti set podatkov uporabimo poljubno mnogokrat. V naslednjem koraku določimo trenutno stanje našega modela, torej stanje iz katerega začeli. V tem trenutku določimo še čas, do katerega želimo oceniti naš proces, naj bo to recimo T_T .

Naslednji korak si lahko predstavljamo kot zanko. Če predpostavimo, da naš parameter t teče od časa T do časa T_T , v vsakem koraku glede na stanje, v katerem se nahajmo, izberemo stanje, v katerega se bomo premaknili v naslednjem koraku. V vsakem koraku še shranimo stanje, v katerem smo bili. Na ta način dobimo pot, ki jo generira skriti markovski model.

Ta postopek lahko ponovimo večkrat, kar se izkaže precej uporabno v primeru Monte Carlo simulacije.

4. UPORABA

Skriti markovski modeli so zelo široko uporabno orodje za modeliranje. Načini uporabe se zelo razlikujejo, od uporabe v financah do določevanja genoma. Ekonomskim, torej tistim, ki jih delujejo kot napovedovalci cen vrednostnih papirjev v prihodnosti, se bom najgloblje posvetil v naslednjem delu, zato sedaj raje pogledimo ostale načine uporabe.

4.1. Procesiranje govora. Prva ideja Leonarda E. Bauma za njegov algoritem je bila prav uporaba v procesiranju govora. Ena najbolj široko uporabljenih načinov uporabe pa je procesiranje govora za posamične glasovne enote. Gre za sistem, kjer želimo prepoznati posamične izgovorjene besede, ne moremo pa ga uporabiti za splošen govor, saj algoritem ni implementiran za žvezen govor. Ideja tega algoritma je razviti najboljše možne aproksimacijske algoritme, da lahko skriti markovski model filtrira naključne šume, zvoke na najboljši možen način.

Da bomo procesiranje lahko zmodelirali moramo najprej določiti slovar glasov, iz katerega vemo, da bo glas prišel. Tu uporabljam besedo glas, ker ni nujno, da bo prepoznan glas dejansko beseda; eden izmed glasov bi lahko bil tudi zgolj črka A. Velja omeniti tudi, da na ta način, le z nekaj adaptacijami, deluje tudi Siri.

4.2. Uporaba v biologiji in biokemiji. Gre za eno izmed prvih uporab skritih markovskih modelov. Ker se je uporaba na tem področju začela precej zgodaj tudi ne preseneča, da je uporaba precej široka.

V zadnjem času se skrite markovske modele vedno več uporablja tudi za modeliranje bioloških in biokemijskih procesov, med katerimi je najbolj znan primer modeliranja proteinov v celični membrani, ki ga bom malo opisal, poleg tega pa med drugim še za **napovedovanje genov?**

Čeprav lahko nekatere proteine in njihovo delovanje znotraj celične membrane enostavno predstavimo, to ne velja za vse. Tako imenovani β -**barrel** membranski proteini zahtevajo več dela. Da bi ugotovili njihovo delovanje in ga primerjali z delovanjem v vodi topnih proteinov so Bagos, Liakoupos et al. razvili model ki to naredi.

Z delom potem nadaljujemo podobno kot pri procesiranju govora, le da je naš začetni slovar tu dolg 20 znakov, toliko kolikor je aminokislin.

4.3. Napovedovanje prevoza. Gre za sistem, kjer želimo za prihodnost napovedati, koliko vozil bo neko prometno infrastrukturo v prihodnosti.

4.4. Prepoznavanje lastnoročne pisave.

4.5. Kriptoanaliza. S tem se je ukvarjajo različni storokovnjaki.

4.6. Prepoznavanje akcij.

5. ČASOVNE VRSTE

Definicija 5.1. Časovna vrsta množica opazovanj x_t , vsako opazovano ob časih t znotraj nekega časovnega intervala.

Definicija 5.2. Model časovne vrste za opazovane podatke x_t je slučajni proces X_t , kjer velja, da so x_t realizacije tega slučajnega procesa v časih t .

Vrednosti delnic po dnevih, število uporabnikov spletne strani po urah, prebivalstvo neke države po letih, . . . To je le nekaj izmed mnogih primerov uporabe časovnih vrst. Časovne vrste nam dajo lepšo idejo o trendu podatkov, ki jih želimo analizirati. Enostavno lahko vidimo, ali je npr. cena jagod odvisna od letnega časa ter ali se cena jabolk skozi leta spreminja.

