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

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 navedi nekaj ključnih pojmov, ki nastopajo v delu

Keywords: hidden markov models, time series, angleški prevod ključnih besed

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.

Slovar strokovnih izrazov

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.

Akaikejev informacijski kriterij Recimo, da imamo podatke s k ocenjenimi parametri in L maksimalna vrednost funkcije verjetja. Akaikejev informacijski kriterij izračunamo kot

$$AIC = 2k - 2ln(L)$$

Akaikejev informacijski kriterij primerja več različnih modelov po kakovosti, kjer je najboljši tisti z najmanjšim AIC. **Baum-Welchov algoritem**

2. Markovski modeli

Preden lahko začnemo govoriti o skritih markovskih modelih, moram nekaj povedati o markovskih modelih. Markovski modeli so modeli, kjer velja markovska lastnost.

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

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

Če imamo S, ki je diskretna množica z diskretno sigma algebro in $I = \mathbb{N}I = \mathbb{N}$, je lahko formulirano tudi kot:

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

Markovska lastnost je torej lastnost stohastičnega procesa, da je njegova vrednost v času t odvisna le od njegove vrednosti v času $t - 1$.
Naravno se nam postavi vprašanje, zakaj je markovska lastnost koristna. Izkaže se, da nam omogoča reševanje problemov, ki jih drugače v primernem času ne bi mogli rešiti. Model, ki ga lahko tako rešimo imenujemo Markov model.
Poznamo več različnih vrst Markovih modelov, ki jih lahko razdelimo med 4 podkategorije.

	V celoti opazovan	Le delno opazovan
Avtonomen	markovska veriga	Skriti markovski model
Kontorliran	markovski proces odločanja	Delno opazovalen proces odločanja

2.1. **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.
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 kašno verjetnostjo se premikamo med posamičnimi stanji.

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

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. Kako to naredimo, si bomo pogledali v nadaljevanju.

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.

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 zakona velikih števil zaradi spora z drugim ruskim matematikom, 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. To je pomagalo razvoju algoritmov pomembnih za skrite markovske modele, kot so viterbijev, EM-algoritem, ...
Algoritem maksimizacije pričakovanega (Expectation-maximization), ali krajše 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.

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. Rezulate imenujemo signali ali opazovanja.
(2) V vsakem izmed časov t je sistem lahko v enem izmed N stanj. Ta stanja so $S_1, S_2, \ldots, S_N$. 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 $Q = (Q_t)_{t=1,2,\ldots}$, 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 odvistnosti 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^t_{ij}.
(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,
- 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 imenoval 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 $t \in (1, \ldots, T)$ 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 z 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 N in izbere najboljšega.
Za pravilno določitev števila mešanic M pa navadno izberemo razvrščanjem v skupine (k-mean clustering).
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 enak način izračunamo tudi $\beta_t(i)$, ki ni nujen za izračun $P(O|\lambda)$, vendar ga nujno rabimo 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 porazdelitivjo, 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 $\lambda = \lambda_t, t \in 0, \ldots$, da bo veljalo

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

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 verjtnost, 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:

$$\overline{\Pi_i} = \gamma_1(i)$$

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1} \xi_t}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\overline{c_{jk}} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{m=1}^{M} \gamma_t(j,m)}$$

$$\overline{\mu_{jk}} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) O_t}{\sum_{t=1}^{T} \gamma_t(j,k)}$$

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 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.

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 najglobje posvetil v nasledenjem delu, zato sedaj raje poglejmo ostale načine uporabe.

4.1. **Procesiranje 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.** 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. **Prepozavanje lastnoročne pisave.**

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

4.6. **Prepoznavanje akcij.**

5. Uporaba v finančnih časovnih vrstah

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

Opazovanja x_t so 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 pravilnih 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.

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.

V našem primeru bomo za njihovo analizo uporabili skrite markovske modele.
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 volatilnost 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.

6. Praktični primer

6.1. **Analiza kakovosti modela.**

6.2. **Napoved.**

Literatura