TARTU ÜLIKOOL

LOODUS- JA TÄPPISTEADUSTE VALDKOND

Matemaatika ja statistika instituut

Hanna Läänemets

Autoregressiivsed peidetud Markovi mudelid

Matemaatilise statistika eriala

Magistritöö (30 EAP)

Juhendaja: Märt Möls, PhD

TARTU 2017

Autoregressiivsed peidetud Markovi mudelid

Käesoleva magistritöö eesmärk on tutvustada tavalise peidetud Markovi mudeli ning autoregressiivse peidetud Markovi mudeli hindamise meetodeid ning võrrelda nende sobivust juhul, kui vaatluste sõltuvust ei tingi mitte ainult peidetud Markovi ahel. Töö esimeses kahes peatükis antakse ülevaade peidetud Markovi mudelist ning autoregressiivsest peidetud Markovi mudelist ning nende hindamise meetoditest. Töö kolmandas peatükis võrreldakse simulatsioonide abil, kuidas käituvad need meetodid juhul, kui tegu on andmetega, mis tegelikult vastavad mingile autoregressiivsele peidetud Markovi mudelile. Lisaks tehakse läbi näide meetodite töötamise kohta teise põlvkonna sekveneerimisandmetel.

CERCS teaduseriala: P160 Statistika, operatsioonianalüüs, programmeerimine, finants- ja kindlustusmatemaatika

Märksõnad: juhuslikud protsessid, Markovi ahelad, autoregressioonimudelid

Autoregressive Hidden Markov Models

The aim of this master's thesis is to introduce the estimation methods of hidden Markov models and autoregressive hidden Markov models and compare how well these methods work in case the dependence of observations is not only caused by hidden states. In the first two paragraphs of the thesis, an overview of hidden Markov model, autoregressive hidden Markov model and their estimation methods is given. In the third paragraph, a comparison of the two methods is done, simulating observations which actually correspond to some autoregressive hidden Markov Model. In addition, an example of how the methods work is conducted, using second-generation sequencing data.

CERCS research specialisation: P160 Statistics, operations research, programming, actuarial mathematics

Keywords: stochastic processes, Markov chains, autoregressive models

Sisukord

Sissejuhatus

Markovi ahelad on statistikas tuntud juhuslikud protsessid, mis kirjeldavad juhuslike suuruste ehk seisundite muutumist ajas. Markovi ahelad rahuldavad nn Markovi omadust: fikseeritud oleviku korral tulevik ei sõltu minevikust. Markovi ahelate teooriat alustas Andrei Markov 20. sajandi alguses. Kõige klassikalisem Markovi ahel on selline, kus vaatlused on juhusliku protsessi seisundid, mis on antud igal diskreetsel ajahetkel t.

Vahel võivad aga ajas muutuvad juhuslikud suurused olla sellised, mille vaatlused ise pole Markovi ahela seisundid. Vaatlused on hoopis sellised, mis sõltuvad mingitest seisunditest, mis meile pole nähtavad. Seejuures saame eeldada, et need seisundid rahuldavad Markovi omadust ning vaatlus ei sõltu teistest vaatlustest (kui on teada, millises seisundis Markovi ahel vastava vaatluse tekkimise ajal oli). Selliste varjatud seisunditega protsessi nimetatakse peidetud Markovi mudeliks. Peidetud Markovi mudeli teooriat tutvustasid L. E. Baum ja tema kolleegid 1960. aastatel ning see on tänapäeval laialt levinud nt kõnetuvastuses ja bioinformaatikas.

Nagu öeldud, on peidetud Markovi mudelite oluliseks eelduseks see, et vaatlus sõltub vaid vastava ajahetke peidetud seisundist, kuid mitte teistest vaatlustest. Praktikas võib aga esineda olukordi, kus selline eeldus pole täidetud, vaid vaatlus sõltub ka näiteks talle eelnevast vaatlusest. See võib tähendada, et tavapärased peidetud seisundite leidmise meetodid ei anna häid tulemusi. Sellisel juhul meenutavad vaatlused justkui autoregressiivset aegrida, kus mudeli parameetrid sõltuvad vastava ajahetke peidetud seisundist. Sellist mudelit nimetatakse autoregressiivseks peidetud Markovi mudeliks ning sellele pani aluse J. D. Hamilton 1980. aastate lõpus.

Käesoleva magistritöö eesmärk on tutvuda tavalise peidetud Markovi mudeli ning autoregressiivse peidetud Markovi mudeli hindamise meetoditega ning võrrelda nende töökindlust juhul, kui vaatluste sõltuvust ei tingi mitte ainult varjatud Markovi ahel. Töö esimeses peatükis antakse ülevaade peidetud Markovi mudelist ning selle hindamiseks vajaminevatest etappidest. Teises peatükis tutvustatakse autoregressiivset peidetud Markovi mudelit ning selle hindamise meetodit. Töö kolmandas peatükis võrreldakse simulatsioonide abil, kuidas käituvad vastavad meetodid juhul, kui tegu on andmetega, mis tegelikult vastavad mingile autoregressiivsele peidetud Markovi mudelile. Samuti uuritakse kahe meetodi sobivust teise põlvkonna sekveneerimisandmetete analüüsimiseks. Lihtsuse huvides käsitletakse

autoregressiivseid peidetud Markovi mudeleid vaid juhul, kui autoregressiivse protsessi kordajaid on ainult 1.

Magistritöö on vormistatud tekstitöötlusprogrammiga MS Word. Praktiliste simulatsioonide ja näidete läbiviimiseks on kasutatud statistikatarkvara R. Simuleerimiseks kasutatud programmikoodid on esitatud töö lisas.

Autor tänab juhendajat Märt Mölsi magistritööd puudutavate asjalike nõuannete ja märkuste eest.

1 Peidetud Markovi mudelid

1.1 Sissejuhatus Markovi mudelitesse: diskreetse ajaga Markovi ahelad

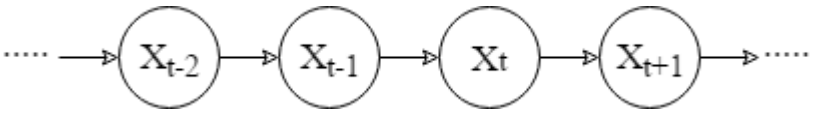
Järgnev alapeatükk tugineb Meelis Kääriku 2014. aasta loengukonspektile õppeaines „Juhuslikud protsessid“.

Juhusliku protsessi all mõistetakse juhuslike suuruste peret $\{X(t): t \in T\}$, mille iga liige $\{X(t)\}$ on juhuslik suurus. Eeldame, et kõik juhuslikud suurused $\{X(t)\}$ on määratud ühel ja samal tõenäosusruumil $(\Omega, \mathcal{F}, \mathrm{P})$. Parameeter t on reaalarvuline muutuja, mida tavaliselt tõlgendatakse ajana, ning hulka T nimetatakse juhusliku protsessi indekshulgaks. Kui T on loenduv hulk, öeldakse, et juhuslik protsess on diskreetse ajaga protsess. Edaspidi tähistame juhuslikku suurust ajahetkel t kui X_t.

Juhuslike suuruste jada $\{X_t\}$, kus t võib omada lõpliku või loendava arvu väärtusi, nimetatakse Markovi ahelaks, kui kehtib

$$\mathrm{P}(X_t = j \mid X_0 = k_0, X_1 = k_1, \dots, X_{t-1} = k_{t-1}) = \mathrm{P}(X_t = j \mid X_{t-1} = k_{t-1}),$$

st fikseeritud oleviku korral tulevik ei sõltu minevikust (Markovi omadus). Seisundi X_t võimalikke väärtusi $1, 2, \dots, N$ nimetatakse Markovi ahela seisunditeks ehk olekuteks. Markovi ahela struktuur on skemaatiliselt esitatud joonisel 1.1.



Joonis 1.1. Markovi ahela struktuur

Markovi ahelatega tegeledes on olulisel kohal üleminekutõenäosused seisundist i seisundisse j:

$$\mathrm{P}(X_t = j \mid X_{t-1} = i).$$

Kui need tõenäosused ei sõltu ajahetkest t, siis nimetatakse Markovi ahelat homogeenseks. Käesolevas magistritöös on edaspidi eeldatud, et tegeleme vaid homogeensete Markovi ahelatega. Sellisel juhul saame üleminekutõenäosused tähistada kui

$$a_{ij} = \mathrm{P}(X_t = j \mid X_{t-1} = i),$$

kusjuures üleminekumaatriks A on defineeritud kui maatriks, mille element (i, j) on a_{ij}.

Üleminekutõenäosuste jaoks kehtivad omadused

$$a_{ij} \geq 0 \quad \forall i, j,$$

$$\sum_j a_{ij} = 1.$$

Tähistame Markovi ahela algseisu tõenäosusjaotuse kui $\pi_i = \mathrm{P}(\mathrm{X}_1 = i)\ \forall i$.

Näide 1.1. Üks levinumaid näiteid Markovi ahelate selgitamiseks on lihtsustatud ilma mudel. Oletame, et homne ilm sõltub ainult tänasest ilmast, kuid mitte varasematest päevadest. Samuti oletame, et igal päeval on kolm erinevat võimalikku ilma – sajune (seisund 1), pilvine (seisund 2) ja päikseline (seisund 3). Olgu üleminekumaatriks A näiteks

$$A = \begin{bmatrix} 0.5 & 0.2 & 0.3 \\ 0.3 & 0.6 & 0.1 \\ 0.2 & 0.1 & 0.7 \end{bmatrix},$$

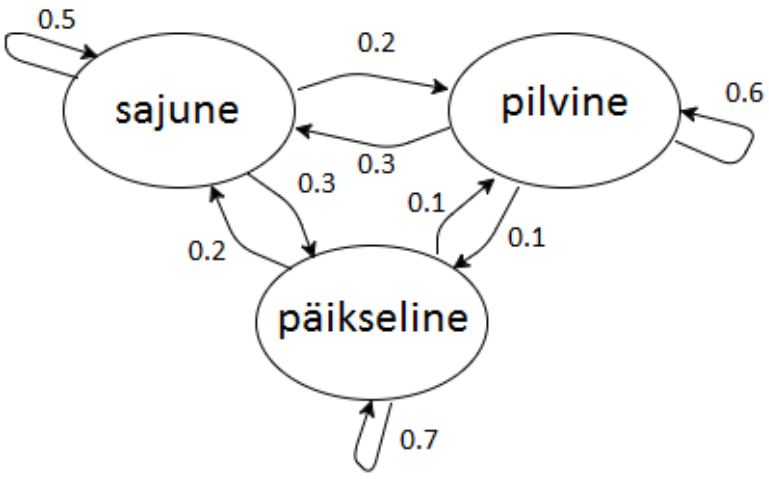
siis juhul, kui täna on päikseline ilm, on homme tõenäosusega 0.2 sajune ilm. Samuti saame välja arvutada näiteks tõenäosuse, et järgmise 3 päeva ilm on päikseline-pilvine-sajune:

$$\mathrm{P}(X_1 = 3, X_2 = 3, X_3 = 2, X_4 = 1) =$$

$$= \mathrm{P}(X_1 = 3) \cdot \mathrm{P}(X_2 = 3 | X_1 = 3) \cdot \mathrm{P}(X_3 = 2 | X_2 = 3) \cdot \mathrm{P}(X_4 = 1 | X_3 = 2) =$$

$$\pi_3 \cdot a_{33} \cdot a_{32} \cdot a_{21} = 1 \cdot 0.7 \cdot 0.1 \cdot 0.3 = 0.021.$$

Antud Markovi ahelat iseloomustab alljärgnev skeem:



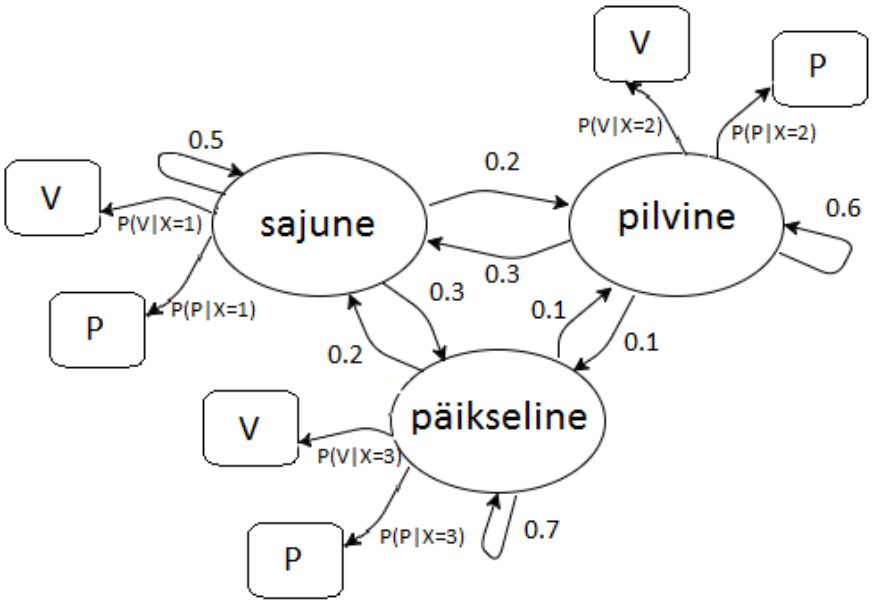
Joonis 1.2. Markovi ahela üleminekutõenäosused näites 1.1

1.2 Peidetud Markovi mudeli definitsioon

Seni vaatasime Markovi mudeleid, kus iga seisund vastas mingile vaadeldavale sündmusele. Nüüd laiendame kontseptsiooni aga Markovi mudeliteni, kus vaadeldavad suurused on seisundist sõltuvad juhuslikud suurused, kuid seisundid ise pole vaadeldavad. Saadavat mudelit nimetatakse peidetud Markovi mudeliks (*Hidden Markov Model,* edaspidi ka HMM). Sellisel juhul on iga vaatlus sõltuv vastava ajahetke peidetud seisundist. Seisundite jada on eelduste kohaselt Markovi ahel, mille praegune seisund sõltub ainult eelmisest. (Rabiner L. R., 1989, lk 259)

Näide 1.2. Oletame endiselt, et homne ilm sõltub ainult tänasest ilmast, kuid mitte varasematest päevadest. Samuti oletame, et igal päeval on kolm erinevat võimalikku ilma – sajune (seisund 1), pilvine (seisund 2) ja päikseline (seisund 3). Muutunud on aga see, et oleme akendeta toas ning ilma seisund on meie jaoks varjatud. Näeme ainult seda, kas tuppa tuleval inimesel on kaasas vihmavari või mitte. Seega peidetud seisundid on sajune, pilvine, päikseline ning seisundist sõltuvad vaatlused on vihmavari (V) ja vihmavarju puudumine (P).

Pärast t päeva möödumist on meil vaatluste jada $Y = \{Y_1, \dots, Y_t\}$, näiteks $\{V, V, P, V, P\}$, ning iga vaatlus sõltub peidetud seisundite ahelast $X = \{X_1, \dots, X_t\}$, kus $X_i \epsilon \{1,2,3\}$. Meie soovime leida kõige tõenäolisemat seisundite jada X vaatluste jada abil. Sellist peidetud Markovi ahelat iseloomustab järgnev skeem:



Joonis 1.3. Peidetud Markovi mudeli skemaatiline esitus näite 1.2 puhul

Peidetud Markovi mudelite põhialuseid ja teooriat tutvustasid L. E. Baum ja tema kolleegid algselt juba 1960. aastatel (vt nt Baum & Petrie, 1966). Kuna tol ajal ei suudetud seda meetodit arvutusliku keerukuse tõttu praktikas kasutada, hakkasid peidetud Markovi mudelid rohkem kasutust leidma alles 1980ndatel. Ühena esimestest kirjutasid Rabiner, Huang ja nende kolleegid artikleid selle kohta, kuidas peidetud Markovi mudeleid rakendada kõnetuvastuses (vt nt Rabiner, 1989). Tänapäeval leiab HMM kasutust paljudes valdkondades: lisaks kõnetuvastusele näiteks ka käekirja tuvastamises, majanduses, bioinformaatikas, geneetikas ning isegi muusikateoorias.

Peidetud Markovi mudelit iseloomustavad järgmised komponendid (Rabiner L. R., 1989, lk 260-261):

1) N, seisundite arv mudelis. Tähistame seisundite klassi kui $X = \{1, \dots, N\}$, seisund ajahetkel t olgu X_t.
2) M, vaadeldavate juhuslike suuruste Y_t võimalike väärtuste arv juhul, kui vaatlused on diskreetsed juhuslikud suurused. Tähistame sel juhul võimalikud vaatlused kui $Y = \{y_1, \dots, y_M\}$.
3) T, vaatluste jada pikkus. Seisundite jada ja vaatluste jada saab kirja panna kui $\{X_1, \dots, X_T\}$ ja $\{Y_1, \dots, Y_T\}$.
4) Seisundite üleminekutõenäosuste maatriks $\mathrm{A} = \{a_{ij}\}$, kus $a_{ij} = \mathrm{P}(X_t = j \mid X_{t-1} = i)$.
5) Algne seisundite jaotus $\pi = \{\pi_i\}$, kus $\pi_i = \mathrm{P}(X_1 = i), \ \ 1 \le i \le N$.
6) Vaatluste jaotus seisundis i (juhul, kui vaatlused on diskreetsed juhuslikud suurused), $B = \{b_i(k)\}$, kus $b_i(k) = \mathrm{P}(Y_t = y_k | X_t = i)$.