Časovna analiza je lahko analitična, kjer preučujemo podatke iz preteklosti, ter prediktivna, kjer želimo napovedati neke trende. Analize lahko poteka na več načinov, od linearnih in nelinearnih modelov do modelov z eno spremenljivko in modelov z več spremenljivkami. Prav zaradi njihove široke uporabnosti se je razvilo zelo veliko modelov za njihovo analizo, začensi leta 1970 z delom Boxa in Jenkinsa na področju samoregresivnega integriranega tekočega povprečja, danes znanega pod kratico ARIMA.

Časovne vrste so po definiciji odvisne od časa. Prav zato je pomembno, da se opazovanj med seboj medčasovno ne menja. Opazovanja x_t so tako razporejena po časih, ko so bila zaznana, torej naraščajočem času t .

V realnosti pogosto velja, da so zaporedna opazovanja med seboj odvisna. To je predvsem očitno, če pomislimo na najbolj tipičen primer finančne časovne vrste, in sicer vrednostnega procesa cene neke delnice. Če vidimo, da ima v času t delnica ceno p , si lahko navadno mislimo, da bo v času $t + 1$ cena blizu p . Glede na čase, v katerih gledamo rezultate naše časovne vrste v grobem ločimo 3 tipe:

- Časovne vrste v zveznih časih. Tu pridobivamo podatke za vsak trenutek znotraj nekega časovnega intervala. Dober primer je tu EKG, kjer v vsakem trenutku preiskave naprava izrisuje graf električne napetosti proti času.

V analizi finančnih časovnih vrst za ta tip uporabljamo Black-Scholesov model. Gre za podvrsto parabolične diferencialne enačbe, katere rešitev nam da predvideno vrednost evropskih opcij *call* in *put* v času, ki nas zanima.

- Časovne vrste z diskretnimi vrednostmi kjer pridobivamo vrednosti, ki lahko zavzamejo le diskretne vrednosti, torej največ števno neskončno različnih možnosti.

Tipičen primer bi bil število klicev med 7.00 in 8.00 znotraj Ljubljane oziroma število nesreč znotraj enega dneva na avtocesti med Postojno in Ljubljano.

- Zadnja vrsta so diskretne časovne vrste, torej vrste, ki vrednosti vzamejo le v določenih časovnih trenutkih, npr. konec vsake ure, konec dneva, . . . Tukaj lahko vrednosti zavzamejo kakršnokoli vrednost, pomembno je le, da te vrednosti vzamemo ob pravih trenutkih. Te vrste so uporabne tudi za aproksimiranje časovnih vrst v zveznih časih. Prav vrste takega tipa so tiste, ki jih lahko analiziramo z skritimi markovskimi modeli.

S finančnimi vrstami se srečujemo vsak dan in v najrazličnejših oblikah. Zaradi njihove raznolikosti je izbira pravega modela za njihovo analizo ključna. V našem

primeru bomo za njihovo analizo uporabili skrite markovske modele.

Navadno je glavni namen analize časovnih vrst napovedovanje trendov, vendar se je to področje z razvojem predvsem podatkovnega rudarjenja razširilo tudi na druga področja.

5.1. Finančna časovna vrsta.

Definicija 5.3. Finančna časovna vrsta je časovna vrsta, kjer so opazovanja x_t vrednosti finančnega instrumenta v času t .

Finančna časovna vrsta je tako zaporedje opaženih vrednosti nekega finančnega instrumenta. Analiza finančnih časovnih vrst se ukvarja z teorijo in prakso določanja vrednosti finančnih instrumentov skozi čas. Zaradi elementa negotovosti, kot je na primer volatilitnost finančnih instrumentov, jo moramo obravnavati ločeno od vseh ostalih časovnih vrst. Velja namreč da ravno ta negotovost prikaže pomembnost statistične teorije in metod, izpeljanih iz le te, v analizi finančnih časovnih vrst. Z finančnimi časovnimi vrstami se ukvarjajo statistiki ter kvantitativni finančniki. V nasprotju z ostalimi finančnimi vrstami se lahko zaradi mikrostrukture trga zgodi, da imajo finančne časovne vrste posebne lastnosti in obliko. Tako se pogosto predpostavlja, da so donosi R normalno porazdeljeni z pričakovano vrednostjo μ in varianco σ^2 , $R \sim N(\mu, \sigma^2)$. Vendar to ne velja, saj lahko hitro ugotovimo, da donos ne bo nikoli manjši od -1 . Če predpostavimo, da je P_t cena instrumenta v času t , potem je očitno, da le ta nikoli ne bo manjša od 0. Potem je donos R_t definiran kot

