Skriveni Markovljevi modeli i primjene

Bauman, Tessa

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:217:731360

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Repository of the Faculty of Science - University of Zagreb

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Tessa Bauman

SKRIVENI MARKOVLJEVI MODELI I PRIMJENE

Diplomski rad

Voditelj rada:
prof. dr. sc. Bojan Basrak

Zagreb, 2020

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Sadržaj

Sadržaj **iii**

Uvod **1**

1 Osnovne definicije **2**
1.1 Markovljevi lanci 2
1.2 Skriveni Markovljevi modeli 3

2 Osnovni problemi **6**
2.1 Problem evaluacije 6
2.2 Problem dekodiranja 12
2.3 Problem učenja 15

3 Primjene **19**
3.1 Nepošteni kasino 19
3.2 Bioinformatika 22
3.3 Analiza teksta 26

4 Zaključak **32**

Bibliografija **33**

Uvod

Statistički modeli su važan alat za proučavanje i predviđanje stvarnih procesa. U ovom radu ćemo se baviti skrivenim Markovljevim modelom (Hidden Markov model - HMM). On pretpostavlja da postoji skriven proces opisan Markovljevim lancem koji direktno utječe na niz podataka koji promatramo. Promatrane podatke gledamo kao posljedicu kretanja lanca. Ono što razlikuje skriveni Markovljev model od mnogih drugih statističkih modela je što pretpostavlja da uzročnik postoji, no informacije o njemu nisu direktno dostupne. Cilj rada je objasniti model te riješiti glavne probleme. Za učinkovito korištenje modela, potrebno je naći parametre koji najbolje odgovaraju promatranim podacima te otkriti skriveni niz koji ih emitira.

Skriveni Markovljev model je uveden krajem 60-ih godina prošlog stoljeća te se prvotno koristio u prepoznavanju govora. Kroz vrijeme se pokazao korisnim u više područja od kojih su neka bioinformatika, detekcija promjene, financije, analiza teksta.

Pregled rada

Na početku rada ćemo definirati osnovne pojmove vezane uz Markovljeve lance kako bi zatim definirali i sam model. U poglavlju 2 su opisana glavna tri problema modela. Za svaki problem je dano rješenje te ponuđen algoritam kojime dolazimo do istog. U trećem poglavlju su obrađene tri primjene modela.

Poglavlje 1

Osnovne definicije

1.1 Markovljevi lanci

Definicija 1.1.1. *Neka je S prebrojiv skup. Slučajni proces $X = (X_n : n \geq 0)$ definiran na vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$ s vrijednostima u skupu S je Markovljev lanac ako vrijedi*

$$\mathbb{P}(X_{n+1} = j \,|\, X_n = i, X_{n-1} = i_{n-1}, ..., X_0 = i_0) = \mathbb{P}(X_{n+1} = j \,|\, X_n = i), \tag{1.1}$$

za svaki $n \in \mathbb{N}$ i za sve $i_0,...,i_{n-1}$, $i, j \in S$ za koje su obje uvjetne vjerojatnosti dobro definirane.

Svojstvo (1.1) zove se Markovljevo svojstvo te ono govori da vjerojatnost da je proces u trenutku $n + 1$ u proizvoljnom stanju ovisi o trenutnom stanju u trenutku n, ali ne i o prošlim stanjima.

U nastavku rada promatrat će se samo homogeni Markovljevi lanci tj. oni za koje vrijedi

$$\mathbb{P}(X_{n+1} = j \,|\, X_n = i) = \mathbb{P}(X_{m+1} = j \,|\, X_m = i), \quad \text{za svaki}\, m, n \geq 0,\ i, j \in S.$$

Drugim riječima, lanac je homogen ako desna strana u relaciji (1.1) ne ovisi o vremenu n.

Neka su sada

- $p_{ij} := \mathbb{P}(X_{n+1} = j \,|\, X_n = i)$, za svaki $n \geq 0$, $i, j \in S$,
- $\pi_i := \mathbb{P}(X_0 = i)$, za svaki $i \in S$.

Matrica $P = (p_{ij} : i, j \in S)$ naziva se *prijelazna matrica*, a vjerojatnosna distribucija $\Pi = (\pi_i : i \in S)$ *početna distribucija* Markovljevog lanca $X = (X_n : n \geq 0)$.

Teorem 1.1.2. *Neka je $X = (X_n : n \leq 0)$ Markovljev lanac. Tada za sve n≥0 i sva stanja i_0, i_1, i_{n-1}, i_n iz S vrijedi*

$$\mathbb{P}(X_0 = i_0, X_1 = i_1, ..., X_{n-1} = i_{n-1}, X_n = i_n) = \pi_{i_0}\, p_{i_0 i_1} \dots p_{i_{n-1} i_n}.$$

Dokaz. Dokaz teorema se nalazi u predavanjima prof. Vondračeka [6]. □

Napomena 1.1.3. *Markovljev lanac može se gledati i kao uređena trojka (S, P, Π), gdje je S skup stanja, P matrica prijelaza te Π početna distribucija. Ovakva karakterizacija pokazat će se korisna za bolje razumijevanje modela.*

1.2 Skriveni Markovljevi modeli

Neformalno, skriveni Markovljev model sastoji se od dva slučajna niza; Markovljevog lanca čija stanja nisu direktno dostupna te niza opažanja koja su rezultat kretanja skrivenog lanca. Oba niza mogu biti definirana s vrijednostima u prebrojivom i neprebrojivom skupu, ali za lakšu prezentaciju modela, njegovih problema te pripadnih algoritama, ograničit ćemo se samo na prebrojiv skup stanja Markovljevog lanca (kao što je zadano u definiciji 1.1.1) i prebrojiv skup opažanja. Unatoč tome, algoritmi koji će biti navedeni slično vrijede i u slučaju neprebrojivog skupa vrijednosti.

Definicija 1.2.1. *Neka je $X = (X_n : n \geq 0)$ Markovljev lanac na skupu stanja S, s matricom prijelaza $P = (p_{ij} : i, j \in S)$ i početnom distribucijom $\Pi = (\pi_i : i \in S)$. Neka je $Y = (Y_n : n \geq 0)$ niz slučajnih varijabli s vrijednostima u prebrojivom skupu O. Slučajan proces $(X, Y) = \{(X_n, Y_n) : n \geq 0)\}$ zovemo skriveni Markovljev model ako vrijedi:*

$$\begin{aligned} \mathbb{P}(Y_n = k_n \,|\, X_0 = j_0, Y_0 = k_0, X_1 = j_1, Y_1 = k_1, ...X_N = j_N, Y_N = k_N) \\ = \mathbb{P}(Y_n = k_n \,|\, X_n = j_n), \end{aligned} \tag{1.2}$$

za svaki $n, N \geq 0, n \leq N,\ j_0, j_1, ..., j_N \in S$ i sve $k_0, k_1, ..., k_N \in O$.

Svojstvo (1.2) zahtijeva da Y_n ovisi samo o X_n tj. stanju lanca u istom trenutku. Kao i kod Markovljevih lanaca, promatrat će se samo homogeni Markovljevi modeli, oni za koje uvjetna vjerojatnost iz (1.2) ovisi samo o stanjima oba niza, a ne i o vremenskom trenutku n.