   Kui vaatlused on pidevad juhuslikud suurused, siis tähistame $b_i(y) = f_{Y_t}(y | X_t = i)$ kui tingliku tõenäosusfunktsiooni juhul, kui $X_t = i$. Kõige tavapärasem on pidevate vaatluste korral eeldada normaaljaotuste segu:

$$b_i(y) = \sum_{m=1}^{M} C_{jm} \mathcal{N}(\mu_{jm}, \Sigma_{jm}),$$

   kus C_{jm} on m. jaotuse kaal seisundis j. Kõik C_{jm} on mittenegatiivsed ja $\sum_{m=1}^{M} C_{jm} = 1, \quad 1 \le j \le N$. Parameetrid μ_{jm} ja Σ_{jm} on m. jaotuse keskväärtus ja dispersioon seisundis j.

Mingi konkreetse peidetud Markovi mudeli määravad kolm mudeli suurust ja vaatluste arvu iseloomustavat parameetrit (N, M, T) ja kolm hinnatavat parameetrit (A, B, π). Edasise lihtsuse

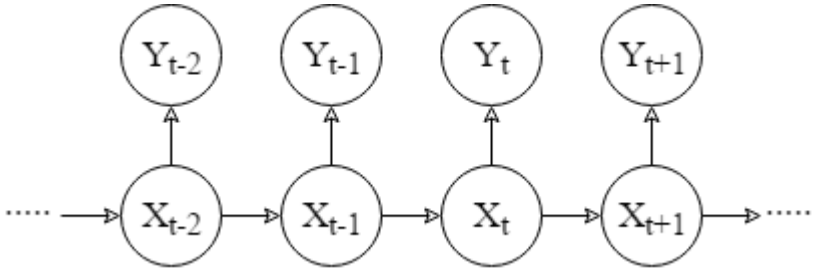
huvides tähistame viimase parameetrite komplekti kui λ. Kui Markovi ahelal $\{X_t\}$ on N erinevat võimalikku seisundit, siis nimetatakse mudelit N-seisundiliseks peidetud Markovi mudeliks.

Peidetud Markovi mudeli korral avalduvad vaatluste ja seisundite tõenäosused järgmiselt (Zucchini & MacDonald, 2009, lk 30):

$$\mathrm{P}(X_t|X_1, \dots, X_{t-1}) = \mathrm{P}(X_t|X_{t-1}), \qquad t = 2, 3, \dots$$

$$\mathrm{P}(Y_t|Y_1, \dots, Y_{t-1}, X_1, \dots, X_t) = \mathrm{P}(Y_t|X_t), \qquad t \in \mathbb{N}.$$

Joonisel 1.4 on kujutatud skemaatiline esitus HMM-i struktuurist. Mudel koosneb kahest osast: esiteks peidetud seisunditest, mis rahuldavad Markovi omadust, ja teiseks seisundist sõltuvate vaatluste protsessist. Kui seisund X_t on teada, siis vaatluse Y_t jaotus sõltub (tavalise peidetud Markovi mudeli korral) ainult seisundist X_t ja mitte eelnevatest seisunditest ega vaatlustest. 2. peatükis vaatame autoregressiivseid peidetud Markovi mudeleid, kus Y_t jaotus võib sõltuda ka eelnevatest vaatlustest.



Joonis 1.4. Peidetud Markovi mudeli struktuur

Rabineri sõnul tuleb selleks, et peidetud Markovi mudelit rakendada saaks, lahendada kõigepealt kolm põhilist probleemi. Need võib sõnastada järgnevalt (Rabiner L. R., 1989, lk 261):

1) Kui on antud vaatluste jada $Y = \{Y_1, \dots, Y_T\}$ ja mudel $\lambda = (A, B, \pi)$, kuidas arvutada efektiivselt $\mathrm{P}(Y|\lambda)$?
2) Olgu antud vaatluste jada $Y = \{Y_1, \dots, Y_T\}$ ja mudel λ. Kuidas valida sobivat seisundite jada $X = \{X_1, \dots, X_T\}$, mis oleks mingis mõttes optimaalne (st kirjeldaks vaatlusi kõige paremini)?
3) Kuidas kohandada mudeli parameetreid $\lambda = (A, B, \pi)$, et maksimiseerida $\mathrm{P}(Y|\lambda)$?

Esimene probleem on hindamise probleem: kuidas hinnata antud vaatluste jada ja mudeli korral tõenäosust, et vaatlus vastab just sellisele mudelile? Seda tõenäosust võib vaadata ka kui mudeli headusele skoori andmist. Sellisel juhul valime mitme mudeli seast sellise, mis sobib kõige paremini meie vaatlustele.

Teise probleemi lahendamiseks tuleb leida mudeli peidetud osa, st leida „õige" seisundite jada. Kui meil pole just enda genereeritud andmed, siis ei saagi „õiget" seisundite jada teada olla. Praktikas kasutatakse probleemi lahendamiseks seetõttu optimaalsuse kriteeriumit.

Kolmanda probleemi lahendamiseks tuleb hinnata mudeli parameetrid selliselt, et saaksime vaatluste jada kõige paremini kirjeldada. See tähendab, et tuleb leida parameetrite väärtused, mis maksimeeriksid tõepära $\mathrm{P}(Y|\lambda)$. Olemasolevat vaatluste jada kasutame justkui treeningandmetena.

Järgnevalt vaatamegi, milliste meetoditega on võimalik Rabineri sõnastatud probleeme lahendada.

1.3 Esimese ja kolmanda probleemi lahendus

Alapeatükkide 1.3 ja 1.3.1 kirjeldamisel on autor tuginenud Rabineri artiklile (Rabiner L. R., 1989, lk 262, 264).

Uurime, kuidas leida optimaalsete parameetritega mudelit, mis sobiks vaatluste jadaga kõige paremini. Esiteks leiame viisi, kuidas võimalikult efektiivselt arvutada $\mathrm{P}(Y|\lambda)$.

Tahame arvutada vaatluste jada $Y = \{Y_1, \dots, Y_T\}$ tinglikku tõenäosust mudeli $\lambda = (A, B, \pi)$ korral, st $\mathrm{P}(Y|\lambda)$. Kõige otsesem viis seda teha on nummerdada kõik võimalikud seisundite jadad pikkusega T. Olgu üks selline fikseeritud seisundite jada $X = \{X_1, \dots, X_T\}$, kus X_1 on esialgne seisund. Vaatluste jada Y tõenäosus sellise seisundite jada X korral on

$$\mathrm{P}(Y|X, \lambda) = \prod_{t=1}^{T} \mathrm{P}(Y_t|X_t, \lambda),$$

sest oleme eeldanud vaatluste sõltumatust. Seega saame

$$\mathrm{P}(Y|X, \lambda) = b_{X_1}(Y_1) \cdot b_{X_2}(Y_2) \cdot \dots \cdot b_{X_T}(Y_T),$$

kus $b_{X_t}(Y_t)$ on tinglik tõenäosus näha vaatlust Y_t seisundi X_t korral.

Fikseeritud seisundite jada X tõenäosuse saame kirja panna kui

$$\mathrm{P}(X|\lambda) = \pi_{X_1} a_{X_1X_2} a_{X_2X_3} \dots a_{X_{T-1}X_T},$$

kus π_{X_1} on tõenäosus olla ajahetkel 1 algseisundis X_1 ning $a_{X_{t-1}X_t}$ on tõenäosus liikuda seisundist X_{t-1} seisundisse X_t.

Tõenäosus, et X ja Y esinevad üheaegselt, on lihtsalt nende kahe tõenäosuse korrutis:

$$\mathrm{P}(Y, X|\lambda) = \mathrm{P}(Y|X, \lambda)\ \mathrm{P}(X|\lambda).$$

Kui tahame arvutada Y tõenäosust konkreetse mudeli korral, siis peame summeerima eelneva tõenäosuse üle kõigi võimalike seisundite jadade X, seega

$$\mathrm{P}(Y|\lambda) = \sum_{kõik\ X} \mathrm{P}(Y|X, \lambda)\mathrm{P}(X|\lambda) = \sum_{kõik\ X} \pi_{X_1} b_{X_1}(Y_1) a_{X_1X_2} b_{X_2}(Y_2) \dots a_{X_{T-1}X_T} b_{X_T}(Y_T).$$

Sellise otsese arvutusmeetodi korral oleks meil vaja teha $2T \cdot N^T$ tehet, sest igal ajahetkel $t = 1, 2, \dots, T$ on N võimalikku seisundit, järelikult on kokku N^T võimalikku seisundite jada X. Iga sellise jada jaoks on vaja teha $2T$ korrutustehet, et saada vajalikku liidetavat kogusummasse. Selline arvutus läheks liiga mahukaks isegi väikeste N ja T korral, nt $N = 5$ ja $T = 100$ korral peaksime tegema $2 \cdot 100 \cdot 5^{100} \approx 10^{72}$ tehet. Parem on kasutada meetodit, mida nimetatakse *forward-backward* meetodiks.

1.3.1 Forward-backward meetod

Defineerime *forward*-muutuja $\alpha_t(i)$ kui tõenäosuse, et mudeli λ korral tekib osaline vaatluste jada ajahetkeni t (st $Y_1, Y_2, \dots, Y_t$) ja ajahetkel t on seisund i, st

$$\alpha_t(i) = \mathrm{P}(Y_1, Y_2, \dots, Y_t, X_t = i\ |\lambda).$$

Saame $\alpha_t(i)$ leida induktiivselt:

1) Algväärtustamine:

$$\alpha_1(i) = \mathrm{P}(Y_1\ , X_1 = i\ |\lambda) = \mathrm{P}(Y_1|X_1 = i, \lambda) \cdot \mathrm{P}(X_1 = i|\lambda) = \pi_i b_i(Y_1), \qquad 1 \le i \le N$$

2) Induktsioon:

$$\alpha_{t+1}(i) = \mathrm{P}(Y_1, Y_2, \dots, Y_{t+1}, X_{t+1} = i \,|\lambda) = \mathrm{P}(Y_{t+1}|Y_1, \dots, Y_t, X_{t+1} = i) \cdot$$
$$\cdot \mathrm{P}(Y_1, \dots, Y_t, X_{t+1} = i) = \mathrm{P}(Y_{t+1}|X_{t+1} = i) \cdot \mathrm{P}(X_{t+1} = i|Y_1, \dots, Y_t) \cdot \mathrm{P}(Y_1, \dots, Y_t) =$$
$$\mathrm{P}(Y_{t+1}|X_{t+1} = i) \cdot \sum_{j=1}^{N} \mathrm{P}(X_{t+1} = i|Y_1, \dots, Y_t, X_t = j) \cdot \mathrm{P}(Y_1, \dots, Y_t|X_t = j) \cdot \mathrm{P}(X_t = j) =$$
$$= \mathrm{P}(Y_{t+1}|X_{t+1} = i) \cdot \sum_{j=1}^{N} \mathrm{P}(X_{t+1} = i|\, X_t = j) \cdot \mathrm{P}(Y_1, \dots, Y_t|X_t = j) \cdot \mathrm{P}(X_t = j) =$$
$$= b_i(Y_{t+1}) \sum_{j=1}^{N} a_{ji}\alpha_t(j)\ , \qquad \begin{matrix} 1 \le t \le T-1, \\ 1 \le i \le N. \end{matrix} \tag{1.1}$$

Idee $\alpha_t(j)$ induktiivse arvutamise taga on järgmine: kuna $\alpha_t(j)$ on tõenäosus, et korraga esinevad vaatluste jada $Y_1, \dots, Y_t$ ja ajahetkel t esineb seisund j, siis korrutis $\alpha_t(j)a_{ji}$ näitab tõenäosust, et korraga esinevad vaatluste jada $Y_1, \dots, Y_t$ ja ajahetkel $t+1$ esinevasse seisundisse i jõutakse ajahetkel t esinevast seisundist j. Summeerides selliseid korrutisi üle kõigi võimalike N seisundi ajahetkel t, saame tõenäosuse, et ajahetkel $t+1$ oleme seisundis i. Seejärel saame leida $\alpha_{t+1}(i)$ väärtuse, korrutades saadud summa tõenäosusega, et seisundi i korral esineb vaatlus Y_{t+1}, st $b_i(Y_{t+1})$.

Kui oleme leidnud iga ajahetke t ja seisundi i jaoks $\alpha_t(i)$ väärtuse, saame nende abil leida vaatluste jada $Y = \{Y_1, \dots, Y_T\}$ tõenäosuse mudeli λ korral:

$$\mathrm{P}(Y|\lambda) = \sum_{i=1}^{N} \alpha_T(i).$$

Kuna definitsiooni kohaselt ajahetke T *forward*-muutujad on $\alpha_T(i) = \mathrm{P}(Y_1, Y_2, \dots, Y_T, X_T = i|\lambda)$, siis tõenäosus $\mathrm{P}(Y|\lambda)$ on lihtsalt kõikide $\alpha_T(i)$ summa (kus $1 \le i \le N$).

Vaadates nüüd $\mathrm{P}(Y|\lambda)$ arvutamiseks vaja läinud tehete arvu, näeme, et peame tegema kõigest N^2T arvutust, mitte $2TN^T$ nagu otsese definitsiooni puhul. Seega $N = 5$, $T = 100$ korral läheks 10^{72} arvutuse asemel vaja kõigest u 3000 tehet.

Samamoodi nagu tegutsesime $\alpha_t(i)$ puhul, saame konstrueerida ka *backward*-muutuja $\beta_t(i)$, mis on tõenäosus, et ajahetkel t antud seisundi i ja mudeli λ korral esineb osaline vaatluste jada ajahetkest $t+1$ kuni lõpuni:

$$\beta_t(i) = \mathrm{P}(Y_{t+1}, Y_{t+2}, \dots, Y_T | X_t = i, \lambda).$$

Ka $\beta_t(i)$ saame arvutada induktiivselt, kuid alustame ajahetkest T ning liigume n-ö ajas tagasi:

1) Algväärtused:

$$\beta_T(i) = 1, \qquad 1 \le i \le N$$

2) Induktsioon:

$$\beta_t(i) = \mathrm{P}(Y_{t+1}, Y_{t+2}, \dots, Y_T | X_t = i, \lambda) = \sum_{j=1}^{N} \mathrm{P}(Y_{t+1}, Y_{t+2}, \dots, Y_T | X_t = i, X_{t+1} = j) \cdot$$

$$\cdot \mathrm{P}(X_{t+1} = j | X_t = i) =$$

$$= \sum_{j=1}^{N} \mathrm{P}(Y_{t+1} | Y_{t+2}, \dots, Y_T, X_t = i, X_{t+1} = j) \cdot \mathrm{P}(Y_{t+2}, \dots, Y_T | X_{t+1} = j) \cdot$$

$$\cdot \mathrm{P}(X_{t+1} = j | X_t = i) =$$

$$= \sum_{j=1}^{N} a_{ij} b_j(Y_{t+1}) \beta_{t+1}(j), \qquad \begin{matrix} t = T-1, T-2, \dots, 1, \\ 1 \le i \le N. \end{matrix}$$

Algväärtustamise samm seab ajahetkel T kõikide i korral $\beta_T(i)$ väärtuseks 1. Induktsiooni samm näitab, et selleks, et olla ajahetkel t seisundis i ja leida tõenäosust, et esineb vaatluste jada $Y_{t+1}, Y_{t+2}, \dots, Y_T$, peame arvesse võtma kõiki võimalikke seisundeid ajahetkel $t+1$, üleminekutõenäosust seisundist i seisundisse j; samuti tõenäosust, et seisundi j korral esineb vaatlus Y_{t+1}, ning seejärel arvestama eelnevaid väärtusi $\beta_{t+1}(j)$. Ka *backward*-muutuja väärtuste arvutamiseks läheb vaja N^2T tehet.

1.3.2 Baum-Welchi meetod

Peidetud Markovi mudelite kolmas ja kõige keerulisem probleem on kirjeldada meetodit, millega leida parimad mudeli parameetrid (A, B, π). Selleks ei ole analüütilisi vahendeid. Kui meil on antud treeningandmetena mingi lõpliku pikkusega vaatluste jada, siis pole optimaalset võimalust, kuidas mudeli parameetreid hinnata. Me saame valida $\lambda = (A, B, \pi)$ selliselt, et $\mathrm{P}(Y|\lambda)$ on lokaalselt maksimeeritud, kasutades näiteks alljärgnevat Baum-Welchi meetodit. Baum-Welchi meetod tugineb idee poolest EM algoritmile (*Expectation-maximization algorithm*), mis on iteratiivne meetod suurima tõepära leidmiseks.