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

kar pomeni, da je

$$-1 = \frac{0 - P_{t-1}}{P_{t-1}} \leq R_t$$

V prvi polovici 20. stoletja se je v povezavi z finančnimi trgi osnovala tako imenovana hipoteza o učinkovitem trgu, ki pravi, da če cene popolnoma predstavljajo pričakovanja in informacije vseh udeležencev trga, potem so njihove spremembe nepričakovane. To lahko povežemo tudi s teorijo slučajnih procesov. Teorija martingalov pravi namreč, da je najboljša ocena jutrišnje cene današnja cena. Zato v finančnih časovnih vrstah večjo težo damo podatkom, ki so nam zgodili kasneje.

6. UPORABA SKRITIH MARKOVSKIH MODELOV V FINANČNIH ČASOVNIH VRSTAH

Glavna ideja uporabe skritih markovskih modelov v finančni časovni sledi iz osnovne ideje Andrewa Viterbija 1, in sicer analiziranje signalov, ki jih prejmemo tako, da izločimo šume v prejetem signalu.

Aplikacija skritih markovskih modelov v finančne časovne vrste je sorazmerno enostavna. Naše izbrane podatke o ceni nekega instrumenta določimo kot zaporedje opazovanj O . Na podlagi tega določimo število stanj ter začetne ocene vseh parametrov.

Primer 6.1. Problem izbire portfelja

Eno izmed klasičnih vprašanj, ki sem nam postavijo je v finančni optimizaciji je problem optimalne izbire portfelja. To vprašanje se da rešiti tudi z uporabo linearnega programiranja in skritih markovskih modelov.

Predpostavimo, da imamo v trenutnem času kapital M , ki ga lahko vložimo v N različnih vrednostnih papirjev. Odločiti se moramo, kako bomo naš kapital razdelili, da bomo maksimizirali svoj donos in hkrati minimizirali tveganje.

Vsak vrednostni papir ima donos R_j , kjer velja $j \in 1, \dots, N$. Za vsak j je donos slučajna spremenljivka, katere porazdelitev aproksimiramo z diskretno slučajno spremenljivko R_j . Določimo še vektor $x = (x_1, \dots, x_N)$, kjer nam x_j pove, kakšen delež kapitala smo vložili v posamezen vrednostni papir, to je

$$x_j = \frac{M_j}{M}$$

če velja $\sum_{j=1}^N M_j = M$. Ob koncu investicije bomo torej zaslužili

$$R_x = \sum_{j=1}^N x_j R_j$$

kjer je R_x slučajna spremenljivka.

To nalogo lahko predstavimo kot linearni program s pomočjo pogojne vrednosti ob padcu CVaR. CVar je mera tveganja, ki nam oceni tržno ali kreditno tveganje portfelja. Tako pridemo do optimizacijskega problema oblike

$$\begin{aligned} \min \quad & CVaR_\alpha(R_x) \\ \text{p.p.} \quad & E(R_x) \geq d \end{aligned}$$

kjer je d prej določen zahtevan donos, α pa stopnja zavrnitve.

Skriti markovski modeli se v igro vključijo, ko moramo aproksimirati donose. Kot že vemo, je R_j diskretna aproksimacija donosa s končnim številom stanj. Ta stanja določimo kot različna možna stanja finančnega instrumenta na naših podatkih. Nato znanih podatkov določimo še vhodne in izhodne matrike za vse slučajne spremenljivke.