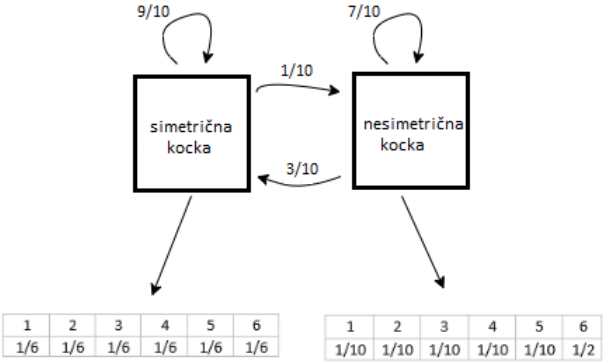
Napomena 1.2.2. *Skriveni Markovljev model možemo okarakterizirati s uređenom petorkom H=(S, O, P, Ψ, Π) gdje S, P i Π predstavljaju parametre Markovljevog lanca $X = (X_n : n \geq 0)$, a O i Ψ sljedeće:*

- *O skup emisijskih simbola tj. vrijednosti niza slučajnih varijabli Y,*

- $\Psi = (\psi_{jk} : j \in S, k \in O)$ *matrica emisijskih vjerojatnosti, uz*

$$\psi_{jk} := \mathbb{P}(Y_n = k \,|\, X_n = j).$$

Primjer 1.2.3. *U igri postoje dvije naizgled iste kocke. Jedna je savršeno simetrična dok druga blago favorizira šesticu. U svakom bacanju baca se samo jedna kocka, no budući da su prividno jednake, igrač ne zna koju, te se nekada zamijene. Ono što se pak može promatrati je niz brojeva koji su dobiveni u bacanju te na temelju tih opažanja procijeniti koja je kocka bila korištena u kojem trenutku.*



Slika 1.1: Motivacijski primjer

Po napomeni 1.2.2, trebaju nam parametri modela za ovaj primjer. Niz korištenih kocaka predstavlja Markovljev lanac $X = (X_n : n \geq 0)$ *koji emitira brojeve tj. opažanja* $Y = (Y_n : n \geq 0)$.

- $S = \{s, n\}$ *skup stanja lanca koji se sastoji od s(imetrične) i n(esimetrične) kocke*
- $O = \{1, 2, 3, 4, 5, 6\}$ *skup emisijskih simbola*
- *P matrica prijelaza*

$$\begin{pmatrix} 9/10 & 1/10 \\ 3/10 & 7/10 \end{pmatrix},$$

gdje je $p_{ss} = \mathbb{P}(X_{n+1} = s \,|\, X_n = s)$ *vjerojatnost da je nakon simetrične kocke opet bačena simetrična kocka,* $p_{ns} = \mathbb{P}(X_{n+1} = s \,|\, X_n = n)$ *vjerojatnost da je nakon nesimetrične kocke došlo do zamjene te je bačena simetrična kocka,* $p_{sn} = \mathbb{P}(X_{n+1} = n \,|\, X_n = s)$ *vjerojatnost da je nakon simetrične kocke došlo do zamjene te je bačena nesimetrična kocka i* $p_{nn} = \mathbb{P}(X_{n+1} = n \,|\, X_n = n)$ *vjerojatnost da je nakon nesimetrične kocke opet bačena nesimetrična kocka.*

- Ψ *matrica emisijskih vjerojatnosti*

$$\begin{pmatrix} 1/6 & 1/6 & 1/6 & 1/6 & 1/6 & 1/6 \\ 1/10 & 1/10 & 1/10 & 1/10 & 1/10 & 1/2 \end{pmatrix}$$

  gdje su npr. $\psi_{s1} = \mathbb{P}(Y_n = 1 \,|\, X_n = s)$ *vjerojatnost dobivanja 1 ako je bačena simetrična kocka,* $\psi_{n1} = \mathbb{P}(Y_n = 1 \,|\, X_n = n)$ *vjerojatnost dobivanja 1 ako je bačena nesimetrična kocka i* $\psi_{n6} = \mathbb{P}(Y_n = 6 \,|\, X_n = n)$ *vjerojatnost dobivanja 6 ako je bačena nesimetrična kocka.*

- Π *početna distribucija:* $\pi_s = \mathbb{P}(X_0 = s) = \frac{1}{2}$, $\pi_n = \mathbb{P}(X_0 = n) = \frac{1}{2}$.

Poglavlje 2

Osnovni problemi

Skriveni Markovljevi modeli omogućuju jednostavno modeliranje procesa iz stvarnog svijeta, ali da bi bili korisni potrebno je riješiti tri glavna problema:

1. Problem evaluacije - odrediti vjerojatnost da je dani model generirao dani slijed opažanja,
2. Problem dekodiranja - odrediti optimalan slijed skrivenih stanja s obzirom na dani niz opažanja,
3. Problem učenja - odrediti model koji maksimizira vjerojatnost emitiranja danog niza simbola.

2.1 Problem evaluacije

Neka je $(X, Y) = \{(X_n, Y_n) : n \geq 0)\}$ HMM karakteriziran s $H = (S, O, P, \Psi, \Pi)$ gdje je S skup stanja skrivenog Markovljevog lanca X, O skup emisijskih (opservacijskih) simbola, $P = (p_{ij} :$ i, j $\in$ S) matrica prijelaza lanca X, $\Psi = (\psi_{jk} : j \in S, k \in O)$ matrica emisijskih vjerojatnosti i $\Pi = (\pi_i :$ i $\in$ S) početna distribucija lanca X. Neka su $k_1, k_2, ..., k_T \in O$. Kod evaluacijskog problema, pitanje je kolika je vjerojatnost da je model H emitirao dani niz simbola $k_1, k_2, ..., k_T$ tj.

$$\mathbb{P}(Y_1 = k_1, Y_2 = k_2, ..., Y_T = k_T \,;\, H). \tag{2.1}$$

Rješenje ovog problema posebno je korisno jer omogućava praktično uspoređivanje različitih modela. Od više ponuđenih modela, može se odabrati onaj koji najbolje odgovara opažanjima. Konkretno, onaj za kojeg je vjerojatnost (2.1) najveća.

Rješenje evaluacijskog problema

Kada se radi samo o Markovljevom lancu, vjerojatnost da je dobiven određeni niz vrlo se lako odredi pomoću teorema 1.1.2. Međutim, u slučaju skrivenog Markovljevog lanca ovaj postupak je kompliciraniji budući da su samo opažanja dostupna, a stanja nepoznata.

Napomena 2.1.1. *Ako su i stanja lanca $j_0, j_1, ..., j_T \in S$ poznata, tada je*

$$\mathbb{P}(Y_1 = k_1, Y_2 = k_2, ..., Y_T = k_T \,|\, X_1 = j_1, X_2 = j_2, ..., X_T = j_T) = \prod_{i=1}^{T} \psi_{j_i k_i}$$

vjerojatnost slijeda simbola uz poznati slijed stanja, a

$$\mathbb{P}(Y_1 = k_1, Y_2 = k_2, ..., Y_T = k_T, X_1 = j_1, X_2 = j_2, ..., X_T = j_T) = \prod_{i=1}^{T} \psi_{j_i k_i} \cdot \prod_{i=1}^{T} p_{j_{i-1} j_i} \quad (2.2)$$

vjerojatnost slijeda simbola i slijeda stanja u danom modelu.

Primjer 2.1.2. *Neka je dan model s kockama iz primjera 1.2.3. Vjerojatnost da je dobiven niz brojeva 1, 1, 6 ako je poznato da je u prvom bacanju korištena simetrična kocka, a u druga dva nesimetrična, je:*

$$\mathbb{P}(Y_1 = 1, Y_2 = 1, Y_3 = 6 \,|\, X_1 = s, X_2 = n, X_3 = n) = \psi_{s1} \cdot \psi_{n1} \cdot \psi_{n6}.$$

Vjerojatnost presjeka ovih događaja je:

$$\mathbb{P}(Y_1 = 1, Y_2 = 1, Y_3 = 6, X_1 = s, X_2 = n, X_3 = n) = \psi_{s1} \cdot \psi_{n1} \cdot \psi_{n6} \cdot \pi_s \cdot pi_{sn} \cdot pi_{nn}.$$

Budući da su stanja od X ipak skrivena, kako bi se dobila tražena vrijednost (2.1), potrebno je sumirati vjerojatnost (2.2) po svim kombinacijama niza stanja lanca X. Uočimo da zato

$$\mathbb{P}(Y_1 = k_1, Y_2 = k_2, ..., Y_T = k_T) = \sum_{j_0, j_1, ..., j_T \in S} \prod_{i=1}^{T} \psi_{j_i k_i} \cdot \prod_{i=1}^{T} p_{j_{i-1} j_i} \quad (2.3)$$

daje rješenje evaluacijskog problema.