Defineerime alustuseks suuruse $\xi_t(i,j)$, mis on tõenäosus, et antud mudeli ja vaatluste jada korral oleme ajahetkel t seisundis i ja ajahetkel $t+1$ seisundis j, st

$$\xi_t(i,j) = \mathrm{P}(X_t = i, X_{t+1} = j | Y, \lambda).$$

Forward-backward-muutujate abil saame $\xi_t(i,j)$ kirjutada ka kujul

$$\xi_t(i,j) = \frac{\mathrm{P}(X_t = i, X_{t+1} = j, Y|\lambda)}{\mathrm{P}(Y|\lambda)} =$$

$$= \frac{\mathrm{P}(Y_1, \dots, Y_t, X_t = i|\lambda) \cdot \mathrm{P}(Y_{t+1}, \dots, Y_T, X_{t+1} = j | X_t = i, \lambda)}{\mathrm{P}(Y|\lambda)} =$$

$$= \frac{\mathrm{P}(Y_1, \dots, Y_t, X_t = i|\lambda) \cdot \mathrm{P}(Y_{t+1}, \dots, Y_T | X_t = i, X_{t+1} = j, \lambda) \cdot \mathrm{P}(X_{t+1} = j | X_t = i)}{\mathrm{P}(Y|\lambda)} =$$

$$= \frac{\mathrm{P}(Y_1, \dots, Y_t, X_t = i|\lambda) \cdot \mathrm{P}(Y_{t+1} | X_{t+1} = j, \lambda) \cdot \mathrm{P}(Y_{t+2}, \dots, Y_T |\, X_{t+1} = j, \lambda) \cdot \mathrm{P}(X_{t+1} = j | X_t = i)}{\mathrm{P}(Y|\lambda)}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(Y_{t+1}) \beta_{t+1}(j)}{\mathrm{P}(Y|\lambda)} = \frac{\alpha_t(i) a_{ij} b_j(Y_{t+1}) \beta_{t+1}(j)}{\sum_{k=1}^{N} \sum_{l=1}^{N} \alpha_t(k) a_{kl} b_l(Y_{t+1}) \beta_{t+1}(l)}. \qquad (1.2)$$

Järgmisena defineerime suuruse $\gamma_t(i)$ kui tõenäosuse, et oleme ajahetkel t seisundis i, kui teame vaatluste jada Y ja mudelit λ:

$$\gamma_t(i) = \mathrm{P}(X_t = i | Y, \lambda).$$

Suuruse $\gamma_t(i)$ saame avaldada *forward-backward*-muutujate kaudu:

$$\gamma_t(i) = \frac{\mathrm{P}(X_t = i, Y|\lambda)}{\mathrm{P}(Y|\lambda)} = \frac{\mathrm{P}(Y_1, \dots, Y_t, X_t = i|\lambda) \cdot \mathrm{P}(Y_{t+1}, \dots, Y_T | X_t = i, \lambda)}{\mathrm{P}(Y|\lambda)} =$$

$$= \frac{\alpha_t(i)\beta_t(i)}{\mathrm{P}(Y|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}$$

ning $\gamma_t(i)$ ja $\xi_t(i,j)$ omavaheline suhe avaldub kui

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j).$$

Kui summeerime $\gamma_t(i)$ üle indeksi t, saame suuruse, mida võib interpreteerida kui oodatud arv kordi, mil peidetud Markovi ahel on seisundis i, või kui oodatav üleminekute arv seisundist i. Summeerides suurust $\xi_t(i,j)$ üle indeksi t, võime seda interpreteerida kui oodatavat üleminekute arvu seisundist i seisundisse j.

Seega nende kahe interpretatsiooni abil võime anda peidetud Markovi mudeli parameetrite ümberhindamiseks järgmise algoritmi:

1) $\overline{\pi_i} = [\text{oodatav tõenäosus olla ajahetkel } t = 1 \text{ seisundis } i] = \gamma_1(i)$ (1.3)

2) $\overline{a_{ij}} = \frac{\text{oodatav üleminekute arv seisundist } i \text{ seisundisse } j}{\text{oodatav üleminekute arv seisundist } i} =$

$$= \sum_{t=1}^{T-1} \xi_t(i,j) \Big/ \sum_{t=1}^{T-1} \gamma_t(i) \tag{1.4}$$

3) $\overline{b_j(k)} = \frac{\text{oodatav arv kordi, kui esineb seisund } j \text{ ja vaatlus } y_k}{\text{oodatav seisundi } j \text{ esinemise arv}} =$

$$= \sum_{\substack{0 \le t \le T \\ Y_t = y_k}} \gamma_t(j) \Big/ \sum_{t=1}^{T} \gamma_t(j) \tag{1.5}$$

Antud algoritmi järgi saab $b_j(k)$ hinnangud leida juhul, kui vaatlused Y on diskreetsed juhuslikud suurused.

Kui defineerime ühe mudeli $\lambda = (A, B, \pi)$ ja arvutame valemite (1.3)-(1.5) järgi vastavad suurused ning seejärel defineerime ümberhinnatud mudeli kui $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$, siis on tõestatud (Baum & Sell, 1968), et kas

a) esialgne mudel λ defineerib tõepärafunktsiooni kriitilise punkti, sellisel juhul $\bar{\lambda} = \lambda$, või
b) mudeli $\bar{\lambda}$ tõepära on suurem kui mudelil λ ehk $\mathrm{P}(Y|\bar{\lambda}) > \mathrm{P}(Y|\lambda)$, st uus mudel $\bar{\lambda}$ sobib vaatluste jadaga paremini kui eelmine.

Kui kasutame eelnevalt välja toodud algoritmi iteratiivselt ja hindame parameetreid järjest uuesti, saame parandada tulemust kuni mingi punktini. Lõplik tulemus on peidetud Markovi mudeli suurima tõepära hinnang. Tuleb välja tuua, et *forward-backward* algoritm viib ainult lokaalse maksimumini ja paljude juhtumite korral on optimiseeritaval avaldisel palju lokaalseid maksimume. Lisaks võib algoritmi kasutamisel tekkida arvutuslik probleem, mida on võimalik lahendada skaleerimisega.

1.3.2 Skaleerimine

Järgnev alapeatükk põhineb Rabineri artiklil (Rabiner L. R., 1989, lk 272) ning Rahimi parandustel (Rahimi, 2000).

Vaatame uuesti valemit (1.1):

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j)a_{ji}\right] b_i(Y_{t+1}), \qquad \begin{matrix} 1 \le t \le T-1, \\ 1 \le i \le N. \end{matrix}$$

Kui vaadata, kuidas $\alpha_t(i)$ induktsiooniga eelmiste väärtuste abil arvutatakse:

$$\alpha_1(i) = \pi_i \cdot b_i(Y_1),$$

$$\alpha_2(j) = [\sum_{i=1}^{N} \alpha_1(i) \cdot a_{ij}] \cdot b_j(Y_2) = [\sum_{i=1}^{N} \pi_i \cdot b_i(Y_1) \cdot a_{ij}] \cdot b_j(Y_2),$$

$$\alpha_3(k) = \left[\sum_{j=1}^{N} \alpha_2(j)a_{jk}\right] b_k(Y_3) = \left[\sum_{j=1}^{N} [\sum_{i=1}^{N} \pi_i \cdot b_i(Y_1) \cdot a_{ij}] \cdot b_j(Y_2)a_{jk}\right] b_k(Y_3),$$

$$\alpha_4(l) = \left[\sum_{k=1}^{N} \alpha_3(k)a_{kl}\right] b_l(Y_4) =$$

$$= \left[\sum_{k=1}^{N} \left[\sum_{j=1}^{N} [\sum_{i=1}^{N} \pi_i \cdot b_i(Y_1) \cdot a_{ij}] \cdot b_j(Y_2)a_{jk}\right] b_k(Y_3)a_{kl}\right] b_l(Y_4),$$

siis näeme, et igal ajahetkel t on $\alpha_t(i)$ arvutamiseks vaja liita kokku N^{t-1} liidetavat, kusjuures igas liidetavas korrutame omavahel $(t-1)$ erinevat maatriksi $A = [a_{ij}]$ elementi ja t maatriksi $B = \{b_i(k)\}$ elementi. Kuna iga A ja B element on tavaliselt ühest tunduvalt väiksem, siis t suurenedes hakkab $\alpha_t(i)$ väärtus eksponentsiaalselt nullile lähenema. Piisavalt suure t puhul ei suuda arvuti enam $\alpha_t(i)$ väärtust arvutada. Seega ainuke võimalus $\alpha_t(i)$ väärtust leida on kasutada mingisugust skaleerimisprotseduuri.

Põhiline skaleerimisvõte, mida $\alpha_t(i)$ puhul kasutatakse, on $\alpha_t(i)$ korrutamine mingi koefitsiendiga, mis sõltub ainult indeksist t ja ei sõltu i-st. Eesmärk on hoida skaleeritud väärtusi sellises vahemikus, millega arvuti tehteid suudab teha. Sarnane skaleerimine tehakse ka $\beta_t(i)$ koefitsientidele ning seejärel arvutuskäigu lõpus skaleeritud koefitsiendid taanduvad.

Rabineri artiklis toodud valemid $\alpha_t(i)$ skaleerimiseks (Rabiner L. R., 1989, lk 272) on pisut eksitavad ja $\beta_t(i)$ jaoks puuduvad üldse. Seepärast on autor siin alapeatükis edaspidi lähtunud Ali Rahimi korrektuuridest (Rahimi, 2000).

Vaatame üleminekutõenäosuste a_{ij} ümberhindamise valemit (1.4) *forward-* ja *backward-*muutujate kaudu:

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{k=1}^{N} \xi_t(i,k)} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(Y_{t+1}) \beta_{t+1}(j) \ /\mathrm{P}(Y|\lambda)}{\sum_{t=1}^{T} \sum_{k=1}^{N} \alpha_t(i) a_{ik} b_k(Y_{t+1}) \beta_{t+1}(k) / \mathrm{P}(Y|\lambda)} =$$

$$\frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(Y_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T} \sum_{k=1}^{N} \alpha_t(i) a_{ik} b_k(Y_{t+1}) \beta_{t+1}(k)}$$

Teame valemit $\alpha_t(i)$ arvutamiseks (1.1). Nüüd leiame iga t korral $\widehat{\alpha_t}(i)$ selliselt, et

$$\widehat{\alpha_t}(i) = \frac{\alpha_t(i)}{\sum_{j=1}^{N} \alpha_t(j)} = C_t \alpha_t(i). \tag{1.6}$$

Rekursioonisammud näevad välja järgmised:

1) $\overline{\alpha_1}(i) = \alpha_1(i)$
2) $\overline{\alpha_{t+1}}(j) = \sum_{i=1}^{N} \widehat{\alpha_t}(i) a_{ij} b_j(Y_{t+1}), \quad 1 \le t \le T-1$ (1.7)
3) $c_{t+1} = \frac{1}{\sum_{i=1}^{N} \overline{\alpha_{t+1}}(i)}$ (1.8)
4) $\widehat{\overline{\alpha_{t+1}}}(i) = c_{t+1} \overline{\alpha_{t+1}}(i)$

Näitame, miks sellised rekursioonisammud soovitud tulemuse (1.6) annavad.

Kui $t = 1$, siis $\overline{\alpha_1}(i) = \alpha_1(i)$ ja $\widehat{\alpha_1}(i) = {\alpha_1(i)}\Big/{\sum_{j=1}^{N} \alpha_1(j)}$, mis vastab valemile (1.6).

Kui $\widehat{\alpha_t}(i) = C_t \alpha_t(i)$, siis (1.7) põhjal $\overline{\alpha_{t+1}}(j) = [\sum_{i=1}^{N} \widehat{\alpha_t}(i) \cdot a_{ij}] \cdot b_j(Y_{t+1}) =$

$$C_t [\sum_{i=1}^{N} \alpha_t(i) \cdot a_{ij}] \cdot b_j(Y_{t+1}) = C_t \alpha_{t+1}(j).$$

Siis (1.8) saame kujul $c_{t+1} = \frac{1}{\sum_j \overline{\alpha_{t+1}}(j)} = \frac{1}{\sum_j C_t \, \alpha_{t+1}(j)}$ ning

$\widehat{\overline{\alpha_{t+1}}}(i) = c_{t+1} \overline{\alpha_{t+1}}(i) = \frac{C_t \alpha_{t+1}(i)}{C_t \sum_j \alpha_{t+1}(j)} = \frac{\alpha_{t+1}(i)}{\sum_j \alpha_{t+1}(j)}$, mis vastabki valemi kujule (1.6).

Saame kirjutada suuruse C_t välja suuruste c_t abil:

$$C_t = \frac{1}{c_{t+1}\sum\alpha_{t+1}(j)} = \frac{C_{t+1}}{c_{t+1}},$$

$$C_t = C_{t-1}c_t = \prod_{\tau=1}^{t} c_\tau.$$

Defineerime ka suuruse D_t, mille abil skaleerime suurust $\beta_t(i)$:

$$D_t = \prod_{\tau=t}^{T} c_\tau.$$

Sellisel juhul avaldub

$$C_t D_{t+1} = \prod_{\tau=1}^{t} c_\tau \prod_{\tau=t}^{T} c_\tau = \prod_{\tau=1}^{T} c_\tau = C_T.$$

Kui soovime skaleerida suurust $\beta_t(i)$ kui $\widehat{\beta_t}(i) = D_t\beta_t(i)$, siis rekursioonisammud on järgmised:

1) $\overline{\beta_T}(i) = \beta_T(i)$
2) $\overline{\beta_t}(j) = \sum_{i=1}^{N} a_{ij} b_j(Y_{t+1})\, \widehat{\beta_{t+1}}(i), \quad t = T-1, T-2, \dots, 1$
3) $\widehat{\beta_t}(i) = c_t\overline{\beta_t}(i)$

Seega oleme leidnud valemid, millega skaleerida $\alpha_t(i)$ ja $\beta_t(i)$ kujudele $\widehat{\alpha_t}(i) = C_t\alpha_t(i)$ ja $\widehat{\beta_t}(i) = D_t\beta_t(i)$. Järgmisena vaatame, kuidas neid suurusi kasutada, et arvutada $\xi_t(i,j)$ ja $\gamma_t(i)$.

Asendades skaleeritud suurused $\xi_t(i,j)$ definitsiooni valemisse (1.2), saame:

$$\xi_t(i,j) = \frac{1}{\mathrm{P}(Y|\lambda)}\alpha_t(i)a_{ij}b_j(Y_{t+1})\beta_{t+1}(j) = \widehat{\alpha_t}(i)a_{ij}b_j(Y_{t+1})\widehat{\beta_{t+1}}(j)\frac{1}{\mathrm{P}(Y|\lambda)}\frac{1}{C_t}\frac{1}{D_{t+1}}\ . (1.9)$$

Teame, et $C_t D_{t+1} = C_T$, $\frac{\alpha_T(i)}{\sum_{j=1}^{N}\alpha_T(j)} = C_T\alpha_T(i)$ ja $\mathrm{P}(Y|\lambda) = \sum_i \alpha_T(i)$, seega

$$\mathrm{P}(Y|\lambda) = \sum_i \alpha_T(i) = \frac{1}{C_T}$$

ja

$$\mathrm{P}(Y|\lambda)C_T = 1.$$

Valem (1.9) lihtsustub niisiis kujule

$$\xi_t(i,j) = \widehat{\alpha_t}(i)a_{ij}b_j(Y_{t+1})\widehat{\beta_{t+1}}(j). \tag{1.10}$$

$\gamma_t(i)$ saab arvutada $\xi_t(i,j)$ abil, kasutades valemit

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j) = \frac{1}{\mathrm{P}(Y|\lambda)}\alpha_t(i)\beta_t(i) = \widehat{\alpha_t}(i)\widehat{\beta}_t(i)\frac{1}{\mathrm{P}(Y|\lambda)}\frac{1}{C_t}\frac{1}{D_t} = \widehat{\alpha_t}(i)\widehat{\beta}_t(i)\frac{1}{c_t}. \tag{1.11}$$

Valemeid (1.10) ja (1.11) saab Baum-Welchi ja Viterbi algoritmides kasutada $\xi_t(i,j)$ ja $\gamma_t(i)$ arvutamiseks. Viterbi algoritm on kirjeldatud alapeatükis 1.4.1.

1.4 Lahendus teisele probleemile: seisundite jada leidmine

Peatükid 1.4 ja 1.4.1 tuginevad Rabineri artiklile (Rabiner L. R., 1989, lk 263-264).

Kui esimesele probleemile leidus täpne lahendus, siis teise probleemi lahendamiseks on mitu erinevat viisi. Seda sellepärast, et „optimaalse“ seisundite jada jaoks võib olla mitu erinevat definitsiooni. Näiteks võib leida seisundid X_t, mis on individuaalselt igal ajahetkel kõige tõenäosemad. Selline optimaalsuse kriteerium maksimeerib õigete seisundite eeldatavat arvu.

Et leida sellise definitsiooni järgi optimaalset seisundite jada, tuletame meelde peatükis 1.3.1 defineeritud suuruse

$$\gamma_t(i) = \mathrm{P}(X_t = i|Y,\lambda),$$

st tõenäosuse, et oleme ajahetkel t seisundis i, kui teame vaatluste jada Y ja mudelit λ. $\gamma_t(i)$ on tõenäosuslik suurus: $\sum_{i=1}^{N}\gamma_t(i) = 1$.