Z zbranim znanjem in podatki izvedemo simulacije več različnih scenarijev, čemur na koncu sledi še reševanje dejanskega linearnega programa. \diamond

7. PRAKTIČNI PRIMER

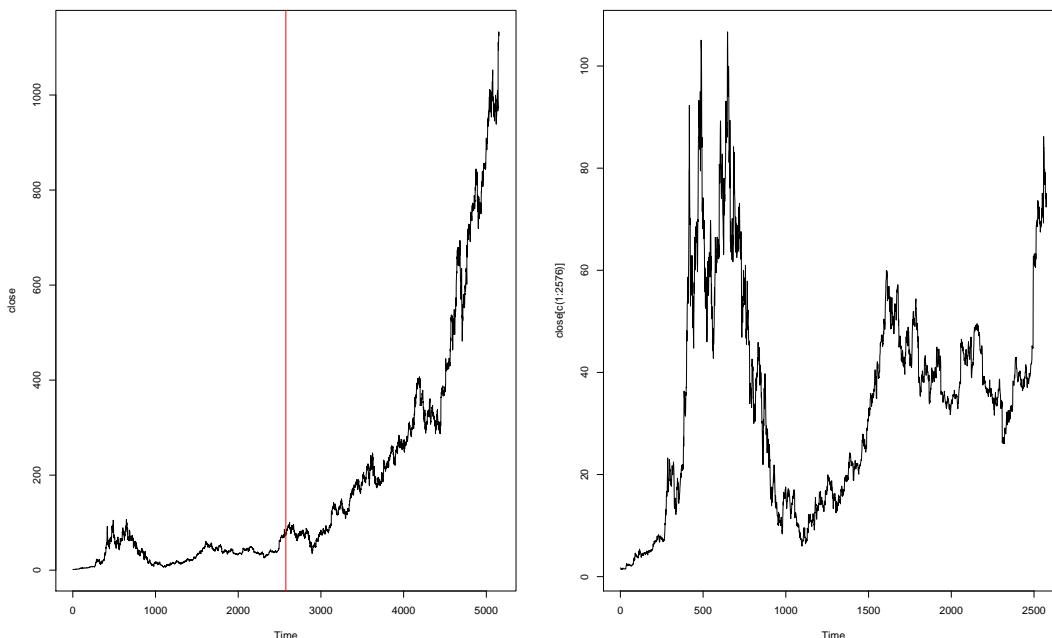
Že v 3.2 sem omenil, da so finančni donosi navadno "left-skewed", torej je mediana manjša od povprečja. Prav zato naj bi bili skriti markovski modeli, po [1] bolj primerni za ugotavljanje padajočih cen finančnih instrumentov. To bom preveril na več setih podatkov.

Podatki, ki sem jih za to pridobil pokrivajo vrednosti delnic na newyorški borzi NYSE za vse delnice, ki so tu kotirale od 1970 naprej, pa do leta 2016, za analizo pa bom uporabil le nekaj podjetij. Te podatke bom uporabil predvsem v splošni analizi kakovosti modela 7.1, kjer bom nato primerjal cene, ki jih predvideva skriti markovski model z dejanskimi cenami.

V drugem delu ekstremni dogodki 7.2 bom poizkušal ugotoviti, ali model deluje tudi ob izjemnih dogodkih na trgu. Zato sem podatke iz NYSE uporabil za analizo stanja delnic ob finančni krizi 2008 ter podatke, ki sem jih pridobil za S&P 500 indeks pred začetkom velike depresije. S tem želim ugotoviti, ali lahko model res zazna hitre padce cen.

Po drugi strani pa želim v 7.2 ugotoviti, ali lahko model predvidi izredno hitro rast cene finančnega instrumenta. Zato sem pridobil podatke o ceni Bitcoina, za katerega vemo, da je ob koncu leta 2017 doživel izredno veliko rast.

V zadnjem delu napoved 7.3 pa bom želel glede na to, kaj mi bosta o kakovosti modela povedali točki 7.1 in 7.2 ugotoviti prihodnjo ceno nekaj finančnih instrumentov. Seveda ne moremo vzeti prevelikega razpona podatkov, saj lahko na dolgi rok cena instrumenta zraste izven ranga, na katerem smo naš model trenirali. To lahko lepo vidimo, ko pogledamo na zaključne cene delnice Amazona.



Na levi sliki je z rdečo črto prikazana datum, do katerega so na desni sliki prikazane cene delnic. Očitno je, da vrednosti, ki jih je cena delnice dosegla ob koncu merjenja, ne moremo doseči z našim modelom. Tako se bomo morali osredotočiti na krajše časovno obdobje.

Prileganje bom ugotavljal s pomočjo razlike med predvideno vrednostjo in dejansko

vrednostjo, ki je bila izmerjena. Za to razliko bom ugotovil varianco, ter minimalno in maksimalno absolutno vrednost.