Primjer 2.1.3 (nastavak primjera 2.1.2)**.** *Vjerojatnost da je zadani model s kockama emitirao niz brojeva 1, 1, 6 je:*

$$\begin{aligned}\mathbb{P}(Y_1 = 1, Y_2 = 1, Y_3 = 6) = {} & \psi_{s1} \cdot \psi_{s1} \cdot \psi_{s6} \cdot \pi_s \cdot p_{ss} \cdot p_{ss} + \\ & \psi_{s1} \cdot \psi_{s1} \cdot \psi_{n6} \cdot \pi_s \cdot p_{ss} \cdot p_{sn} + \\ & \psi_{s1} \cdot \psi_{n1} \cdot \psi_{n6} \cdot \pi_s \cdot p_{sn} \cdot p_{nn} + ...\end{aligned}$$

Rješenje prvog problema je relativno jednostavno, no nije i efikasno. Za skriveni Markovljev model s N stanja ($|S| = N$) i niz opažanja duljine T, direktan izračun (2.3) uključuje broj računskih operacija reda $T \cdot N^T$, budući da postoji N^T mogućih nizova skrivenih stanja i za svaki takav potrebno je okvirno $2T$ operacija u sumi. Preciznije, potrebno je $(2T-1) \cdot N^T$ množenja i $N^T - 1$ zbrajanja. Već u slučaju malih N i T, taj broj je prilično velik. Na sreću, postoji puno efikasniji algoritam za rješenje problema evaluacije.

Algoritam za evaluacijski problem

Definicija 2.1.4. *Neka su dana opažanja $k_1, k_2, ..., k_T \in O$. Definiramo*

$$\alpha_t(j) := \mathbb{P}(Y_1 = k_1, Y_2 = k_2, ..., Y_t = k_t, X_t = j), \text{ za svaki } t \leq T,\ j \in S. \tag{2.4}$$

Ovako definirani parametri uobičajeno se nazivaju forward varijablama. $\alpha_t(j)$ predstavlja vjerojatnost da se model u trenutku t nalazi u stanju j i da je emitirao prvih t elemenata niza opažanja.

Propozicija 2.1.5. *Za forward varijable vrijedi rekurzija:*

$$\alpha_t(j) = \sum_{i \in S} \alpha_{t-1}(i)\, p_{ij}\, \psi_{jk_t},\ \ j \in S, 1 \leq t \leq T.$$

Dokaz.

$$\begin{aligned}
\alpha_t(j) &= \mathbb{P}(Y_1 = k_1, ..., Y_{t-1} = k_{t-1}, Y_t = k_t, X_t = j) \\
&= \sum_{i \in S} \mathbb{P}(Y_1 = k_1, ..., Y_{t-1} = k_{t-1}, Y_t = k_t, X_t = j, X_{t_1} = i) \\
&= \sum_{i \in S} \mathbb{P}(Y_t = k_t \,|\, Y_1 = k_1, ..., Y_{t-1} = k_{t-1}, X_t = j, X_{t-1} = i) \\
&\qquad \cdot \mathbb{P}(Y_1 = k_1, ..., Y_{t-1} = k_{t-1}, X_t = j, X_{t-1} = i) \\
&= \sum_{i \in S} \mathbb{P}(Y_t = k_t \,|\, X_t = j)\, \mathbb{P}(X_t = j \,|\, Y_1 = k_1, ..., Y_{t-1}, X_{t-1} = i) \\
&\qquad \cdot \mathbb{P}(Y_1 = k_1, ..., Y_{t-1}, X_{t-1} = i) \\
&= \mathbb{P}(Y_t = k_t \,|\, X_t = j) \sum_{i \in S} \mathbb{P}(X_t = j \,|\, X_{t-1} = i)\, \mathbb{P}(Y_1 = k_1, ..., Y_{t-1}, X_{t-1} = i) \\
&= \psi_{jk_t} \sum_{i=1}^{N} p_{ij}\, \alpha_{t-1}(i).
\end{aligned}$$

□

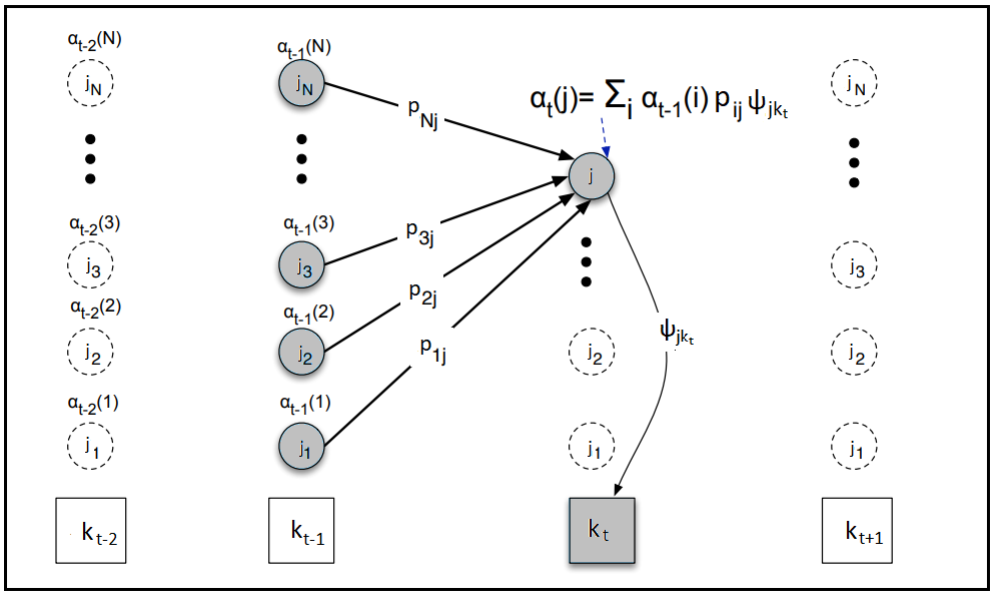
Forward algoritam može se opisati sljedećim koracima:

1. Inicijalizacija:
$$\alpha_1(j) = \pi_j \psi_{jk_1}, \quad j \in S$$

2. Rekurzija:
$$\alpha_t(j) = \sum_{i \in S} \alpha_{t-1}(i) p_{ij} \psi_{jk_t}, \quad j \in S, 1 \leq t \leq T$$

3. Kraj:
$$\mathbb{P}(Y_1 = k_1, Y_2 = k_2, ..., Y_T = k_T) = \sum_{i=1}^{N} \alpha_T(i).$$

Prvi korak *forward* varijabli pridružuje vjerojatnost (2.4) za niz duljine 1, tj. uz $t = 1$. Korak rekurzije je ilustriran na slici 2.1.



Slika 2.1: Ilustracija niza operacija (varijabli) potrebnih za računanje $\alpha_t(j)$. Slika je preuzeta iz Jurafsky i Martin [3].