Kasutades $\gamma_t(i)$, saame leida ajahetkel t kõige tõenäosema seisundi X_t väärtuse:

$$X_t = \underset{1\le i\le N}{\mathrm{argmax}}[\gamma_t(i)], \qquad 1 \le t \le T. \tag{1.12}$$

Kuigi eelnev avaldis maksimeerib oodatavat õigete seisundite arvu, võib saadava seisundite jadaga esineda probleeme. Näiteks, kui peidetud Markovi mudelil on nullilisi üleminekutõenäosusi ($a_{ij} = 0$ mingi i,j korral), siis võib tekkida seisundite jada, mis on

tegelikult ebasobiv. Seda sellepärast, et avaldis (1.12) leiab vaid igal ajahetkel tõenäoseima seisundi väärtuse, kuid ei võta üldse arvesse seisundite jadade esinemise tõenäosusi.

Üks variant sellist probleemi vältida on muuta optimaalsuse kriteeriumit. Näiteks võib leida hoopis seisundite jada, mis maksimeerib oodatavat õigete seisundite paaride (X_t, X_{t+1}) leidmist, kolmikute leidmist vms. Kuigi sellised meetodid võivad teatud juhtudel head olla, siis kõige levinum kriteerium on leida üks parim seisundite jada, st maksimeerida $\mathrm{P}(X|Y, \lambda)$, mis on ekvivalentne $\mathrm{P}(X, Y|\lambda)$ maksimeerimisega. Sellise tulemuse saamiseks kasutatakse Viterbi algoritmi.

1.4.1 Viterbi algoritm

Et leida parimat seisundite jada $X = \{X_1, X_2, \dots, X_T\}$ antud vaatluste jada $Y = \{Y_1, Y_2, \dots, Y_T\}$ korral, peame defineerima suuruse

$$\delta_t(i) = \max_{X_1, X_2, \dots, X_{t-1}} \mathrm{P}(X_1, X_2, \dots, X_t = i,\ Y_1, Y_2, \dots, Y_t|\lambda),$$

st $\delta_t(i)$ on suurim tõenäosus üle kõigi võimalike seisundite jadade ajahetkeni t, mis lõppeb seisundis i. Induktsiooni abil saame

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) a_{ij}\right] \cdot b_j(Y_{t+1}).$$

Et suurusele $\delta_T(i)$ vastavat seisundite jada kätte saada, peame meeles pidama, milline argument maksimeeris δ iga t ja j puhul. Seda teeme vektori $\psi_t(i)$ abil. Kogu Viterbi algoritm parima seisundite jada leidmiseks on kokkuvõtlikult järgmine:

1) Algväärtused:

$$\delta_1(i) = \pi_i b_i(Y_1), \qquad 1 \le i \le N,$$
$$\psi_1(i) = 0.$$

2) Rekursioon:

$$\delta_t(j) = \left[\max_{1 \le i \le N} \delta_{t-1}(i) a_{ij}\right] \cdot b_j(Y_{t+1}), \qquad 2 \le t \le T, \qquad 1 \le j \le N,$$

$$\psi_t(j) = \operatorname*{argmax}_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}], \qquad 2 \le t \le T, \qquad 1 \le j \le N.$$

3) Lõppväärtused:

$$P^* = \max_{1 \le i \le N} [\delta_T(i)],$$

$$X_T^* = \operatorname*{argmax}_{1 \le i \le N} [\delta_T(i)].$$

4) Seisundite jada leidmine (tagant poolt ette):

$$X_t^* = \psi_{t+1}(X_{t+1}^*), \qquad t = T-1, T-2, \dots, 1.$$

2 Autoregressiivsed peidetud Markovi mudelid

2.1 Sissejuhatus aegridadesse

Alapeatükid 2.1 ja 2.1.1 tuginevad peamiselt Raul Kangro 2016. aasta konspektile aines „Aegridade analüüs“ (Kangro, 2016).

Aegridade analüüsiga puutume kokku, kui uurime mingit kronoloogiliselt järjestatud vaatluste jada. Aegridasid kasutatakse paljudes erinevates valdkondades: uuritav tunnus võib olla näiteks aktsiahind, veetase jões vms. Uuritav tunnus $\{X_t\}$ on vaadeldud ajahetkedel $t \in T$; vaatleme hetkel vaid diskreetse ajaga aegridasid, kus väärtused vastavad võrdsete ajavahemike tagant tehtud mõõtmistele. See tähendab, et uuritava tunnuse X väärtusi mõõdetakse ajamomentidel $t_i = t_0 + ih$, kus $i \in \mathbb{N}$ või $i \in \mathbb{Z}$.

Aegread erinevad paljudest teistest statistilistest andmestikest selle poolest, et nende puhul ei saa eeldada mingi konkreetse jaotusega juhusliku suuruse sõltumatuid vaatlusi. Aegridade teoorias on sageli kasutatav eeldus aga nn statsionaarsuse nõue. Juhuslikku protsessi $\{X_t\}_{t\in\mathbb{Z}}$ nimetatakse (tugevalt) statsionaarseks, kui iga positiivse täisarvu m ning iga täisarvu q korral on juhuslikud vektorid $(X_1, \dots, X_m)$ ning $(X_{1+q}, \dots, X_{m+q})$ sama jaotusega. Kui iga positiivse täisarvu m ning iga täisarvu q korral on $(X_1, \dots, X_m)$ ning $(X_{1+q}, \dots, X_{m+q})$ kõik kuni k järku momendid võrdsed, siis nimetatakse protsessi X_t k-järku nõrgalt statsionaarseks.

Olgu meil tegemist teist järku nõrgalt statsionaarse protsessiga, siis juhul $m = 1$ järeldub, et

$$E(X_t) = \mu, \qquad D(X_t) = \sigma^2 \ \ \forall t$$

mingite konstantide μ ja σ korral. Samuti järeldub, et suuruste X_t ja X_{t+p} autokorrelatsioon $\rho(p)$ ja autokovariatsioon $\gamma(p)$ sõltub ainult ajamomentide vahest p:

$$\gamma(p) = cov\big(X_t, X_{t+p}\big), \qquad p \in \mathbb{Z},$$

$$\rho(p) = cor\big(X_t, X_{t+p}\big) = \frac{\gamma(p)}{\sigma^2}, \qquad p \in \mathbb{Z}.$$

Lisaks autokorrelatsioonile ja autokovariatsioonile pakub aegridade puhul huvi ka osakorrelatsioon. Juhuslike suuruste X_1 ja X_2 osakorrelatsiooniks pärast suuruste $Y_1, \dots, Y_k$ mõju eemaldamist nimetatakse suuruste $(X_1 - PX_1)$ ja $(X_2 - PX_2)$ vahelist korrelatsiooni, kus P on vähimruutude projektor suurustega $Y_1, \dots, Y_k$ määratud alamruumile. Teist järku nõrgalt

statsionaarse protsessi X k-ndat järku osaautokorrelatsioonikordajaks nimetatakse suuruste X_t ja X_{t-k} osakorrelatsiooni pärast suuruste $X_{t-1}, \dots, X_{t-(k-1)}$ mõju eemaldamist.

2.1.1 Lineaarsed mudelid ühemõõtmelise aegrea jaoks

Vaatleme selliseid aegrea mudeleid, kus aegrea hetkeväärtus avaldub lineaarse kombinatsioonina aegrea minevikuväärtustest ning juhusliku häirituse hetkeväärtusest ja minevikuväärtusest. Selliseid aegrea mudeleid nimetatakse lineaarseteks mudeliteks.

Üldiseks lineaarseks protsessiks nimetatakse protsesse, mis on esitatavad kujul

$$X_t = \mu + A_t + \sum_{i=1}^{\infty} \psi_i A_{t-i},$$

kus ψ_i on mingid tingimust $\sum_{i=1}^{\infty} \psi_i^2 < \infty$ rahuldavad reaalarvud, μ on reaalarv ning $\{A_t\}_{t\in\mathbb{Z}}$ on vähemalt teist järku statsionaarne tsentreeritud ning mittekorreleeritud väärtustega protsess.

Edaspidi vaatame alapeatükis 2.1.1 tsentreeritud protsesse, st eeldame, et $EX_t = 0\ \forall t$. Lisaks eeldame, et $cov(X_i, A_j) = 0\ \forall j > i$.

Praktikas kasutatakse aegrea mudeleid, kus on lõplik arv parameetreid, mida andmete põhjal hinnatakse. Lõpliku arvu kordajatega lineaarsete protsesside klassid on järgmised:

- Järguga p autoregressiivseteks protsessideks ehk $AR(p)$-protsessideks nimetatakse teist järku nõrgalt statsionaarseid protsesse kujul
$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + A_t. \qquad (2.1)$$
- Järguga q liikuva keskmisega protsessideks ehk $MA(q)$-protsessideks nimetatakse protsesse kujul
$$X_t = A_t - \sum_{i=1}^{q} \theta_i A_{t-i}.$$
- $ARMA(p,q)$-protsessideks nimetatakse teist järku nõrgalt statsionaarseid protsesse kujul
$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + A_t - \sum_{i=1}^{q} \theta_i A_{t-i}.$$

Uurime lähemalt $AR(p)$ mudelit. Selle protsessi käitumine on seotud polünoomi $\phi(x) = 1 - \sum_{i=1}^{p} \phi_i x^i$ nullkohtadega. Korrutades võrrandi (2.1) mõlemaid pooli suurusega X_{t-k} ning võttes keskväärtuse, saame

$$\gamma(k) = \sum_{i=1}^{p} \phi_i \gamma(k-i), \qquad k > 0. \tag{2.2}$$

Jagades võrrandi (2.2) suurusega $\gamma(0)$, saame võrduse

$$\rho(k) = \sum_{i=1}^{p} \phi_i \rho(k-i), \qquad k > 0. \tag{2.3}$$

$AR(p)$-protsesside puhul on lõpmatult paljud autokorrelatsioonid nullist erinevad. Sellise protsessi osaautokorrelatsioonikordajad on aga alates järgust $p+1$ võrdsed nulliga. Autokorrelatsioonikordajad $\rho(k)$ peavad definitsiooni kohaselt olema vahemikus $[-1, 1]$, mistõttu ei saa statsionaarsuse eeldus olla täidetud, kui mõni polünoomi ϕ nullkohtadest on mooduli poolest ühest väiksem ja vastav kordaja autokorrelatsioonide esituses nullkohtade kaudu on nullist erinev. Seetõttu on $AR(p)$-protsess nõrgalt teist järku statsionaarne parajasti siis, kui polünoomi $\phi(x) = 1 - \sum_{i=1}^{p} \phi_i x^i$ nullkohad on mooduli poolest ühest suuremad.

Vaatame lähemalt $AR(1)$ tüüpi mudeleid. Sellisel juhul avaldub mudeli kuju (2.1) kui

$$X_t = \phi_1 X_{t-1} + A_t.$$

Sellise mudeli korral $\phi(x) = 1 - \phi_1(x)$, mille ainsaks nullkohaks on $x_1 = \frac{1}{\phi_1}$. Seega on statsionaarsuse jaoks vajalik tingimuse $|\phi_1| < 1$ täidetus. Et kehtib (2.3), siis autokorrelatsioonid avalduvad kujul

$$\rho(k) = \phi_1^k, \qquad k = 1,2, \dots .$$

Seega kahanevad autokorrelatsioonide absoluutväärtused eksponentsiaalselt. Osakorrelatsioonid on alates järgust 2 võrdsed nulliga ning esimest järku osaautokorrelatsioon on võrdne $\rho(1)$-ga.

2.2 Autoregressiivse peidetud Markovi mudeli definitsioon

Aegrida võib vahel koosneda vaatlustest, mille on erinevatel ajahetkedel genereerinud erinevad protsessid. Sellisel juhul oleksid aegrea vaatlused justkui erinevatel ajahetkedel erinevates seisundites. Kui seisund muutub, võib aegreal toimuda oluline muutus keskväärtuses või hajuvuses. Sellisel juhul kasutatakse modelleerimiseks tihti autoregressiivseid peidetud Markovi mudeleid (*Autoregressive Hidden Markov Model,* ARHMM).

Nagu nimigi ütleb, on ARHMM kombinatsioon autoregressiivsest aegreast ja peidetud Markovi mudelist. Autoregressiivne struktuur näitab erinevate ajahetkede vaatluste omavahelist sõltuvust, peidetud Markovi mudel aga võtab arvesse peidetud seisundeid, millest vaatlused sõltuvad. Ökonomeetrias kutsutakse ARHMM-i ka „*time series with change in regime*" – muutuvate režiimidega aegrida. Algselt tutvustas selliseid mudeleid James D. Hamilton oma 1989. aasta artiklis (Hamilton, 1989).

Olgu $Y = \{Y_1, Y_2, \dots, Y_T\}$ vaatluste jada. Olgu $X = \{X_1, X_2, \dots, X_T\}$ peidetud seisundite jada, kusjuures igal ajahetkel t on seisundil X_t N erinevat võimalikku väärtust. Eeldatakse, et X on Markovi ahel üleminekumaatriksiga $A = [a_{ij}]$ ja algseisundite jaotusega $\pi = [\pi_i]$.

Autoregressiivne peidetud Markovi mudel on diskreetse ajaga juhuslik protsess, millel on kaks eeldust (Ailliot & Monbet, 2011, lk 2):

- seisundi X_t tinglik jaotus, kui on teada $\{X_{t'}\}_{t'<t}$ ja $\{Y_{t'}\}_{t'<t}$ väärtused, sõltub ainult seisundi X_{t-1} väärtusest: $\mathrm{P}(X_t|X_1, \dots, X_{t-1}) = \mathrm{P}(X_t|X_{t-1}),\ t = 2, 3, \dots$;
- vaatluse Y_t tinglik jaotus, kui on teada $\{X_{t'}\}_{t'<t}$ ja $\{Y_{t'}\}_{t'<t}$ väärtused, sõltub ainult seisundi X_t ja vaatluste $Y_{t-1}, \dots, Y_{t-p}$ väärtustest, kus p on autoregressiivse protsessi parameeter: $\mathrm{P}(Y_t|Y_1, \dots, Y_{t-1}, X_1, \dots, X_t) = \mathrm{P}(Y_t|X_t, Y_{t-1}, Y_{t-2}, \dots, Y_{t-p}),\ t \in \mathbb{N}$.

Autoregressiivse peidetud Markovi mudeli puhul vastab vaatluste jada Y autoregressiivsele $AR(p)$-protsessile, mida võib kirja panna kujul (Xuan, 2004, lk 38)

$$Y_t = \beta_0^{(X_t)} + \beta_1^{(X_t)} Y_{t-1} + \beta_2^{(X_t)} Y_{t-2} + \cdots + \beta_p^{(X_t)} Y_{t-p} + \varepsilon_t \tag{2.4}$$

või

$$Y_t = S_t \beta^{(X_t)} + \varepsilon_t, \tag{2.5}$$

kus

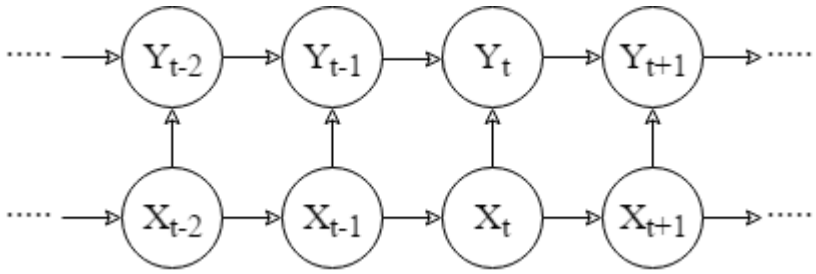
$$S_t = \left(1, Y_{t-1}, Y_{t-2}, \dots, Y_{t-p}\right),$$

$$\beta^{(X_t)} = \left(\beta_0^{(X_t)}, \beta_1^{(X_t)}, \beta_2^{(X_t)}, \dots, \beta_p^{(X_t)}\right)',$$

$$\varepsilon_t \sim i.i.d\ \ N(0, \sigma^2).$$

$\beta_i^{(X_t)}$ on autoregressiivse protsessi i. parameeter peidetud seisundi X_t korral. Seetõttu sõltubki vaatluse Y_t väärtus mitte ainult eelmisest p vaatlusest, vaid ka sama ajahetke seisundist X_t. ε_t on nn valge müra – i.i.d juhuslikud suurused, mille keskväärtus on 0 ning kovariatsioonimaatriks σ^2.