7.1. Splošna analiza kakovosti modela. Za splošno analizo kakovosti modela sem pridobil podatke iz newyorške borze NYSE za ceno 10 delnic. Ta podjetja so Amazon, Google, Tesla, TripAdvisor, Starbucks, Novartis, Microsoft, General Motors, Wells Fargo in Chevron. V mojem portfelju se tako nahajajo podjetja iz različnih industrij, od finančne do avtomobilske in farmacevtske. Za vsako izmed teh delnic bom analiziral ceno ob koncu trgovalnega dne.

Za vsako izmed teh podjetij bom izbral dve obdobji, vsako dolgo 100 dni ter na njiju izvedel trening modela.

7.2. Ekstremni dogodki.

7.3. Napoved.

LITERATURA

- [1] D. Roman, G. Mitra in N. Spagnolo, *Hidden Markov models for financial optimization problems*, IMA Journal of Management Mathematics **21** (2010) 111–129.
- [2] I.L. MacDonald in W. Zucchini, *Hidden Markov and Other Models for Discrete-valued Time Series*, Chapman & Hall/CRC Monographs on Statistics & Applied Probability **70**, Chapman & Hall, London, 1997.
- [3] R.S. Mamon in R.J. Elliott *Hidden Markov Models in Finance*, International Series in Operations Research & Management Science **104**, Springer, New York, 2007.
- [4] P.J. Brockwell, R.A. Davis *Introduction to Time Series and Forecasting*, 2nd edition, Springer, 2002.
- [5] I. Priimek, *Naslov članka*, okrajšano ime revije **letnik revije** (leto izida) strani od–do.
- [6] C. Velkovrh, *Nekaj navodil avtorjem za pripravo rokopisa*, Obzornik mat. fiz. **21** (1974) 62–64.
- [7] P. Angelini, F. Frati in M. Kaufmann, *Straight-line rectangular drawings of clustered graphs*, Discrete Comput. Geom. **45** (2011) 88–140.
- [8] I. Priimek, *Naslov knjige*, morebitni naslov zbirke **zaporedna številka**, založba, kraj, leto izdaje.
- [9] J. Globevnik in M. Brojan, *Analiza I*, Matematični rokopisi **25**, DMFA – založništvo, Ljubljana, 2010.
- [10] J. Globevnik in M. Brojan, *Analiza I*, Matematični rokopisi **25**, DMFA – založništvo, Ljubljana, 2010; dostopno tudi na <http://www.fmf.uni-lj.si/~globevnik/skripta.pdf>.
- [11] S. Lang, *Fundamentals of differential geometry*, Graduate Texts in Mathematics **191**, Springer-Verlag, New York, 1999.
- [12] I. Priimek, *Naslov članka*, v: naslov zbornika (ur. ime urednika), morebitni naslov zbirke **zaporedna številka**, založba, kraj, leto izdaje, str. od–do.
- [13] S. Cappell in J. Shaneson, *An introduction to embeddings, immersions and singularities in codimension two*, v: Algebraic and geometric topology, Part 2 (ur. R. Milgram), Proc. Sympos. Pure Math. **XXXII**, Amer. Math. Soc., Providence, 1978, str. 129–149.
- [14] I. Priimek, *Naslov dela*, diplomsko/magistrsko delo, ime fakultete, ime univerze, leto.
- [15] J. Kališnik, *Upodobitev orbiterosti*, diplomsko delo, Fakulteta za matematiko in fiziko, Univerza v Ljubljani, 2004.
- [16] I. Priimek, *Naslov spletnega vira*, v: ime morebitne zbirke/zbornika, ki vsebuje vir, verzija številka/datum, [ogled datum], dostopno na spletni.naslov.
- [17] J. Globevnik in M. Brojan, *Analiza 1*, verzija 15. 9. 2010, [ogled 12. 5. 2011], dostopno na <http://www.fmf.uni-lj.si/~globevnik/skripta.pdf>.
- [18] *Matrix (mathematics)*, v: Wikipedia: The Free Encyclopedia, [ogled 12. 5. 2011], dostopno na [http://en.wikipedia.org/wiki/Matrix_\(mathematics\)](http://en.wikipedia.org/wiki/Matrix_(mathematics)).