Slika 2.1 pokazuje kako se do stanja j u trenutku t može doći iz N prethodnih stanja $j_1, j_2, ..., j_N$. Po definiciji *forward* varijable slijedi da je

$$\alpha_{t-1}(i)\, p_{ij} \tag{2.5}$$

vjerojatnost da je model generirao prvih $t-1$ simbola te da je, nakon posjeta stanja i u trenutku $t-1$, lanac prešao u stanje j. Sumiranje (2.5) po svim mogućim prethodnim stanjima daje vjerojatnost da je lanac u trenutku t u stanju j uz generiran niz prvih $t-1$ simbola:

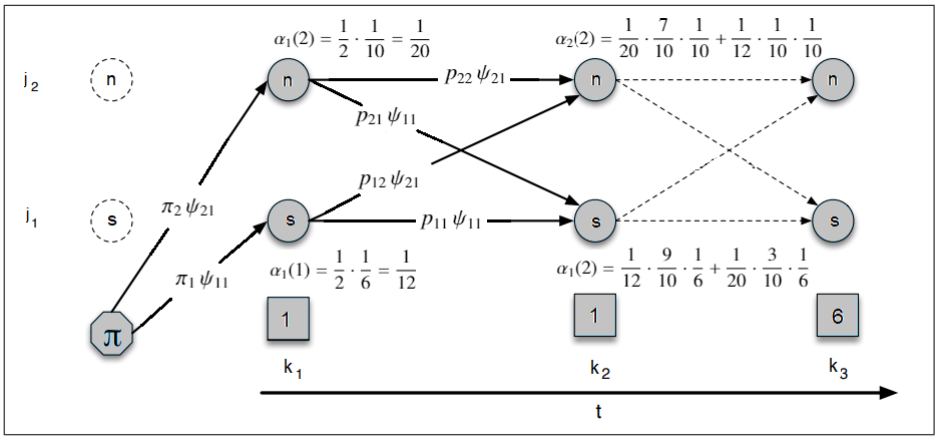
$$\sum_{i \in S} \alpha_{t-1}(i)\, p_{ij}.$$

Nadalje, kako bi se dobila tražena vrijednost (2.4), potrebno je još gornju sumu pomnožiti s vjerojatnosti da je skriveno stanje j emitiralo sljedeći po redu simbol, tj. k_t. Drugi korak algoritma provodi se za sva stanja iz S, sve do zadnjeg trenutka T. Konačno, treći korak algoritma daje rješenje problema evaluacije jer po definiciji $\alpha_t(j)$ slijedi

$$\alpha_T(j) = \mathbb{P}(Y_1 = k_1, Y_2 = k_2, ..., Y_T = k_T, X_T = j),$$

pa se sumiranjem po svim stanjima $j \in S$ u kojima X_T može biti dobije (2.1). Upravo opisani *forward* algoritam je složenosti reda TN^2, puno manje nego direktno računanje.

Primjer 2.1.6. *Na slici je prikazan račun pomoću forward algoritma za model s kockama iz primjera 2.1.2 i problem evaluacije da je model emitirao niz brojeva 1, 1, 6.*



Slika 2.2: Ilustracija niza operacija (varijabli) potrebnih za računanje $\alpha_t(j)$ za primjer s kockama. Slika je preuzeta iz Jurafsky i Martin [3].

Napomena 2.1.7. *Rješenje evaluacijskog problema nudi i backward algoritam koji će biti korišten u rješenju problema učenja pa ga je korisno uvesti.*

Definicija 2.1.8. *Neka su dana opažanja $k_1, k_2, ..., k_T \in O$ i neka je*

$$\beta_t(i) := \mathbb{P}(Y_{t+1} = k_{t+1}, Y_{t+2} = k_{t+2}, ..., Y_T = k_T \,|\, X_t = i), \text{ za svaki } t \leq T,\ i \in S.$$

Ovako definirane parametre uobičajeno nazivamo backward varijablama. $\beta_t(i)$ predstavlja vjerojatnost da je model emitirao zadnjih $T-t$ elemenata niza opažanja uz uvjet da se u trenutku t skriveni lanac nalazi u stanju i.

Propozicija 2.1.9. *Za backward varijable vrijedi rekurzija:*

$$\beta_{t-1}(i) = \sum_{j\in S} \beta_t(j)\, p_{ij}\, \psi_{jk_t}\,, \ \ \textit{za svaki } i \in S, 1 \leq t < T.$$

Dokaz.

$$\begin{aligned}
\beta_{t-1}(i) &= \mathbb{P}(Y_t = k_t, Y_{t+1} = k_{t+1}, ..., Y_T = k_T \,|\, X_{t-1} = i) \\
&= \sum_{j\in S} \mathbb{P}(Y_t = k_t, Y_{t+1} = k_{t+1}, ..., Y_T = k_T, X_t = j \,|\, X_{t-1} = i) \\
&= \sum_{j\in S} \mathbb{P}(Y_t = k_t, Y_{t+1} = k_{t+1}, ..., Y_T = k_T, X_t = j, X_{t-1} = i) \cdot \frac{1}{\mathbb{P}(X_{t-1} = i)} \\
&= \sum_{j\in S} \mathbb{P}(Y_t = k_t \,|\, Y_{t+1} = k_{t+1}, ..., Y_T = k_T, X_t = j, X_{t-1} = i) \\
&\qquad \cdot \mathbb{P}(Y_{t+1} = k_{t+1}, ..., Y_T = k_T, X_t = j \,|\, X_{t-1} = i) \\
&= \sum_{j\in S} \mathbb{P}(Y_t = k_t \,|\, X_t = j)\, \mathbb{P}(Y_{t+1} = k_{t+1}, ..., Y_T = k_T, \,|\, X_t = j, X_{t-1} = i)\, \mathbb{P}(X_t = j \,|\, X_{t-1} = i) \\
&= \sum_{j\in S} \mathbb{P}(Y_t = k_t \,|\, X_t = j)\, \mathbb{P}(Y_{t+1} = k_{t+1}, ..., Y_T = k_T, \,|\, X_t = j)\, \mathbb{P}(X_t = j \,|\, X_{t-1} = i) \\
&= \sum_{j\in S} \psi_{jk_t} \beta_t(j)\, p_{ij}.
\end{aligned}$$

□

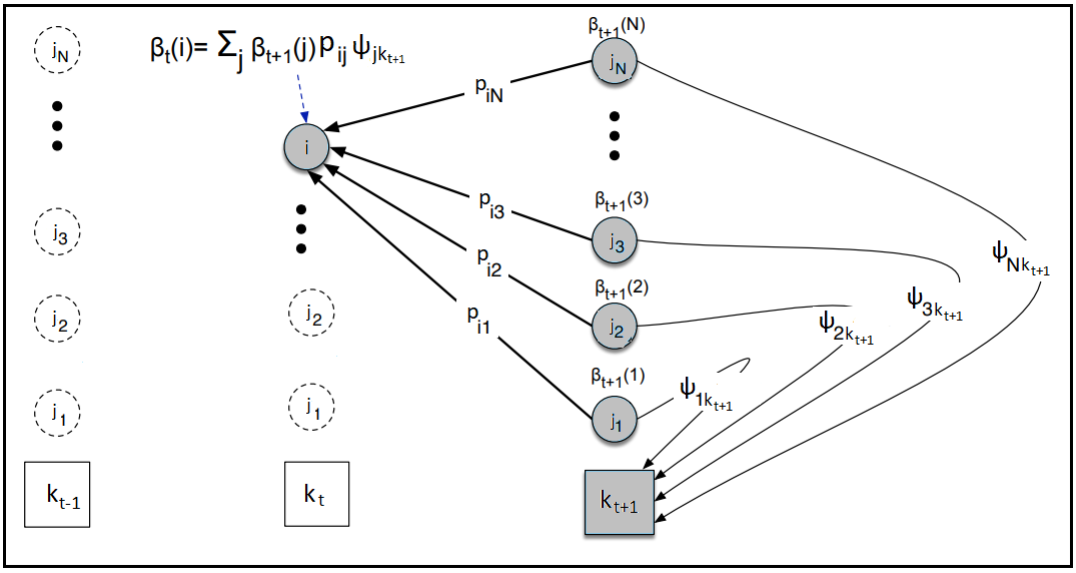
Backward algoritam može se opisati sljedećim koracima:

1. Inicijalizacija:
$$\beta_T(i) = 1, \ \ i \in S$$

2. Rekurzija:
$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j)\, p_{ij}\, \psi_{jk_{t+1}}, \ \ i \in S, 1 \leq t < T$$

3. Kraj:
$$\mathbb{P}(Y_1 = k_1, Y_2 = k_2, ..., Y_T = k_T) = \sum_{j=1}^{N} \pi_j \beta_1(j)\, \psi_{jk_1}.$$

Sam postupak je sličan onome za *forward* algoritam uz glavnu razliku da *backward* algoritam, kako mu ime kaže, ”računa unatrag”. Po definiciji *backward* varijabli, slijedi da je $\beta_{t+1}(j)\, p_{ij}\, \psi_{jk_{t+1}}$ vjerojatnost da je model generirao zadnjih $T-(t+1)$ vrijednosti, te u trenutku t prešao iz skrivenog stanja i u j pošto je emitirao simbol k_{t+1}. Sumiranjem po svim stanjima $j \in S$ dobije se tražena $\beta_t(i)$.



Slika 2.3: Ilustracija niza operacija potrebnih za računanje $\beta_t(i)$. Slika je preuzeta iz Jurafsky i Martin [3].

Kao i kod *forward* algoritma, složenost *backward* algoritma je $O(TN^2)$.

2.2 Problem dekodiranja

Cilj dekodiranja je otkriti skriveni dio modela koji najbolje opisuje opaženi niz emitiranih simbola. Preciznije, uz dani model potrebno je naći ”optimalan” niz skrivenih stanja koji je emitirao dani slijed opažanja tj. onaj koji na neki smisleni način najbolje objašnjava niz opservacijskih simbola. Za razliku od problema evaluacije za koji postoji jedinstveno rješenje, rješenja problema dekodiranja može biti više ovisno o kriteriju optimalnosti.

Rješenje problema dekodiranja

Jedan način odabira optimalnog niza može biti taj da se bira stanje po stanje lanca, ono koje je individualno najvjerojatnije uz zadani niz $k_1, k_2, ..., k_T \in O$. Neka je

$$\gamma_t(j) := \mathbb{P}(X_t = j \,|\, Y_1 = k_1, Y_2 = k_2, ..., Y_T = k_T), \text{za svaki } t \in \{1, 2, ..., T\},\ j \in S. \tag{2.6}$$

Riječima, $\gamma_t(j)$ je vjerojatnost da je u trenutku t lanac u stanju j ako znamo da je emitiran niz opservacija $k_1, k_2, ..., k_T$.

Napomena 2.2.1. *$\gamma_t(j)$ može se izračunati pomoću već poznate forward i backward varijable.*

$$\begin{aligned}
\gamma_t(j) = \mathbb{P}(X_t = j \,|\, Y_1 = k_1, ..., Y_t = k_t, ..., Y_T = k_T) &= \frac{\mathbb{P}(X_t = j, Y_1 = k_1, ..., Y_t = k_t, ..., Y_T = k_T)}{\mathbb{P}(Y_1 = k_1, ..., Y_t = k_t, ..., Y_T = k_T)} \\
&= \frac{\mathbb{P}(X_t = j, Y_1 = k_1, ..., Y_t = k_t, Y_{t+1} = k_{t+1}, ..., Y_T = k_T)}{\sum_{j \in S} \mathbb{P}(Y_1 = k_1, ..., Y_t = k_t, Y_{t+1} = k_{t+1}, ..., Y_T = k_T, X_t = j)} \\
&= \frac{\mathbb{P}(Y_1 = k_1, ..., Y_t = k_t, X_t = j)\,\mathbb{P}(Y_{t+1} = k_{t+1}, ..., Y_T = k_T \,|\, Y_1 = k_1, ..., Y_t = k_t, X_t = j)}{\sum_{j \in S} \mathbb{P}(Y_1 = k_1, ..., Y_t = k_t, X_t = j)\,\mathbb{P}(Y_{t+1} = k_{t+1}, ..., Y_T = k_T \,|\, Y_1 = k_1, ..., Y_t = k_t, X_t = j)} \\
&= \frac{\mathbb{P}(Y_1 = k_1, ..., Y_t = k_t, X_t = j)\,\mathbb{P}(Y_{t+1} = k_{t+1}, ..., Y_T = k_T \,|\, X_t = j)}{\sum_{j \in S} \mathbb{P}(Y_1 = k_1, ..., Y_t = k_t \,|\, X_t = j)\,\mathbb{P}(Y_{t+1} = k_{t+1}, ..., Y_T = k_T, X_t = j)} \\
&= \frac{\alpha_t(j)\beta_t(j)}{\sum_{j \in S} \alpha_t(j)\beta_t(j)}.
\end{aligned}$$

Optimalan niz skrivenih stanja $j_1, j_2, ..., j_T$ dobije se pomoću iterativne formule

$$j_t = \arg\max_{j \in S} \gamma_t(j), \quad t \in \{1, 2, ..., T\},$$

gdje je stanje j_t odabrano kao ono za koje je γ_t najveće, tj. ono u kojem je lanac najvjerojatnije bio u trenutku t s obzirom na dani niz simbola.

Iako na prvu razuman odabir stanja, ovaj pristup može dovesti do nemogućeg scenarija u slučaju prijelaznih vjerojatnosti jednakih 0. Budući da se ovim postupkom gleda samo jedno stanje u danom trenutku, ne provjerava se povezanost susjednih stanja. Postoje načini koji bi zaobišli ovaj problem za određene primjene, ali najčešće se ipak gleda drugačiji kriterij optimalnosti.

Problemom dekodiranja uobičajeno se smatra pronalazak niza $j_1, j_2, ..., j_T \in S$ koji maksimizira vjerojatnost

$$\mathbb{P}(X_1 = j_1, X_2 = j_2, ..., X_T = j_T \,|\, Y_1 = k_1, Y_2 = k_2, ..., Y_T = k_T), \tag{2.7}$$

uz određen slijed opažanja $k_1, k_2, ..., k_T$.

Napomena 2.2.2. *Maksimiziranje vjerojatnosti (2.7) ekvivalentno je maksimiziranju*

$$\mathbb{P}(X_1 = j_1, X_2 = j_2, ..., X_T = j_T, Y_1 = k_1, Y_2 = k_2, ..., Y_T = k_T), \tag{2.8}$$

što slijedi iz definicije uvjetne vjerojatnosti.