Tavaline HMM on erijuht ARHMM-ist juhul, kui $p = 0$. Kui $p = 1$, vastab ARHMM järgmisele skeemile:



Joonis 2.1. ARHMM-i struktuur juhul, kui p=1

Joonisel 2.1 kujutatud skeemi puhul saab valem (2.4) lihtsama kuju:

$$Y_t = \beta_0^{(X_t)} + \beta_1^{(X_t)} Y_{t-1} + \varepsilon_t.$$

Tihti pannakse aegridasid kirja ka kujul, kus vaatlustest on maha lahutatud keskväärtus. Sellise mudeli kuju puhul saaksime $p = 1$ korral valemi

$$(Y_t - \mu^{X_t}) = \beta_1^{X_t}(Y_{t-1} - \mu^{X_{t-1}}) + \varepsilon_t.$$

Tuletame meelde peatükist 1.2, et pidevate vaatluste korral saime vaatluste jaotuse kirja panna kui

$$b_i(y) = \sum_{m=1}^{M} C_{im} \mathcal{N}(\mu_{im}, \Sigma_{im}).$$

Vaatame erijuhtu sellest, kui $M = 1$, seega tihedusfunktsioon avaldub kujul

$$b_i(y) = \mathcal{N}(\mu_i, \Sigma_i).$$

Sellisel juhul on ARHMM-i keskväärtus $S_t\beta^{(X_t)}$ ja kovariatsioon σ^2 ning tihedusfunktsioon on esitatav kujul

$$\mathrm{P}(Y|Z_t,\theta) = \frac{1}{\sqrt{(2\pi\sigma^2)}}\exp[-(Y_t - S_t\beta^{(X_t)})'\big(Y_t - S_t\beta^{(X_t)}\big)], \qquad (2.6)$$

kus θ on parameetrid, mis sõltuvad tõenäosusjaotusest $B = (b_j(Y))$:

$$\theta = \big\{\sigma^2, \delta', \beta_1', \beta_2', \dots, \beta_p'\big), \qquad \delta = (\delta^1, \delta^2, \dots, \delta^N), \qquad \beta_i = (\beta_i^1, \beta_i^2, \dots, \beta_i^N),$$

Z_t on informatsioon viimasest $p+1$ seisundist ja viimasest p vaatlusest:

$$Z_t = \big\{X_{t-p}, \dots, X_{t-2}, X_{t-1}, X_t, Y_{t-p}, \dots, Y_{t-2}, Y_{t-1}\big\}.$$

2.3 ARHMM-i parameetrite hindamine

Järgnev alapeatükk põhineb Xuani magistritööl (Xuan, 2004, lk 46-50) ja Hamiltoni artiklil (Hamilton, 1990, lk 51-58).

Esimeses peatükis tutvusime Baum-Welchi meetodiga, mis põhineb EM-algoritmil. On näidatud (Hamilton, 1990, lk 51), et EM-algoritmi kasutamiseks ARHMM-i puhul leitakse ümberhinnatud parameetrid $\lambda_{l+1} = (A_{l+1}, \theta_{l+1}, \pi_{l+1})$ järgmiste valemite abil:

$$a_{ij}^{l+1} = \frac{\sum_{t=p+1}^{T} P(X_t = j, X_{t-1} = i|Y, \lambda_l)}{\sum_{t=p+1}^{T} P(X_{t-1} = i|Y, \lambda_l)},$$

$$\sum_{t=p+1}^{T}\sum_{X_t=1}^{N}\dots\sum_{X_{t-p}=1}^{N}\left.\frac{\partial \log \mathrm{P}(Y_t|Z_t,\lambda)}{\partial\theta}\right|_{\theta=\theta_{l+1}} \cdot P\big(X_t, \dots, X_{t-p}\big|Y, \lambda_l\big) = 0, \qquad (2.7)$$

$$\pi_{X_1,X_2,\dots,X_p}^{l+1} = \mathrm{P}\big(X_1, X_2, \dots, X_p\big|Y_1, \dots, Y_p, \lambda_l\big).$$

Et hinnata ARHMM-i parameetreid, tuletame meelde valemid (2.5) ja (2.6) autoregressiivse peidetud Markovi mudeli ja vaatluste vektori tingliku tihedusfunktsiooni kirjapanekuks. (2.6) on avaldis, mida tahame maksimeerida, seega logaritmime (2.6) ja leiame osatuletise $\beta^{(j)}$ ja σ^2 suhtes:

$$\frac{\partial \log \mathrm{P}(Y_t|Z_t,\lambda)}{\partial \beta^{(j)}} = \begin{cases} \frac{\left(Y_t - S_t\beta^{(j)}\right)S_t}{\sigma^2}, & \text{kui } X_t = j, \\ 0 & \text{muidu.} \end{cases} \tag{2.8}$$

$$\frac{\partial \log \mathrm{P}(Y_t|Z_t,\lambda)}{\partial \sigma^2} = \frac{\sigma^2}{2} - \frac{\left(Y_t - S_t\beta^{(X_t)}\right)^2}{2}. \tag{2.9}$$

Asendades valemid (2.8) ja (2.9) valemisse (2.7), saame

$$\sum_{t=p+1}^{T} \frac{\left(Y_t - S_t\beta_{(l+1)}^{(j)}\right)\cdot S_t}{\sigma_{(l+1)}^2} \cdot \mathrm{P}(X_t = j|Y,\lambda_l) = 0, \tag{2.10}$$

$$\left(\frac{1}{2}\sigma_{(l+1)}^2[T-p] - \sum_{t=p+1}^{T}\sum_{j=1}^{N} \frac{\left(Y_t - S_t\beta_{(l+1)}^{(j)}\right)^2}{2}\right) \cdot \mathrm{P}(X_t = j|Y,\lambda_l) = 0\,.$$

Hinnangu parameetritele $\beta_{(l+1)}^{(j)}$, mis lahendab võrrandi (2.10), saab leida vähimruutude meetodiga ($\hat{\beta} = (X^T X)^{-1}X^T y$):

$$\hat{\beta}_{(l+1)}^{(j)} = \left[\sum_{t=p+1}^{T} [\widetilde{S}_t(j)][\widetilde{S}_t(j)]'\right]^{-1}\left[\sum_{t=p+1}^{T} [\widetilde{S}_t(j)]\widetilde{Y}_t(j)\right], \quad j = 1,\dots,N \tag{2.11}$$

kus

$$\widetilde{Y}_t(j) = Y_t \cdot \sqrt{\mathrm{P}(X_t = j|Y,\lambda_l)},$$

$$\widetilde{S}_t(j) = S_t \cdot \sqrt{\mathrm{P}(X_t = j|Y,\lambda_l)}.$$

$\sigma_{(l+1)}^2$ hinnanguks saame

$$\hat{\sigma}_{(l+1)}^2 = \sum_{t=p+1}^{T}\sum_{j=1}^{N} \frac{\left(\widetilde{Y}_t(j) - \widetilde{S}_t(j)\beta_{(l+1)}^{(j)}\right)^2}{T-p}. \tag{2.12}$$

Üleminekumaatriksi $A = [a_{ij}]$ ja algsete tõenäosuste $\pi = [\pi_j]$ hinnangud on vastavalt

$$\hat{a}_{ij}^{(l+1)} = \frac{\sum_{t=p+1}^{T} \mathrm{P}(X_t = j, X_{t-1} = i|Y,\lambda_l)}{\sum_{t=p+1}^{T} \mathrm{P}(X_{t-1} = i|Y,\lambda_l)}, \tag{2.13}$$

$$\hat{\pi}_j^{(l+1)} = \mathrm{P}\left(X_p = j \middle| Y, \lambda_l\right). \tag{2.14}$$

Seega kokkuvõtlikult on EM-algoritm ARHMM-i parameetrite hindamiseks järgmine:

1) Valitakse algväärtused $\lambda_0 = (A_0, \theta_0, \pi_0)$, kus $A_0 = \left[a_{ij}\right]_0$, $\theta_0 = \left(\sigma_0^2, \beta_0^{1\prime}, \beta_0^{2\prime}, \dots, \beta_0^{N\prime}\right)'$, $\pi_0 = (\pi_0^1, \dots, \pi_0^N)'$.
2) Valemite (2.11)-(2.14) abil leitakse uued parameetrite väärtused $\lambda_1 = (A_1, \theta_1, \pi_1)$.
3) Protsessi korratakse, kuni EM-algoritm jõuab mingi fikseeritud punktini, mis vastab tõepärafunktsiooni lokaalsele maksimumile.

2.4 Silutud tõenäosuste arvutamine

Alljärgnev põhineb töödel (Hamilton, 1990, lk 67-69) ja (Xuan, 2004, lk 51-53).

Eelmises alapeatükis esitatud ARHMM-i parameetrite ümberhindamise valemid kasutavad igal sammul tõenäosusi $\mathrm{P}(X_t, X_{t-1}|Y)$ ja $\mathrm{P}(X_t|Y)$. Esimeses peatükis saime sellised tõenäosused kirja panna suuruste $\gamma_t(i)$ ja $\xi_t(i,j)$ abil. Autoregressiivsete peidetud Markovi mudelite korral on aga arvutuskäik pisut keerulisem. Järgnevalt on esitatud sammud tõenäosuse $\mathrm{P}(X_t, X_{t-1}, \dots, X_{t-p}|Y)$ arvutamiseks, kus p on autoregressiivse mudeli parameeter.

Tähistame vaatlused ajahetkel $1, \dots, t$ kui $Y^{(t)}$:

$$Y^{(t)} = (Y_1, \dots, Y_t).$$

Nagu varemgi, tähistame $\pi_{X_1, X_2, \dots, X_p} = \mathrm{P}\left(X_1, X_2, \dots, X_p \middle| Y_1, \dots, Y_p\right)$. Siis algoritm tõenäosuste $\mathrm{P}(X_t, X_{t-1}|Y)$ ja $\mathrm{P}(X_t|Y)$ arvutamiseks on järgmine:

1) Leiame $t = p + 1$ korral tõenäosused

$$\mathrm{P}\left(Y_{p+1} \middle| Y_1, \dots, Y_p\right) =$$

$$= \sum_{X_{p+1}=1}^{N} \sum_{X_p=1}^{N} \dots \sum_{X_1=1}^{N} \mathrm{P}\left(X_{p+1} \middle| X_p\right) \cdot \mathrm{P}\left(Y_{p+1} \middle| Y_1, \dots, Y_p, X_1, \dots, X_{p+1}\right) \cdot \pi_{X_1, X_2, \dots, X_p}$$

ja

$$\mathrm{P}\left(X_1, \dots, X_{p+1} \middle| Y_1, \dots, Y_{p+1}\right) =$$

$$= \frac{\mathrm{P}\left(X_{p+1}\middle|X_p\right) \cdot \mathrm{P}\left(Y_{p+1}\middle|Y_1, \dots, Y_p, X_1, \dots, X_{p+1}\right) \cdot \pi_{X_1, X_2, \dots, X_p}}{\mathrm{P}(Y_{p+1}|Y_1, \dots, Y_p)}$$

2) Leiame iteratiivselt $t = p+2, p+3, \dots, T$ korral tõenäosused

$$\mathrm{P}(Y_t|Y_1, \dots, Y_{t-1}) =$$

$$= \sum_{X_t=1}^{N} \sum_{X_{t-1}=1}^{N} \dots \sum_{X_{t-p-1}=1}^{N} \mathrm{P}(X_t|X_{t-1}) \cdot \mathrm{P}(Y_t|Y_1, \dots, Y_{t-1}, X_1, \dots, X_t)$$

$$\cdot \mathrm{P}\left(X_{t-1}, \dots, X_{t-p-1}\middle|Y_1, \dots, Y_{t-1}\right)$$

ja

$$\mathrm{P}\left(X_t, X_{t-1}, \dots, X_{t-p}\middle|Y_1, \dots, Y_t\right) =$$

$$= \frac{\sum_{X_{t-p-1}=1}^{N} \mathrm{P}(X_t|X_{t-1}) \cdot \mathrm{P}(Y_t|Y_1, \dots, Y_{t-1}, X_1, \dots, X_t) \cdot \mathrm{P}(X_{t-1}, \dots, X_{t-p-1}|Y_1, \dots, Y_{t-1})}{\mathrm{P}(Y_t|Y_1, \dots, Y_{t-1})},$$

kus $\mathrm{P}(X_t = j|X_{t-1} = i) = a_{ij}$.

3) Iga fikseeritud t puhul hindame etteulatuvad tõenäosused iga $\tau = t+1, \dots, t+p$ korral:

$$\mathrm{P}\left(X_{t-p}, X_{t-p+1}, \dots, X_{\tau-1}, X_\tau\middle|Y_1, \dots, Y_\tau\right) =$$

$$= \frac{\mathrm{P}(X_\tau|X_{\tau-1}) \cdot \mathrm{P}\left(Y_\tau\middle|Y_{\tau-1}, \dots, Y_{\tau-p}, X_\tau, \dots, X_{\tau-p}\right) \cdot \mathrm{P}\left(X_{\tau-1}, \dots, X_{t-p}\middle|Y_1, \dots, Y_{\tau-1}\right)}{\mathrm{P}(Y_\tau|Y_1, \dots, Y_{\tau-1})}.$$

4) Kanname edasi tõenäosuste arvutamist $\tau = t+p+1, \dots, T$ korral:

$$\mathrm{P}\left(X_\tau, \dots, X_{\tau-p}, X_t, \dots, X_{t-p}\middle|Y_1, \dots, Y_\tau\right) =$$

$$\frac{\sum_{X_{\tau-p-1}=1}^{N} \mathrm{P}(X_\tau|X_{\tau-1}) \cdot \mathrm{P}(Y_\tau|Y_1, \dots, Y_{\tau-1}, X_1, \dots, X_\tau) \cdot \mathrm{P}(X_{\tau-1}, \dots, X_{\tau-p-1}, X_t, \dots, X_{t-p}|Y_1, \dots, Y_{\tau-1})}{\mathrm{P}(Y_\tau|Y_1, \dots, Y_{\tau-1})}.$$

5) Silutud tõenäosused saame arvutada, liites kokku viimase p seisundi tõenäosused:

$$\mathrm{P}\left(X_t, \dots, X_{t-p}\middle|Y\right) = \sum_{X_T=1}^{N} \sum_{X_{T-1}=1}^{N} \dots \sum_{X_{T-p}=1}^{N} \mathrm{P}\left(X_T, \dots, X_{T-p}, X_t, \dots, X_{t-p}\middle|Y_1, \dots, Y_T\right) \cdot$$

2.5 *Segmental K-means* algoritm

Üks võimalus mudeli peidetud seisundeid hinnata on seega leida EM-algoritmi abil mudeli parameetrite hinnangud ning seejärel Viterbi algoritmiga peidetud seisundite hinnangud.

Praktikas kasutatakse vahel aga EM-algoritmi asemel hoopis *Segmental K-means Algorithm*'i (edaspidi SKA). Suurte andmestike korral võib SKA olla lihtsam kasutada ning kiirem ja efektiivsem.

Nagu EM-algoritm, on ka SKA iteratiivne protsess. Igal iteratsioonisammul leitakse parameetrite komplekti λ_l abil uus komplekt λ_{l+1} ning lõpuks jõutakse funktsiooni $g(\lambda) = \max_{X^*} \mathrm{P}(Y, X^*|\lambda)$ lokaalse maksimumini, kus X^* on optimaalne seisundite jada ja Y on vaatluste jada. Lõplik parameetrite komplekti hinnang on siis $\bar{\lambda} = \operatorname{argmax}_{\lambda}[g(\lambda)]$. (Rabiner & Juang, 1990)

SKA ja EM-algoritm erinevad selle poolest, et EM-algoritm maksimeerib suurust $\mathrm{P}(Y|\lambda)$, SKA aga suurust $\mathrm{P}(Y, X^*|\lambda)$. Rabiner ja Juang on artiklis „The Segmental K-means Algorithm for Estimating Parameters of Hidden Markov Models" näidanud, et ka SKA koondub.

Algoritmi sammud on järgmised (Xuan, 2004, lk 19-21, 60-63; Rabiner & Juang, 1990):

1) Olgu võimalikke erinevaid seisundeid N ja vaatlusi T. Valime N vaatlust klastrite tsentriteks ning ülejäänud $T - N$ vaatlused määrame neile lähimasse klastrisse. Kaugust mõõdetakse tavaliselt eukleidilise kaugusena. N klastrit vastavad peidetud seisundite algsetele hinnangutele.
2) Hindame algsed tõenäosused ja üleminekutõenäosuste maatriksi:

$$\hat{\pi}_i = \frac{seisundi\ X_t = i\ esinemiste\ arv}{vaatluste\ arv},$$

$$\hat{a}_{ij} = \frac{\text{ü}leminekute\ arv\ seisundist\ i\ seisundisse\ j}{\text{ü}leminekute\ arv\ seisundist\ i}.$$

3) Leiame vaatluste jaotusega seotud parameetrid. Juhul kui eeldame vaatluste normaaljaotust, hindame keskväärtuse ja dispersiooni, samuti leiame autoregressiivsed parameetrid seisundis i. β ja σ^2 hinnangud leiame valemite (2.11)-(2.12) abil ja keskväärtused erinevates seisundites valemiga

$$\hat{\mu}_i = \frac{\sum_{X_t=i} Y_t}{n_i},$$

kus n_i on seisundi i esinemiste arv, $1 \leq i \leq N$.