Algoritam za problem dekodiranja

Algoritam za pronalazak jedinstvenog optimalnog niza latentnih stanja postoji i zove se Viterbijev algoritam. Za potrebe algoritma, potrebno je uvesti sljedeću oznaku uz dane $k_1, k_2, ..., k_t \in O$:

$$\delta_t(i) := \max_{j_1, j_2, ..., j_{t-1} \in S} \mathbb{P}(X_1 = j_1, X_2 = j_2, ..., X_t = i, Y_1 = k_1, Y_2 = k_2, ..., Y_t = k_t).$$

Dakle, $\delta_t(i)$ je najveća vjerojatnost koju neki niz stanja duljine t može postići tako da završi u stanju i i emitira prvih t zadanih simbola $k_1, k_2, ..., k_t$. Slično kao za *forward* i *backward* varijable, pokaže se rekurzija:

$$\delta_t(j) = [\max_{i \in S} \delta_{t-1}(i)\, p_{ij}] \cdot \psi_{jk_t}. \tag{2.9}$$

Viterbijev algoritam je opisan sljedećim koracima:

1. Inicijalizacija:
$$\delta_1(i) = \pi_i\, \psi_{ik_1}, \quad i \in S$$
$$\zeta_1(i) = 0.$$

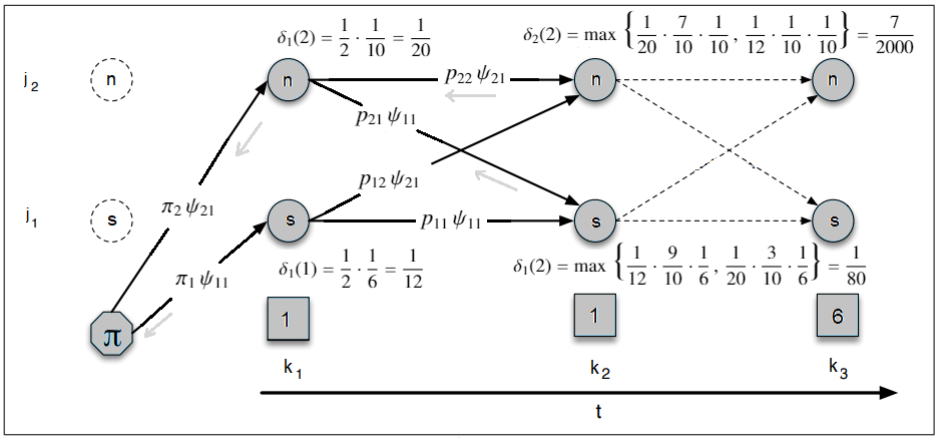
2. Rekurzija:
$$\delta_t(j) = \max_{i \in S} \delta_{t-1}(i)\, p_{ij}\, \psi_{jk_t}, \quad j \in S,\ 2 \leq t \leq T$$
$$\zeta_t(j) = \arg\max_{i \in S} \delta_{t-1}(i)\, p_{ij}, \quad j \in S,\ 2 \leq t \leq T$$

3. Kraj:
$$P* = \max_{i \in S} \delta_T(i),$$
$$j_T* = \arg\max_{i \in S} \delta_T(i).$$

4. Dobivanje traženog niza:
$$j_t* = \zeta_{t+1}(j_{t+1}*), \quad t = T-1, T-2, ..., 1.$$

Po definiciji parametra $\delta_t(i)$, $\max_{i \in S} \delta_T(i)$ daje najveću moguću vjerojatnost (2.8). Cilj dekodiranja je pronaći niz stanja na kojem se ona postiže pa uz rekurzivno računanje vjerojatnosti *Viterbijevim* algoritmom treba pratiti stanja puta, što je u algoritmu zadano s parametrom $\zeta_t(j)$. U 4.koraku provodi se unatražna rekurzija za dohvat stanja.

Primjer 2.2.3. *Za primjer s kockama, pitanje koja kocka je kada najvjerojatnije korištena uz dobivene 1, 1, 6 bilo bi pitanje dekodiranja.*



Slika 2.4: Ilustracija Viterbijevog algoritma za primjer s kockama. Slika je preuzeta iz Jurafsky i Martin [3].

2.3 Problem učenja

Treći problem skrivenih Markovljevih modela je problem optimalne procjene parametara, onih za koje je, za pripadni model, vjerojatnost dobivanja određenog niza opažanja najveća. Rješenje prvog problema daje vjerojatnost da ga je generirao početni model. Rješenje drugog problema daje ”optimalan” niz skrivenih stanja koji emitira te simbole. Oba problema rješavaju se pomoću danih parametara modela. U trećem zadatku potrebno je iz dobivenih podataka i inicijalnog modela procijeniti optimalne verzije prijelazne matrice P, tranzicijske matrice Ψ i početne distribucije Π.

Rješenje problema učenja

Procijenimo parametre na sljedeći način:

$$\hat{\pi}_i = \text{očekivani broj posjeta stanju } i \text{ u trenutku } 1, \tag{2.10}$$

$$\hat{p}_{ij} = \frac{\text{očekivani broj prelazaka iz stanja } i \text{ u } j}{\text{očekivani broj prolazaka kroz stanje } i}, \tag{2.11}$$

$$\hat{\psi}_{jk} = \frac{\text{očekivani broj posjeta stanju } j \text{ i viđanja simbola } k}{\text{očekivani broj posjeta stanju } j}. \tag{2.12}$$

Dakle, potrebno je naći očekivani broj posjeta svakom stanju lanca uz dani niz simbola:

$$\mathbb{E}\left(\sum_{t=1}^{T} 1_{\{X_t=i\}} \,|\, Y_1 = k_1, Y_2 = k_2, ..., Y_T = k_T\right), \; i \in S, \tag{2.13}$$

očekivani broj emitiranja određenog simbola k iz stanja j:

$$\mathbb{E}\left(\sum_{t=1}^{T} 1_{\{X_t=i, Y_t=k\}} \,|\, Y_1 = k_1, Y_2 = k_2, ..., Y_T = k_T\right), \; i \in S, \; k \in O, \tag{2.14}$$

i očekivani broj prelazaka lanca iz jednog stanja u drugo:

$$\mathbb{E}\left(\sum_{t=1}^{T-1} 1_{\{X_t=i, X_{t+1}=j\}} \,|\, Y_1 = k_1, Y_2 = k_2, ..., Y_T = k_T\right), \; i, \; j \in S. \tag{2.15}$$

Napomena 2.3.1. *Zbog linearnosti uvjetnog očekivanja vrijedi:*

$$\mathbb{E}\left(\sum_{t=1}^{T} 1_{\{X_t=i\}} \,|\, Y_1 = k_1, ..., Y_T = k_T\right) = \sum_{t=1}^{T} \mathbb{P}(X_t = i \,|\, Y_1 = k_1, ..., Y_T = k_T);$$

$$\mathbb{E}\left(\sum_{t=1}^{T} 1_{\{X_t=i, Y_t=k\}} \,|\, Y_1 = k_1, ..., Y_T = k_T\right) = \sum_{t=1}^{T} \mathbb{P}(X_t = i, Y_t = k \,|\, Y_1 = k_1, ..., Y_T = k_T);$$

$$\mathbb{E}\left(\sum_{t=1}^{T-1} 1_{\{X_t=i, X_{t+1}=j\}} \,|\, Y_1 = k_1, ..., Y_T = k_T\right) = \sum_{t=1}^{T-1} \mathbb{P}(X_t = i, X_{t+1} = j \,|\, Y_1 = k_1, ..., Y_T = k_T).$$

Za parametar početne distribucije $\hat{\pi}_i$ računamo očekivani broj posjeta stanju i u trenutku 1:

$$\mathbb{E}\left(1_{\{X_1=i\}} \,|\, Y_1 = k_1, ..., Y_T = k_T\right) = \mathbb{P}(X_1 = i \,|\, Y_1 = k_1, ..., Y_T = k_T) = \gamma_1(i).$$

Za izračun očekivanja (2.13) također možemo iskoristiti oznaku $\gamma_t(j)$ uvedenu za rješenje problema dekodiranja s (2.6) i dobiveni rezultat iz napomene 2.2.1:

$$\mathbb{E}\left(\sum_{t=1}^{T} 1_{\{X_t=i\}} \,|\, Y_1 = k_1, ..., Y_T = k_T\right) = \sum_{t=1}^{T} \gamma_t(i) = \sum_{t=1}^{T} \frac{\alpha_t(i)\beta_t(i)}{\sum_{i\in S} \alpha_t(i)\beta_t(i)}.$$