4) Leiame uue parameetrite komplekti $\hat{\lambda}$ abil uue optimaalse seisundite jada X^*, kasutades Viterbi algoritmi.
5) Kui hinnatud seisundite jadas X^* toimus muutusi, korratakse samme 2)-5).

3 HMM-i ja ARHMM-i rakendamine praktikas

Käesoleva magistritöö eesmärk on võrrelda, kas autoregressiivne peidetud Markovi mudel suudab täpsemini peidetud seisundeid hinnata kui tavaline peidetud Markovi mudel juhul, kui vaatluste sõltuvus pole tingitud mitte ainult peidetud Markovi ahelast. Sellise võrdluse läbiviimiseks tegi autor enda kirjutatud programmikoodidega läbi erinevate parameetritega simulatsioone. Samuti prooviti algoritmide võimekust teise põlvkonna sekveneerimisandmete näitel. Alljärgnevalt on alustuseks esitatud lihtsad näited selle kohta, kuidas HMM-ile ja ARHMM-ile vastavad algoritmid töötavad.

3.1 HMM-i hindamine

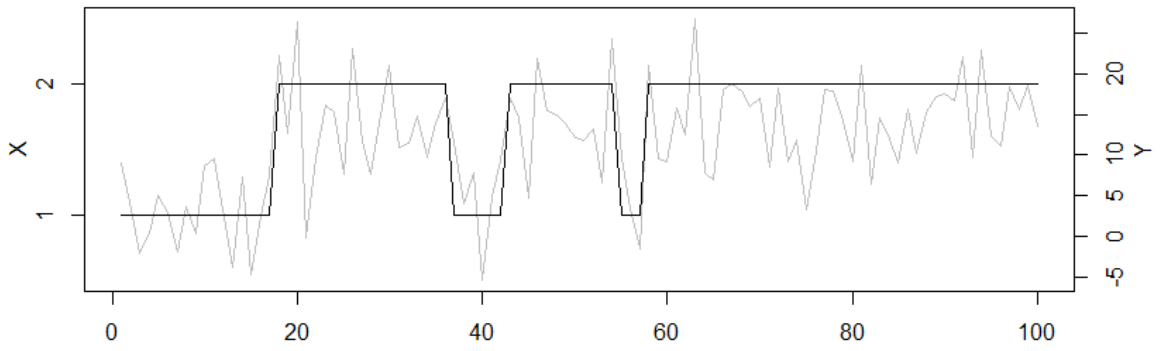
Näide 3.1. Esimesena vaatame lihtsat näidet selle kohta, kuidas suudavad peatükis 1 kirjeldatud meetodid hinnata HMM-i peidetud seisundeid kahe võimaliku seisundi korral ($N = 2$). Selleks genereeris autor $T = 100$ vaatlust, kusjuures $(Y_t|X_t = 1) \sim \mathcal{N}(5, 5^2)$ ning $(Y_t|X_t = 2) \sim \mathcal{N}(15, 5^2)$. Vaatlused vastavad tavalisele peidetud Markovi mudelile, st

$$\mathrm{P}(Y_t|Y_1, \dots, Y_{t-1}, X_1, \dots, X_t) = \mathrm{P}(Y_t|X_t).$$

Peidetud seisundite teoreetiline üleminekumaatriks on

$$A = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix}.$$

Genereeritud andmestiku vaatlused ja tegelikud seisundid on esitatud joonisel 3.1:



Joonis 3.1. Genereeritud andmestiku vaatlused ja tegelikud seisundid näites 3.1

Et hinnata HMM-i peidetud seisundeid, võime kasutada Baum-Welchi algoritmi või SKA-algoritmi. Kuna edaspidises kasutame autoregressiivsete peidetud Markovi mudelite hindamiseks *Segmental K-means* algoritmi, on ka HMM-i näidete puhul sama meetodit kasutatud.

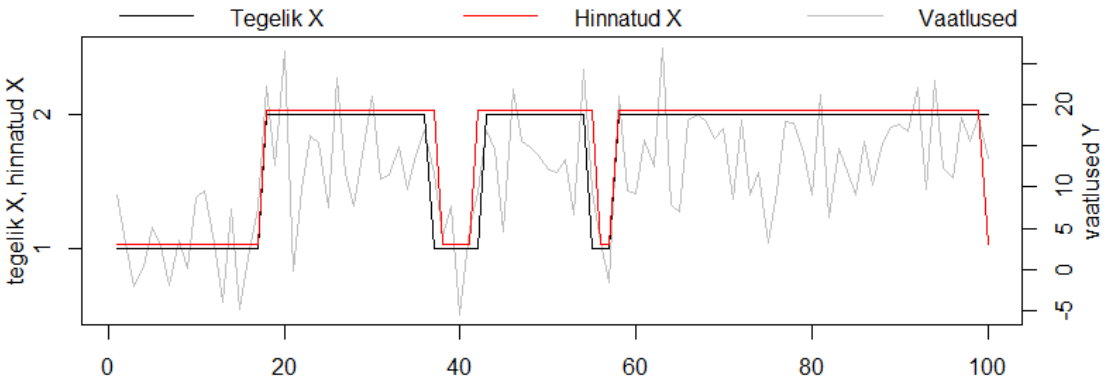
SKA kasutamiseks peame esiteks andma mingi reegli kohaselt ette algsed hinnangud peidetud seisunditele. Kui $N = 2$, on algsete hinnangute leidmiseks kõige lihtsam jagada vaatlused Y kahte klassi, kus ühes on vaatlused $Y_i < \bar{Y}$ ning teises vaatlused $Y_i \geq \bar{Y}$, kus $\bar{Y}$ on vaatluste keskmine. Need kaks vaatluste klassi vastavad seisunditele 1 ja 2. Seejärel leitakse seisundite hinnangute abil alghinnangud mudeli parameetritele. Kasutades alghinnanguid, leitakse SKA abil lõplikud hinnangud peidetud seisunditele ja mudeli parameetritele.

Tabelis 3.1 on esitatud andmestiku genereerimiseks kasutatud teoreetilised parameetrid, andmestiku tegelikud parameetrid (teades tegelikke seisundeid) ning SKA-ga saadud parameetrite hinnangud.

Tabel 3.1. Tegelikud ja hinnatud mudeli parameetrid näites 3.1

Parameetrid	Teoreetilised	Hinnangud, teades tegelikke seisundeid	SKA hinnangud
A	$\begin{pmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{pmatrix}$	$\begin{pmatrix} 0.885 & 0.115 \\ 0.027 & 0.973 \end{pmatrix}$	$\begin{pmatrix} 0.870 & 0.130 \\ 0.039 & 0.961 \end{pmatrix}$
π	$(1 \quad 0)$	$(1 \quad 0)$	$(0.24 \quad 0.76)$
μ	$(5 \quad 15)$	$(3.63 \quad 14.23)$	$(3.22 \quad 14.08)$
σ	$(5 \quad 5)$	$(4.87 \quad 5.40)$	$(4.91 \quad 5.39)$

Näeme, et sellise lihtsa näite puhul suudab algoritm väga hästi hinnata peidetud Markovi mudeli seisundeid. Joonisel 3.2 on kujutatud genereeritud andmestiku vaatlused, tegelikud seisundid ning SKA hinnatud seisundid. Näeme, et algoritm hindas õigesti 96 peidetud seisundit 100-st:



Joonis 3.2. Genereeritud andmestiku vaatlused ning tegelikud ja hinnatud seisundid näites 3.2

3.2 ARHMM-i hindamine

Näide 3.2. Järgmisena uurime, kui hästi suudab peatükis 2 kirjeldatud ARHMM-i hindamise meetod leida üles õiged peidetud seisundid. Vaatame juhtu, kus $N = 2$ ning $p = 1$, st vaatlused vastavad mudelile

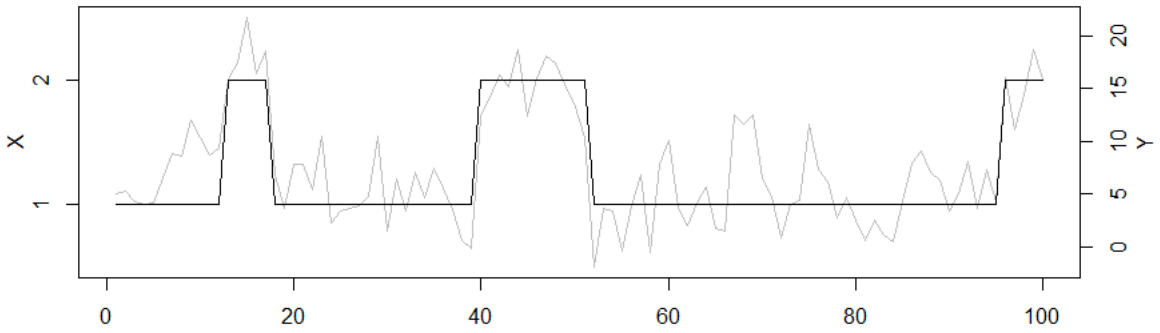
$$(Y_t - \mu^{X_t}) = \beta_1^{X_t}(Y_{t-1} - \mu^{X_{t-1}}) + \varepsilon_t,$$

kus $\varepsilon_t \sim i.i.d\ N(0, \sigma^2)$.

Autor genereeris näiteandmestiku, kus $T = 100$, $\mu = (5 \quad 15)$, $\sigma^2 = 3^2$ ning $\beta = (0.5 \quad 0.3)$. Peidetud seisundite teoreetiline üleminekumaatriks on

$$A = \begin{bmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{bmatrix}.$$

Genereeritud andmestiku vaatlused ja tegelikud seisundid on esitatud joonisel 3.3:



Joonis 3.3. Genereeritud vaatlused ja tegelikud seisundid näites 3.2

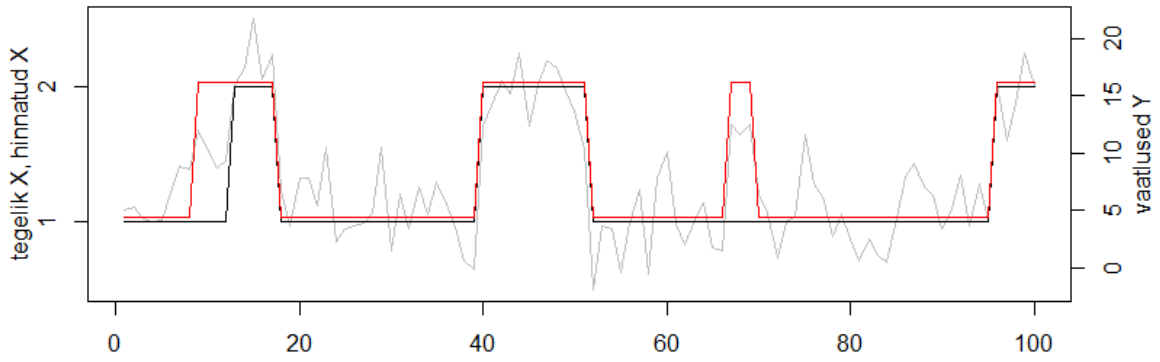
Et hinnata ARHMM-i peidetud seisundeid, kasutame taas SKA-d. Anname algoritmile ette algsed seisundite hinnangud samal viisil kui näites 3.1 ning leiame lõplikud hinnangud seisunditele ja mudeli parameetritele.

Antud andmestiku puhul saadi SKA-d kasutades järgmised tulemused:

Tabel 3.2. Tegelikud ja hinnatud mudeli parameetrid näites 3.2

Parameetrid	Teoreetilised	Hinnangud, teades tegelikke seisundeid	SKA hinnangud
A	$\begin{pmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{pmatrix}$	$\begin{pmatrix} 0.962 & 0.038 \\ 0.095 & 0.905 \end{pmatrix}$	$\begin{pmatrix} 0.944 & 0.056 \\ 0.107 & 0.893 \end{pmatrix}$
π	$(1 \quad 0)$	$(1 \quad 0)$	$(0.71 \quad 0.29)$
μ	$(5 \quad 15)$	$(5.26 \quad 15.72)$	$(4.70 \quad 14.58)$
σ	3	2.96	2.77
β	$(0.5 \quad 0.3)$	$(0.41 \quad 0.39)$	$(0.26 \quad 0.47)$

Käesolevas näites suudab algoritm hästi hinnata ARHMM-i parameetreid ja peidetud seisundeid. Algoritm hindas õigesti 93 peidetud seisundit 100-st (kujutatud joonisel 3.4).



Joonis 3.4. Genereeritud vaatlused, tegelikud seisundid ning SKA-ga hinnatud seisundid näites 3.2

Niisiis oleme näidetega 3.1 ja 3.2 vaadanud kahte juhtumit, kus andmed on genereeritud peidetud Markovi mudeli või autoregressiivse peidetud Markovi mudeliga ning peidetud seisundite leidmiseks on kasutatud vastavatele mudelitele sobivaid algoritme. Pelgalt nende kahe näite puhul ei saa loomulikult veel teha üldistavaid järeldusi, kui hästi need algoritmid töötavad. Lisaks soovime tegelikult uurida seda, kumb meetod töötab paremini juhul, kui andmed vastavad tegelikult autoregressiivsele peidetud Markovi mudelile. Tahame teada, kas selliste andmete puhul annab ARHMM-meetod parema tulemuse kui tavaline HMM-meetod?

3.3 HMM-i ja ARHMM-i võrdlus simulatsioonide korral

Et otsustada, kumb meetod sobib paremini, tuleb simuleerida palju erinevaid andmestikke ning kasutada nende peal HMM-ile ja ARHMM-ile vastavaid algoritme. Lihtsuse huvides jääme endiselt juhtude juurde, kus $N = 2$ ja $p = 1$. Kuna ARHMM-ile vastav SKA on üsna aeglane, simuleerime vaid andmestikke, kus $T = 100$. Sellise T korral suutis ARHMM-i simulatsioon ära käia u 15 minutiga, HMM-i simulatsioonid võtavad aega aga vaid mõne sekundi.

See, kui hästi meetodid töötavad, võib sõltuda ka mudeli tegelikest parameetritest. Autor valis mudeli parameetriteks $\pi = (1 \quad 0)$, $\mu = (5 \quad 15)$, $\sigma = 3$. Eeldades, et tulemused võiksid enim sõltuda üleminekumaatriksist ja autoregressiivse protsessi kordajatest, tehti simulatsioonid läbi neljal erineval juhul, kus üleminekumaatriks on kas

$$A = \begin{pmatrix} 0.95 & 0.05 \\ 0.05 & 0.95 \end{pmatrix} \text{ või } A = \begin{pmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{pmatrix}$$

ning autoregressiivse protsessi kordajad on kas

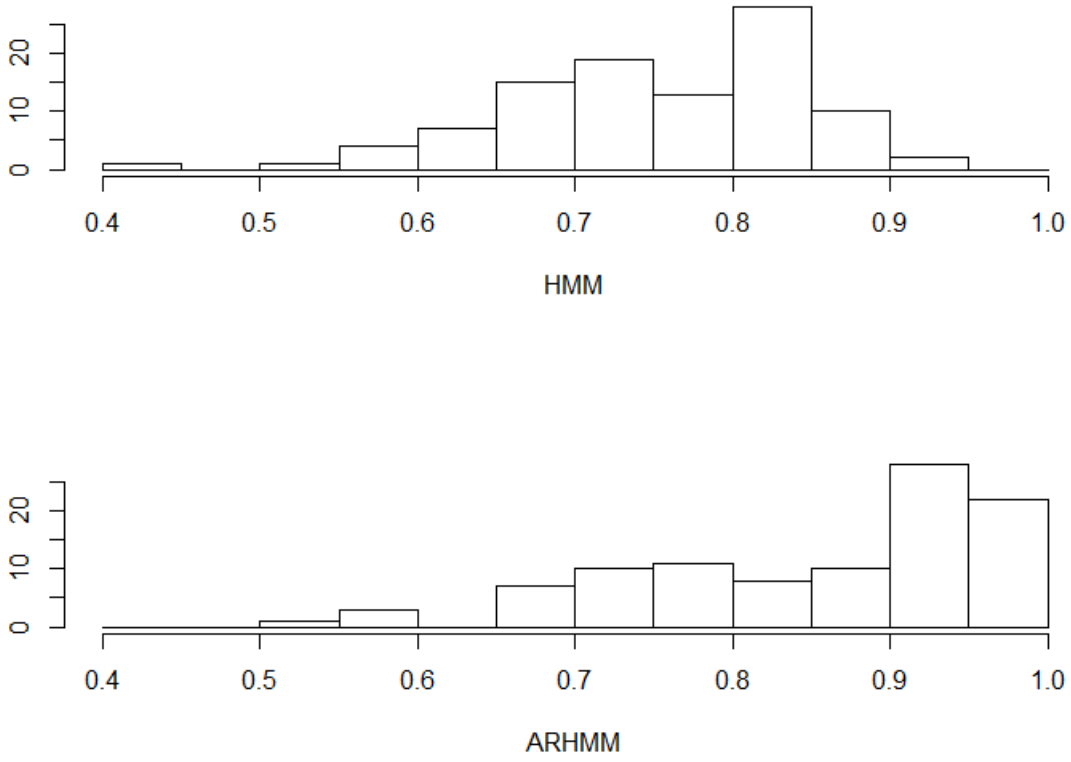
$$\beta = (0.3 \quad 0.4) \text{ või } \beta = (0.9 \quad 0.7).$$

Iga simulatsiooni korral tehti 100 katset, misjärel võrreldi ARHMM ja HMM meetodiga korrektselt hinnatud seisundite arvu. Alljärgnevas tabelis on simulatsioonidele vastavad mudelite teoreetilised parameetrid ning HMM ja ARHMM algoritmidega hinnatud õigete seisundite arvu karakteristikud. Lisaks on tabelis 3.3 toodud välja, mitmel juhul 100 katsest hindas ARHMM meetod rohkem õigeid seisundeid kui HMM.

Tabel 3.3. Simulatsioonide (100 ahelat) tulemused

Para-meetrid	Simulatsioon 1		Simulatsioon 2		Simulatsioon 3		Simulatsioon 4	
A	$\begin{pmatrix}0.95 & 0.05\\0.05 & 0.95\end{pmatrix}$		$\begin{pmatrix}0.7 & 0.3\\0.2 & 0.8\end{pmatrix}$		$\begin{pmatrix}0.95 & 0.05\\0.05 & 0.95\end{pmatrix}$		$\begin{pmatrix}0.7 & 0.3\\0.2 & 0.8\end{pmatrix}$	
π	$(1 \quad 0)$		$(1 \quad 0)$		$(1 \quad 0)$		$(1 \quad 0)$	
μ	$(5 \quad 15)$		$(5 \quad 15)$		$(5 \quad 15)$		$(5 \quad 15)$	
σ	3		3		3		3	
β	$(0.3 \quad 0.4)$		$(0.3 \quad 0.4)$		$(0.9 \quad 0.7)$		$(0.9 \quad 0.7)$	
	HMM	ARHMM	HMM	ARHMM	HMM	ARHMM	HMM	ARHMM
keskmine	97.4	98.2	93.3	94.8	83.3	86.3	75.9	86.1
min	84	74	86	84	55	57	41	53
max	99	100	99	100	99	100	92	100
st.hälve	2.2	3.1	2.9	3.0	9.6	9.7	9.3	11.2
ARHMM > HMM	72/100		65/100		65/100		90/100	

Tabelist 3.3 on näha, et kõigi 4 simulatsiooni puhul andis ARHMM-ile vastav SKA keskmiselt paremaid tulemusi kui tavalise HMM-i oma. Keskmised tulemused polnud enamasti kuigi palju erinevaid, kuid neljanda simulatsiooni korral hindas ARHMM õigesti keskmiselt u 10 seisundit rohkem. Selle simulatsiooni korral saavutas ARHMM lausa 90 juhul 100-st parema tulemuse kui HMM. Joonisel 3.5 on kujutatud neljanda simulatsiooni korral leitud õigesti hinnatud seisundite arvu histogrammid HMM ja ARHMM puhul.



Joonis 3.5. Õigesti hinnatud seisundite arvu jaotumine 4. simulatsiooni korral

Vaadates tabelit 3.3, on silmatorkav ka see, et HMM-ile vastav algoritm ei suutnud mitte ühegi katse puhul üles leida 100% õiget peidetud seisundite jada. ARHMM-i meetod suutis aga mudeli parameetritest hoolimata alati vähemalt mõne vaatluste jada puhul hinnata seisundite jada, mis oli täiesti õige. Samas on näha, et kõigi nelja simulatsioonikatse korral oli õigesti hinnatud seisundite arvu standardhälve ARHMM meetodi puhul pisut suurem kui tavalise HMM-i puhul.

Kokkuvõtlikult võib arvata, et juhul, kui andmed tegelikult vastavad autoregressiivsele peidetud Markovi mudelile, võiks SKA puhul ARHMM-i eripära arvestamine viia parema tulemuseni kui tavalise peidetud Markovi mudeli eeldamine. Järgnevalt vaatame, millise tulemuse annavad ARHMM ja HMM sekveneerimisandmete korral.

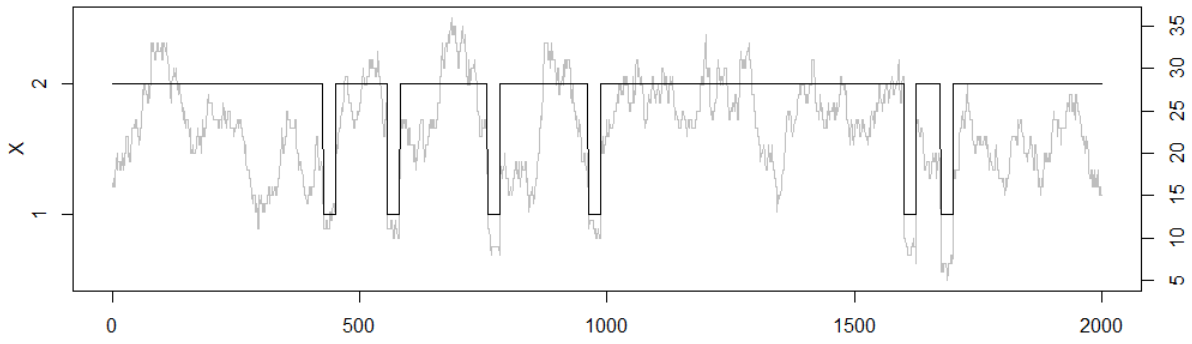
3.4 HMM-i ja ARHMM-i võrdlus teise põlvkonna sekveneerimisandmete korral

Sekveneerimine e järjendamine on protsess, mille käigus määratakse kindlaks monomeeride järjestus DNA molekulides. Käesoleva alapeatüki andmestik on simuleeritud, kasutades ühe konkreetse eestlase oletuslikku genotüüpi. Eeldati, et inimese genotüüp on täpselt selline, nagu teadlased arvavad ta olevat, ja selle põhjal simuleeriti, milliseid lugemeid võiks välja anda sekveneerimisplatvorm Illumina HiSeq2500. Simulatsioonid on teinud bioinformaatik Fanny-Dhelia Pajuste tarkvara wgsim (https://github.com/lh3/wgsim) abil.

Lugem (inglise keeles *read*) on ühe DNA segmendi sekveneeritud jupp. Antud näites on *paired-end* lugemid simuleeritud ligikaudu 30-se katvusega, vigade osakaal on 0.005, lugemi pikkus on 100. Andmestiku eesmärk on lugemeid kasutades tuvastada kõrvalekaldeid referentsgenoomist. Iga genoomse positsiooni korral on vaadatud positsiooni ümbritsevat DNA-järjestust ning loetud kokku, mitmel korral sellist DNA-järjestust lugemites nähti. See arv on antud andmestikus vaatluse Y_t rollis. Referentsgenoomiga kokkulangevas piirkonnas peaksime ootuspäraselt nägema vastavat DNA-lõiku u 30 lugemis. Kui aga ühes DNA koopiatest esineb mutatsioon (nt emalt päritud kromosoomis vastavat DNA-lõiku pole), siis peaksime vastavat DNA-järjestust nägema keskmiselt 15 lugemis. Antud näites esineb seega 2 erinevat võimalikku peidetud seisundit X_t: üks seisund tähistab muteerunud genoomi osa ja teine normaalset (referentsgenoomile vastavat) piirkonda.

Kuna üks lugem katab referentsgenoomis järjest paiknevaid genoomseid positsioone, siis ühe positsiooni üle- või alakaetus lugemitega tähendab, et tõenäoliselt on ka naaberpositsioonides lugemeid vastavalt oodatust rohkem/vähem. Seega on järjestikused Y_t väärtused omavahel sõltuvad isegi siis, kui teame, millises seisundis ahel parasjagu oli. Seetõttu võiks nende andmete puhul ARHMM-i kasutamine anda paremaid tulemusi kui tavaline HMM. Et algoritm ARHMM-i hindamiseks on väga aeglane, kasutati andmestikust vaid juppi pikkusega $T = 2000$. ARHMM-i hindamine võttis sellise suurusega andmestiku puhul aega üle kahe tunni, HMM-i hindamine aga vaid umbes 3 sekundit.

Joonisel 3.6 on esitatud näiteandmestiku vaatlused ja tegelikud seisundid:



Joonis 3.6. Näiteandmestiku vaatlused ja tegelikud seisundid

Tabel 3.4 kirjeldab andmestiku tegelikke parameetreid ning HMM-i ja ARHMM-iga hinnatud parameetreid. Samuti on mõlema meetodi puhul välja toodud, mitu peidetud seisundit meetod õigesti hinnata suutis.

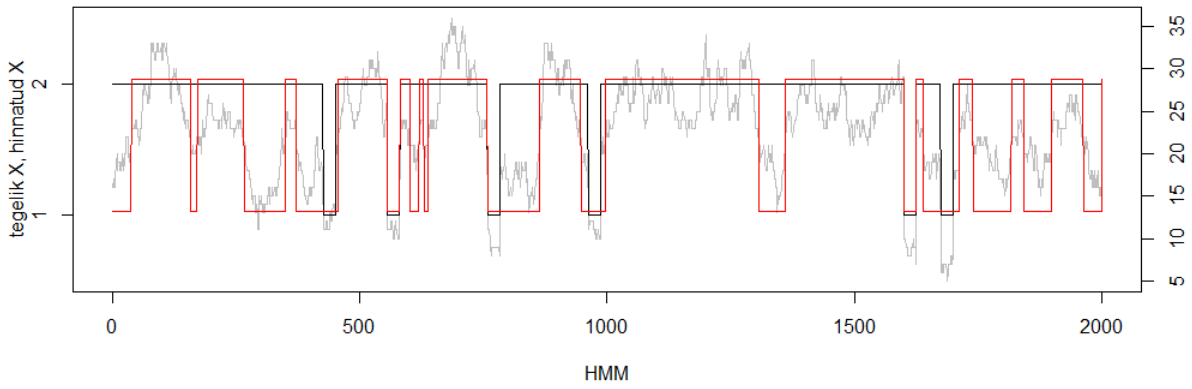
Tabel 3.4. Tegelikud ja hinnatud parameetrid sekveneerimisandmete näite korral

Parameetrid	Tegelikud	HMM	ARHMM
A	$\begin{pmatrix} 0.96 & 0.04 \\ 0.003 & 0.997 \end{pmatrix}$	$\begin{pmatrix} 0.980 & 0.020 \\ 0.011 & 0.989 \end{pmatrix}$	$\begin{pmatrix} 0.96 & 0.04 \\ 0.0027 & 0.9973 \end{pmatrix}$
π	$(0 \quad 1)$	$(0.375 \quad 0.625)$	$(0.0625 \quad 0.9375)$
μ	$(9.86 \quad 23.63)$	$(16.54 \quad 26.22)$	$(9.40 \quad 23.48)$
σ	0.98	$(4.01 \quad 3.22)$	0.93
β	$(0.97 \quad 0.99)$	-	$(0.92 \quad 0.98)$
õigesti hinnatud seisundeid		1401/2000	1975/2000

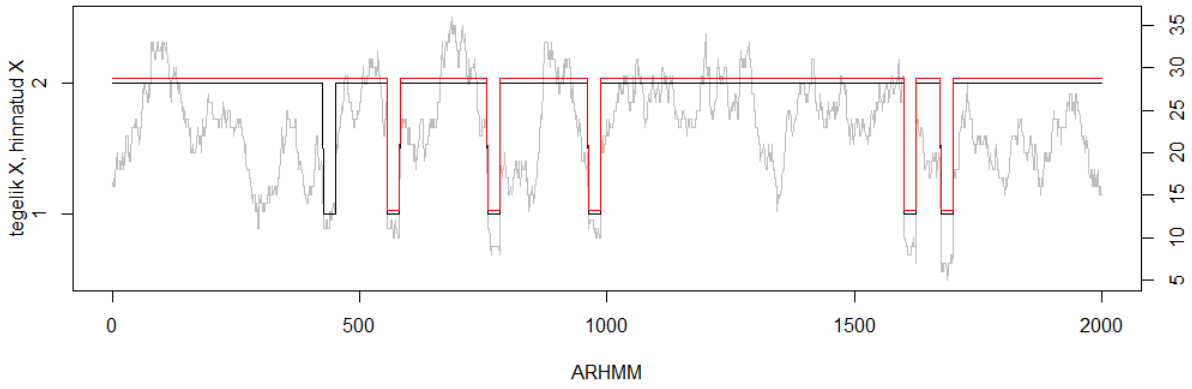
Tabelist on näha, et autoregressiivne peidetud Markovi mudel sobis nendele andmetele märgatavalt paremini kui tavaline peidetud Markovi mudel. ARHMM hindas seisundeid valesti vaid 25 juhul 2000-st. HMM hindas õigesti aga vaid u 70% seisunditest. Jooniste 3.7 ja 3.8 pealt järgmisel leheküljel on täpsemalt näha, kuidas hindasid HMM-ile ja ARHMM-ile vastavad algoritmid antud näite peidetud seisundeid.

Antud näite puhul olid autoregressiivse peidetud Markovi mudeli hinnangud peidetud seisunditele palju täpsemad kui tavalise peidetud Markovi mudeli omad. Seega võib öelda, et

juhul, kui andmete sõltuvus on tingitud ka muust kui vaid peidetud Markovi ahelast, peaks eelistama pigem autoregressiivse peidetud Markovi mudeli parameetreid hindavaid meetodeid. Loomulikult tuleb siin vaadata ka meetodite arvutuskiirust. Kui vaatlusi oleks veel rohkem kui antud näites, kuluks ARHMM-i hindamiseks märgatavalt rohkem aega, seega suurte andmestike puhul võib see muutuda tehniliselt võimatuks. Samuti muutub olukord keerulisemaks ja aeganõudvamaks, kui seisundeid on rohkem kui 2.



Joonis 3.7. Vaatlused, tegelikud seisundid ja HMM-iga hinnatud seisundid sekveneerimisandmete korral



Joonis 3.8. Vaatlused, tegelikud seisundid ja ARHMM-iga hinnatud seisundid sekveneerimisandmete korral

Kokkuvõte

Käesolevas magistritöös tutvustati peidetud Markovi mudelit ja autoregressiivset peidetud Markovi mudelit ning nende hindamismeetodeid. Peamiste meetoditena toodi välja teooria Baum-Welchi meetodi, Viterbi algoritmi ning *Segmental K-means* algoritmi kohta. Samuti uuriti, kui hästi töötavad peidetud Markovi mudel ja autoregressiivne peidetud Markovi mudel juhul, kui vaatlused vastavad tegelikult mingile autoregressiivsele peidetud Markovi mudelile. Selleks viidi läbi neli erinevat simulatsiooni 100 ahela korral ning kasutati näiteandmestikuna teise põlvkonna sekveneerimisandmeid.

Töö käigus viidi läbi neli erinevate parameetritega simulatsiooni, kus genereeriti 100 andmestikku juhul, kui erinevaid võimalikke peidetud seisundeid on $N = 2$ ning autoregressiivseid parameetreid on $p = 1$. Simulatsioone analüüsides selgus, et ARHMM suutis alati keskmiselt paremini peidetud seisundid üles leida kui tavaline HMM. Samuti ilmnes, et HMM ei suutnud ühegi genereeritud andmestiku puhul peidetud seisundeid 100% täpsusega hinnata. ARHMM suutis aga iga simulatsiooni puhul leida vähemalt ühel andmestikul 100% õige peidetud seisundite jada.

Analüüsi viimases osas kasutati ARHMM-i ja HMM-i hindamise algoritme näiteandmestiku puhul, mis kirjeldab simuleeritud teise põlvkonna sekveneerimisandmeid. Kuna antud andmestikus on vaatlused sõltuvad isegi siis, kui teame ahela seisundeid, võis eeldada, et ARHMM võiks nende andmetega paremini sobida kui tavaline HMM. Algoritmide kasutamise tulemusena selguski, et autoregressiivne peidetud Markovi mudel hindas õigesti 98.8% andmestiku peidetud seisunditest. Tavaline peidetud Markovi mudel suutis aga õigesti hinnata vaid 70.1% tegelikest peidetud seisunditest. Samas võttis ARHMM-i hindamine aega üle 2 tunni, kuid HMM-i hindamine vaid mõne sekundi.

Kokkuvõtlikult võib öelda, et juhul, kui andmete sõltuvus on tingitud ka muust peale peidetud Markovi ahela, suudab ARHMM peidetud seisundeid märksa paremini hinnata kui tavaline peidetud Markovi mudel. Negatiivse poolena tuleb tuua välja, et autoregressiivse peidetud Markovi mudeli hindamismeetod on võrreldes HMM-i hindamismeetodiga väga aeglane. Seega on sellise mudeli kasutamine praktikas võimalik vaid siis, kui andmestik on küllaltki lühike. Samuti võib meetodi kiirust ja efektiivsust mõjutada erinevate peidetud seisundite arv N ja autoregressiivsete parameetrite arv p. Käesolevas magistritöös kasutati vaid selliseid andmeid, kus $N = 2$ ja $p = 1$. Suurema arvu parameetrite korral võib ARHMM-i ja HMM-i võimekus

olla teistsugune ning vajab edasist uurimist. Samuti väärib uurimist, kas leidub väiksema keerukusega algoritme, mille abil adekvaatselt ARHMM-i hinnata.

Kasutatud kirjandus

Ailliot, P., & Monbet, V. (2011). Markov-switching autoregressive models for wind time series. *Environmental Modelling and Software, vol 30,* lk 92-101.

Baum, L. E., & Petrie, T. (1966). Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *Ann. Math. Statist. vol 37, no 6,* lk 1554-1563.

Baum, L. E., & Sell, G. R. (1968). Growth functions for transformations on manifolds. *Pacific Journal of Mathematics, vol 27, no 2,* 211-227.

Hamilton, J. D. (1989). A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. *Econometrica, vol 57,* lk 357-384.

Hamilton, J. D. (1990). Analysis of Time Series Subject to Changes in Regime. *Journal of Econometrics 45,* lk 39-70.

Kangro, R. (2016). *Aegridade analüüs.* Loengukonspekt, matemaatika ja statistika instituut, Tartu Ülikool.

Käärik, M. (2014). *Juhuslikud protsessid.* Loengukonspekt, matemaatika-informaatikateaduskond, Tartu Ülikool.

Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE, vol 77, no 2,* lk 257-286.

Rabiner, L. R., & Juang, B.-H. (1990). The Segmental K-means Algorithm for Estimating Parameters of Hidden Markov Models. *IEEE Transactions on Acoustics, Speech and Signal Processing, vol 38, no 9,* lk 1639-1641.

Rahimi, A. (30.12.2000. a.). *An Erratum for "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition"*. Allikas: http://alumni.media.mit.edu/~rahimi/rabiner/rabiner-errata/rabiner-errata.html. Vaadatud 01.05.2017

Zucchini, W., & MacDonald, I. L. (2009). *Hidden Markov Models for Time Series: an Introduction Using R.* Chapman & Hall/CRC.

Xuan, T. (2004). *Autoregressive Hidden Markov Model with Application in an El Nino Study.* Saskatoon: University of Saskatchewan.

Lisa. Simulatsioonide läbiviimiseks kasutatud kood

```
#SIMULATSIOONID samaaegselt ARHMM-i ja HMM-iga

start.time = Sys.time() #mõõdab, kui kaua kood käib

Q=100 #mitu simulatsiooni
T=100 #igal simulatsioonil mitu vaatluste jada
N=2 #erinevate võimalike seisundite arv

A=matrix(c(0.7, 0.3, 0.3, 0.7), nrow=2) #üleminekutõenäosuste maatriks
seisundid = c(1,2)  #võimalikud seisundid
s0 = 1 #X_1

#tegelike seisundite genereerimine
X = rep(NA, T)
s=s0
X[1]=s
for (i in 2:T) {
  s <- sample(seisundid, size=1, prob=A[s, ])
  X[i]=s
}

#joonis tegelikele seisunditele
plot(c(1:T),X,xlab=NA ,type="l", yaxt="n")
axis(at=X,labels=X,side=2)

mu_tegelik=c(5,15) #keskväärtused seisundites
sd_tegelik=3 #standardhälve
beta_tegelik=c(0.9, 0.7) #AR-protsessi parameetrid erinevate seisundite
korral

ARHMM=rep(0,Q) #ARHMM õigesti hinnatud seisundite arv
HMM=rep(0,Q) #HMM õigesti hinnatud seisundite arv

for (sim in 1:Q){

print(sim) #prindib välja, kui kaugel simulatsioonidega oleme

#andmestiku genereerimine
Y=rep(NA,T)
Y[1]=mu_tegelik[X[1]]+rnorm(1,0,sd_tegelik)

for (i in 2:T){
  Y[i]=mu_tegelik[X[i]] + beta_tegelik[X[i]]*(Y[i-1]-mu_tegelik[X[i-1]]) +
rnorm(1,0,sd_tegelik)
}
data=data.frame(X,Y)


######## 1. TAVALISE HMM-iga seisundite leidmine #######

#1) algne klasterdamine
data$Xest=1
data$Xest[data$Y>=(mean(data$Y))]=2

koondunud=FALSE

while(koondunud==FALSE){
  #2) algseisundite tõenäosused ja üleminekumaatriks
  pii=prop.table(table(data$Xest))
```

```
  a=prop.table(table(data$Xest[1:(T-1)],data$Xest[2:T]),1)

  #3) normaaljaotuse parameetrid seisundites
  mu=as.vector(by(data$Y,data$Xest,mean))
  sigma=as.vector(by(data$Y,data$Xest,var))

  #4) uue seisundite hinnangu leidmine
  delta=matrix(NA,nrow=T,ncol=N)
  psii=matrix(NA,nrow=T,ncol=N)

  delta1=-log(pii*dnorm(Y[1],mu,sqrt(sigma)))
  delta[1,]=delta1
  psii[1,]=rep(0,N)

  i=c(1:N)
  for (t in 2:T){
    for (j in 1:N){
      abi=delta[t-1,i]-log(a[j,i]*dnorm(Y[t],mu[j],sqrt(sigma[j])))
      delta_abi=min(abi)
      psii_abi=min(which(abi==delta_abi))
      delta[t,j]=delta_abi
      psii[t,j]=psii_abi
    }
  }

  P_tärn=max(delta[T,])
  q_T_tärn=min(which(delta[T,]==max(delta[T,])))

  state_est=rep(NA,T)
  state_est[T]=q_T_tärn
  for (t in (T-1):1){
    state_est[t]=psii[t+1,state_est[t+1]]
  }

  data$Xest2=state_est #uus seisundite hinnang

  oige=sum(data$X==data$Xest2) #mitu seisundit hindab õigesti
  #print(oige)
  koondunud=(sum(data$Xest==data$Xest2)==T) #kordab koodi seni, kuni Xest
ei muutu
  data$Xest=data$Xest2

}

HMM[sim]=oige


######## 2. ARHMM-iga seisundite leidmine #######

#1. algsed seisundite hinnangud
data$Xest=1
data$Xest[data$Y>(mean(data$Y))]=2

#parameetrite alghinnangud
a=prop.table(table(data$Xest[1:(T-1)],data$Xest[2:T]),1)
pii=prop.table(table(data$Xest))
pii2=a
mu =as.vector(by(data$Y,data$Xest,mean))

beta=rep(NA,N)
lugeja=rep(0,N)
```

```
nimetaja=rep(0,N)
for (t in 2:T){
  for (i in 1:N){
    if (data$Xest[t]==i){
      lugeja[i]=lugeja[i]+(data$Y[t]-mu[data$Xest[t]])*(data$Y[t-1]-
mu[data$Xest[t-1]])
      nimetaja[i]=nimetaja[i]+(data$Y[t]-mu[data$Xest[t]])^2
    }
  }
}
for (i in 1:N){
  beta[i]=lugeja[i]/nimetaja[i]
}
beta

t=c(2:T)
sigma2=sum((data$Y[t]-mu[data$Xest[t]]-beta[data$Xest[t]]*(data$Y[t-1]-
mu[data$Xest[t-1]]))^2)/T
sigma=sqrt(sigma2)

koondunud=FALSE

while(koondunud==FALSE){
  #2. Silutud tõenäosuste leidmine
  #P(X_t, X_(t-1)|Y) ja P(X_t|Y)
  P_Xt_Xt1_list=array(0, dim=c(N,N,T))

  #1) leiame t=p+1 korral tõenäosused P(Y_2|Y_1) ja P(X_1,X_2|Y_1,Y_2)
  t=2
  P_Y2=0
  for (i in 1:N){
    for (j in 1:N){
      P_Y2=P_Y2+(a[i,j]*dnorm(data$Y[t],
                              (c(mu[j],beta[j]) %*% c(1,data$Y[t-1]-
mu[i])),sigma)*pii[i])
    }
  }
  P_Y2

  P_Y_list=rep(0,T-1)
  P_Y_list[t-1]=P_Y2

  P_X2_X1=matrix(NA,ncol=N,nrow=N)
  for (i in 1:N){
    for (j in 1:N){
      P_X2_X1[i,j]=a[i,j]*dnorm(data$Y[t],
                                (c(mu[j],beta[j]) %*% c(1,data$Y[t-1]-
mu[i])),sigma)*pii[i]/P_Y2
    }
  }

  P_X1X2_list=array(0, dim=c(N,N,T-1))
  P_X1X2_list[,,t-1]=P_X2_X1

  P_Xt_Xt1_list[,,t]=P_X2_X1

  #2) leiame t=p+1,...,T korral samad tõenäosused
  for (t in 3:T){
    #P(Y_t|Y_1,...,Y_(t-1))
    P_Yt=0
    for (x1 in 1:N){
```

```
      for (x2 in 1:N){
        for (x3 in 1:N){
          P_Yt=P_Yt+(a[x2,x3]*dnorm(data$Y[t],
            (c(mu[x3],beta[x3]) %*% c(1,data$Y[t-1]-
mu[x2])),sigma)*P_X1X2_list[,,t-2][x1,x2])
        }
      }
    }
    P_Y_list[t-1]=P_Yt #lisame listi, et oleks mälus

    #P(X_t, X_(t-1)|Y_1,...,Y_t)
    P_Xt_Xt1=matrix(NA,ncol=N,nrow=N)
    for (x2 in 1:N){
      for (x3 in 1:N){
        P_Xt_Xt1[x2,x3]=a[x2,x3]*dnorm(data$Y[t],
                                       (c(mu[x3],beta[x3]) %*%
c(1,data$Y[t-1]-mu[x2])),sigma)*
          (sum(P_X1X2_list[,,t-2][,x2]))/P_Y_list[t-1]
      }
    }
    P_X1X2_list[,,t-1]=P_Xt_Xt1 #lisame listi
  }

  #3) iga t korral arvutame etteulatuvad tõenäosused, tau=t+1:
  #P(X_(t-1),X_t,X_(t+1)|Y_1,...,Y_(t+1))
  for (t in 3:(T-3)){

    P_X234=array(NA, dim=c(N,N,N))
    # X(t-1),X(t),X(t+1)
    for (x2 in 1:N){
      for (x3 in 1:N){
        for (x4 in 1:N){
          P_X234[x2,x3,x4]= a[x3,x4]*dnorm(data$Y[t+1],
                                (c(mu[x4],beta[x4]) %*% c(1,data$Y[t]-
mu[x3])),sigma)*
            P_X1X2_list[,,t-1][x2,x3]/P_Y_list[t]
        }
      }
    }

    P_X2345=array(NA, dim=c(N,N,N,N))
    #X(tau),X(tau-1),X(t),X(t-1)
    for (x2 in 1:N){
      for (x3 in 1:N){
        for (x4 in 1:N){
          for (x5 in 1:N){
            P_X2345[x2,x3,x4,x5]=a[x4,x5]*dnorm(data$Y[t+2],
                                (c(mu[x5],beta[x5]) %*% c(1,data$Y[t+1]-
mu[x4])), sigma)*
              P_X234[x2,x3,x4]/P_Y_list[t+1]
          }
        }
      }
    }

    eelmine=P_X2345

    for (indeks in (t+3):T){
      #indeks on tau rollis
      P_tau_tau1_t_t1=array(NA, dim=c(N,N,N,N))
```

```
      for (t1 in 1:N){
        for (t2 in 1:N){
          for (tau1 in 1:N){
            for (tau in 1:N){
              P_tau_tau1_t_t1[t1,t2,tau1,tau] =
a[tau1,tau]*dnorm(data$Y[indeks],
                                (c(mu[tau],beta[tau]) %*% c(1,data$Y[indeks-
1]-mu[tau1])), sigma)*

(eelmine[t1,t2,1,tau1]+eelmine[t1,t2,2,tau1])/P_Y_list[indeks-1]
            }
          }
        }
      }
      eelmine=P_tau_tau1_t_t1

    }

    #nüüd viimane mis saame (mis =eelmine siin koodis), on P(X_T,X_(T-
1),X_t,X_(t-1)|Y_1,...,Y_T)
    #eelmine[t1,t,tau1,tau], meil nüüd tau=T

    P_Xt_Xt1_Y=matrix(0,nrow=N,ncol=N)
    for (i in 1:N){
      for (j in 1:N){
        P_Xt_Xt1_Y[i,j]=sum(eelmine[i,j,,])
      }
    }
    P_Xt_Xt1_list[,,t]= P_Xt_Xt1_Y
  }

  ####t=T-2,T-1,T korral tuleb eraldi välja kirjutada
  t=T-2
  P_X234=array(NA, dim=c(N, N, N))
  for (x2 in 1:N){
    for (x3 in 1:N){
      for (x4 in 1:N){
        P_X234[x2,x3,x4]= a[x3,x4]*dnorm(data$Y[t+1],
                        (c(mu[x4],beta[x4]) %*% c(1,data$Y[t]-
mu[x3])),sigma)*
          P_X1X2_list[,,t-1][x2,x3]/P_Y_list[t]
      }
    }
  }

  P_X2345=array(NA, dim=c(N,N,N,N))

  #X(tau),X(tau-1),X(t),X(t-1), t=T-2 korral tau=T
  for (x2 in 1:N){
    for (x3 in 1:N){
      for (x4 in 1:N){
        for (x5 in 1:N){
          P_X2345[x2,x3,x4,x5]=a[x4,x5]*dnorm(data$Y[t+2],
           (c(mu[x5],beta[x5]) %*% c(1,data$Y[t+1]-mu[x4])), sigma)*
            P_X234[x2,x3,x4]/P_Y_list[t+1]
        }
      }
    }
  }

  P_Xt_Xt1_Y=matrix(0,nrow=N,ncol=N)
```

```
for (i in 1:N){
  for (j in 1:N){
    P_Xt_Xt1_Y[i,j]=sum(P_X2345[i,j,,])
  }
}
P_Xt_Xt1_list[,,t]= P_Xt_Xt1_Y

###
t=T-1
P_X234=array(NA, dim=c(N,N,N))
for (x2 in 1:N){
  for (x3 in 1:N){
    for (x4 in 1:N){
      P_X234[x2,x3,x4]= a[x3,x4]*dnorm(data$Y[t+1],
          (c(mu[x4],beta[x4]) %*% c(1,data$Y[t]-mu[x3])),sigma)*
        P_X1X2_list[,,t-1][x2,x3]/P_Y_list[t]
    }
  }
}

P_Xt_Xt1_Y=matrix(0,nrow=N,ncol=N)
for (i in 1:N){
  for (j in 1:N){
    P_Xt_Xt1_Y[i,j]=P_X234[1,i,j]+P_X234[2,i,j]
  }
}

P_Xt_Xt1_list[,,t]= P_Xt_Xt1_Y

###
t=T
P_Xt_Xt1_Y=P_X1X2_list[,,t-1]
P_Xt_Xt1_list[,,t]= P_Xt_Xt1_Y


#3. Uue seisundite jada hinnangu leidmine
P_Xt_Y=matrix(0,nrow=T,ncol=N)
P_Xt_Y[1,]=pii

for (t in 2:T){
  P_Xt_Y[t,1]=sum(P_Xt_Xt1_list[,,t][,1])
  P_Xt_Y[t,2]=sum(P_Xt_Xt1_list[,,t][,2])
}

for (t in 1:T){
  if (P_Xt_Y[t,1]>P_Xt_Y[t,2]){
    data$Xest2[t]=1
  }
  else data$Xest2[t]=2
}

#4. Parameetrite ümberhindamine
pii=prop.table(table(data$Xest2))
a=prop.table(table(data$Xest2[1:(T-1)],data$Xest2[2:T]),1)
mu =as.vector(by(data$Y,data$Xest2,mean))
beta=rep(NA,N)
lugeja=rep(0,N)
nimetaja=rep(0,N)
for (t in 2:T){
  for (i in 1:N){
    if (data$Xest2[t]==i){
```

```
        lugeja[i]=lugeja[i]+(data$Y[t]-mu[data$Xest2[t]])*(data$Y[t-1]-
mu[data$Xest2[t-1]])
        nimetaja[i]=nimetaja[i]+(data$Y[t]-mu[data$Xest2[t]])^2
      }
    }
  }
  for (i in 1:N){
    beta[i]=lugeja[i]/nimetaja[i]
  }
  t=c(2:T)
  sigma2=sum((data$Y[t]-mu[data$Xest2[t]]-beta[data$Xest2[t]]*(data$Y[t-1]-
mu[data$Xest2[t-1]]))^2)/T
  sigma=sqrt(sigma2)

  oige=sum(data$X==data$Xest2) #kui palju seisundeid hindab õigesti
  #print(oige)
  koondunud=(sum(data$Xest==data$Xest2)==T) #kordab algoritmi seni, kuni
Xest ei muutu

  data$Xest=data$Xest2
}

ARHMM[sim]=oige
}

end.time = Sys.time()
time.taken = end.time - start.time
time.taken
```

Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks

Mina, Hanna Läänemets,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Autoregressiivsed peidetud Markovi mudelid“, mille juhendaja on Märt Möls,

1.1. reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;

1.2. üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace´i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.

2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.

3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, 16. mail 2017