Za očekivani broj prolazaka kroz stanje i potrebno je gornju sumu provesti do trenutka $T-1$ umjesto do T. Uočimo da slično vrijedi za (2.14):

$$\mathbb{E}\left(\sum_{t=1}^{T} 1_{\{X_t=i, Y_t=k\}} \,|\, Y_1 = k_1, ..., Y_T = k_T\right) = \sum_{\substack{t=1 \\ Y_t=k}}^{T} \gamma_t(i).$$

Neka je

$$\xi_t(i, j) := \mathbb{P}(X_t = i, X_{t+1} = j \,|\, Y_1 = k_1, ..., Y_T = k_T).$$

Propozicija 2.3.2. *Za svaki trenutak $t \in \{1, ..., T\}$ i sva stanja $i, j \in S$ vrijedi:*

$$\xi_t(i, j) = \frac{\alpha_t(i)\, p_{ij}\, \psi_{jk_{t+1}}\, \beta_{t+1}(j)}{\sum_{i\in S} \alpha_t(i)\beta_t(i)}.$$

Koristeći uvedeni parametar $\xi_t(i, j)$ očekivanje (2.15) se može izračunati kao:

$$\mathbb{E}\left(\sum_{t=1}^{T-1} 1_{\{X_t=i, X_{t+1}=j\}} \,|\, Y_1 = k_1, Y_2 = k_2, ..., Y_T = k_T\right) = \sum_{t=1}^{T-1} \xi_t(i, j).$$

Iz navedenih računa za potrebna očekivanja slijedi da za (2.10), (2.11) i (2.12) redom vrijedi:

$$\hat{\pi}_i = \gamma_1(i), \quad i \in S,$$

$$\hat{p}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad i, j \in S$$

$$\hat{\psi}_{jk} = \frac{\sum_{\substack{t=1 \\ Y_t=k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}, \quad j \in S, k \in O.$$

Dakle, za izračun procijenjenih parametara potrebni su pomoćni parametri $\gamma_t(j)$ i $\xi_t(i, j)$. Njih računamo preko početnih parametara modela i *forward* i *backward* varijable koje računamo algoritmima opisanim za rješenje evaluacijskog problema.

Algoritam za problem učenja

Problem učenja smatra se i najtežim budući da se ne može direktno doći do parametara modela koji optimiziraju vjerojatnost (2.1). Ono što se ipak može pronaći je model takav da je (2.1) lokalni maksimum koristeći tzv. *Baum-Welschovu* metodu. *Baum-Welchova* metoda je iterativni postupak, usko vezan uz metodu maksimalnog procjenitelja. Problem koji treba maksimizirati je

$$\max_{P,\Psi,\Pi} \mathbb{P}(Y_1 = k_1, ..., Y_T = k_T \,;\, P, \Psi, \Pi).$$

BW postupak spada u algoritme maksimiziranja očekivanja (*Expectation-Maximization algorithm*). Funkcija vjerodostojnosti definira se s

$$L(P, \Psi, \Pi) = \mathbb{P}(Y_1 = k_1, ..., Y_T = k_T, X_1 = j_1, ..., X_T = j_T \,;\, P, \Psi, \Pi),$$

a maksimizira se uvjetno očekivanje log-funkcije vjerodostojnosti:

$$\mathbb{E}\left[log\left(L\left(P, \Psi, \Pi\right)\right) | \, Y_1 = k_1, ..., Y_T = k_T; P^{(1)}, \Psi^{(1)}, \Pi^{(1)}\right].$$

Rješavanjem ovog problema dobiju se procijenjeni parametri $\hat{\pi}$, $\hat{p}$ i $\hat{\psi}$ kao (2.10), (2.11) i (2.12) što opravdava uvedeno rješenje problema učenja.

Za početni model s parametrima P, Ψ, Π na gore opisani način (2.3) procijene se novi parametri $P^{(1)}$, $\Psi^{(1)}$, $\Pi^{(1)}$. Sada uz te nove parametre modela (i samim time novi model) opet se provodi isti postupak te se dobiju parametri $P^{(2)}$, $\Psi^{(2)}$, $\Pi^{(2)}$. Postupak se ponavlja i za svaku iteraciju n vrijedi

$$\mathbb{P}(Y_1 = k_1, ..., Y_T = k_T \,;\, P^{(n)}, \Psi^{(n)}, \Pi^{(n)}) \geq \mathbb{P}(Y_1 = k_1, ..., Y_T = k_T \,;\, P^{(n-1)}, \Psi^{(n-1)}, \Pi^{(n-1)}).$$

Na taj način, vrijednost konvergira u (lokalni) maksimum funkcije vjerodostojnosti.

Poglavlje 3

Primjene

Skriveni Markovljevi modeli koriste se u raznim područjima. Model se pokazao izrazito koristan u prepoznavanju govora i uzoraka te analizi jezika. Poznate su primjene i u biologiji; za analiziranje DNK, predviđanje genetskog sadržaja, itd. U financijama se može koristiti za analizu i predviđanje tržišta dionica. Ostala područja gdje je poznato korištenje skriv.Mar.modela uključuju i termodinamiku, fiziku i kemiju [13].

U ovom poglavlju provest će se nekoliko simulacija skriv.Mar.modela te riješiti neki od osnovnih problema koristeći R i programski paket "HMM". Ovaj paket je prikladan za obrađeni model s diskretnim stanjima i opažanjima.

3.1 Nepošteni kasino

Primjer 1.2.3 u literaturi se može naći pod nazivom *Dishonest Casino*. U primjeru imamo dvije kocke, jednu savršeno simetričnu, a drugu s naklonosti prema šestici. Prijelazna matrica definirana je s

$$\begin{pmatrix} 0.99 & 0.01 \\ 0.02 & 0.98 \end{pmatrix},$$

a početna distribucija i emisijska matrica kao u primjeru. Funkcija *initHMM* inicijalizira model uz dane parametre. Ako početna distribucija nije zadana pridružuje modelu uniformnu početnu distribuciju. Ispod koda je ispisan inicijalizirani model. Uz zadani model, funkcija *simHMM* simulira niz (duljine 1000 u ovom slučaju) korištenih kocaka te brojeva koji su pali. Ispod modela je prikazano prvih 10. Slijedi kod i output:

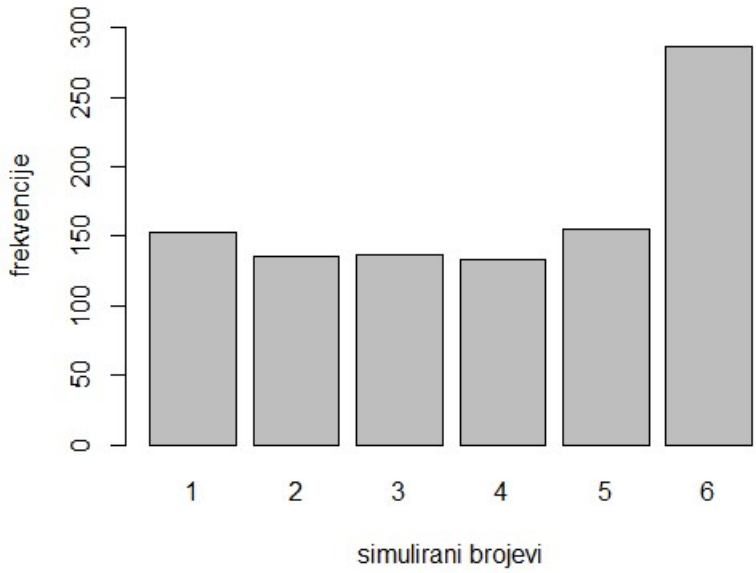
```
brojSim = 1000
Stanja = c("s", "n")
Simboli = 1:6
```

```
pMatrica = matrix(c(0.99, 0.01, 0.02, 0.98), ncol=2,
           byrow=TRUE)
eMatrica = matrix(c(rep(1/6, 6), c(rep(0.1, 5), 0.5)), ncol=6,
           byrow=TRUE)
hmm = initHMM(Stanja, Simboli, transProbs = pMatrica,
      emissionProbs = eMatrica)
hSim = simHMM(hmm, brojSim)
```

```
---------------------------------------------------------------------

$States
[1] "s" "n"

$Symbols
[1] 1 2 3 4 5 6

$startProbs
  s   n
0.5 0.5

$transProbs
    to
from    s    n
   s 0.99 0.01
   n 0.02 0.98

$emissionProbs
      symbols
states         1         2         3         4         5         6
     s 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667
     n 0.1000000 0.1000000 0.1000000 0.1000000 0.1000000 0.5000000

hSim$states[1:10]
 [1] "n" "n" "n" "n" "n" "n" "n" "n" "n" "n"

hSim$observation[1:10]
 [1] 6 4 6 6 5 3 6 5 6 5
```

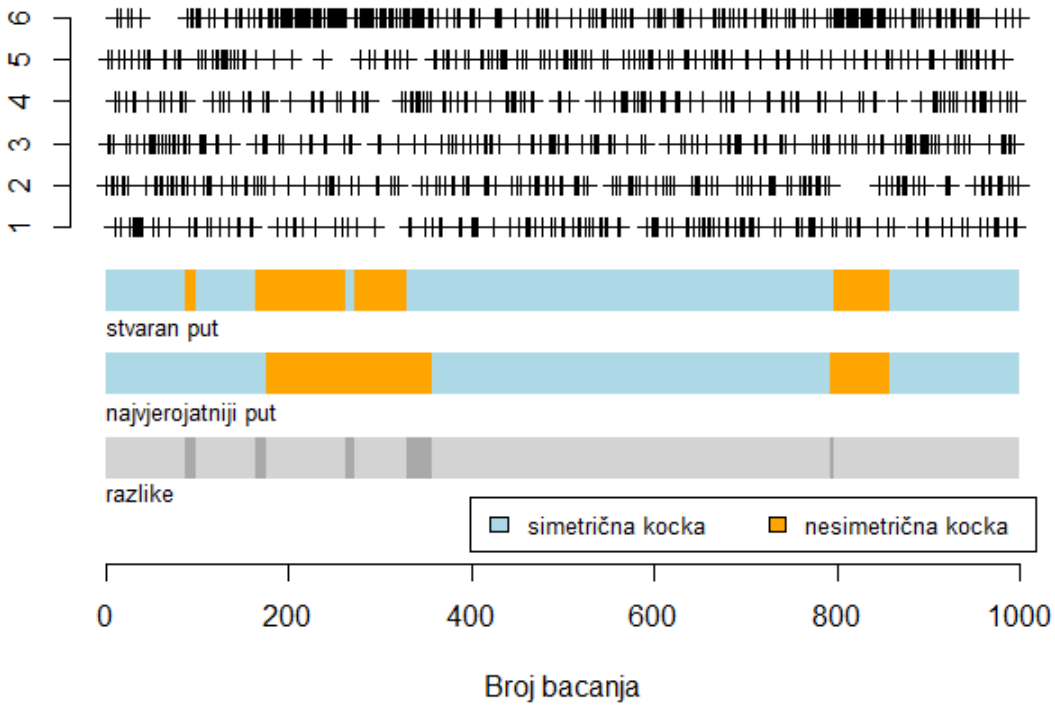
Pomoću histograma se može vidjeti velik udio broja 6 u uzorku simuliranih brojeva što se i očekuje s obzirom na emisijske vjerojatnosti u slučaju nesimetrične kocke.

Slika 3.1: Histogram simuliranih brojeva



Pomoću funkcija *forward*, *backward*, *posterior* dobiju se redom $\alpha_t(j), \beta_t(j), \gamma_t(j)$ (definirani u poglavlju 2) za svaki trenutak $t \in \{1, 2, ..., 1000\}$ i svako skriveno stanje $j \in \{s, n\}$. Funkcija *viterbi* provodi Viterbijev algoritam za dani model i slijed brojeva. Pomoću njega, pronađen je najvjerojatniji put tj. niz skrivenih stanja.

Slika 3.2: Grafički prikaz dobivenih podataka



Na slici 3.2 su crticama obilježeni brojevi dobiveni u svakom bacanju. Bojama su označene korištene kocke. Prikazan je stvarni niz bačenih kocaka koji je emitirao opažanja, a ispod njega onaj dobiven Viterbijevim algoritmom. Zadnje su prikazane razlike, tj. trenuci u kojima se stvarni i najvjerojatniji put ne poklapaju. Vidi se da je pomoću algortima dobivena prilično dobra procjena skrivenog puta.

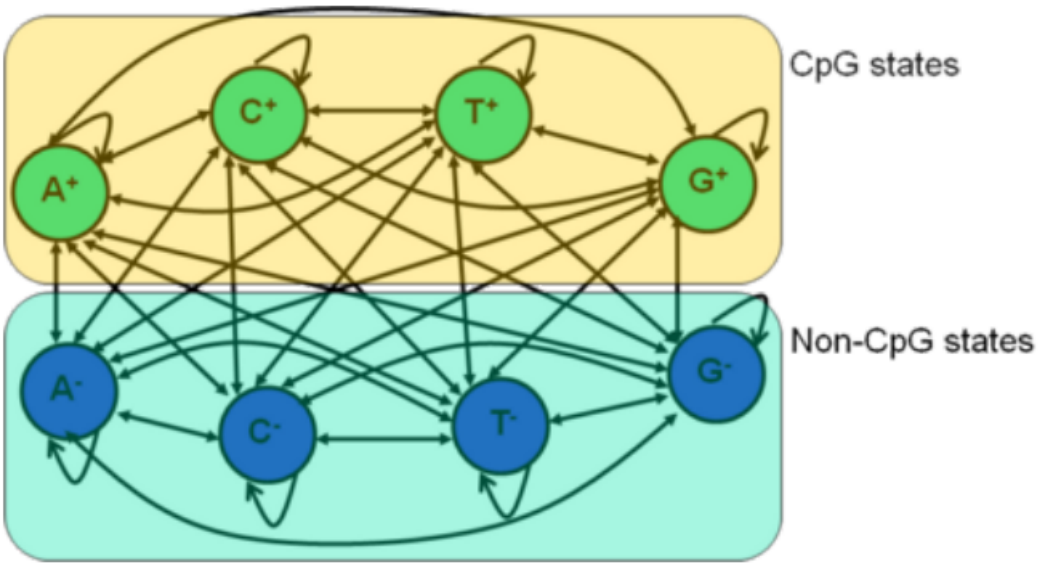
3.2 Bioinformatika

U ovom bioinformatičkom primjeru promatrat će se problem lociranja određenog mjesta u DNK lancu. DNK lanac ima četiri različite nukleotidne baze: adenin (A), timin (T), citozin (C) i gvanin (G). CpG je skraćena oznaka za ”—C—fosfat—G—”, tj. mjesta gdje su citozin i gvanin susjedni nukleotidi razdvojeni jednim fosfatom. Učestalost CpG dinukleotida je vrlo niska u ljudskom genomu, javljaju se čak 5-10 puta rjeđe od ostalih dinukleotida.

Iznimka od relativno rijetke distribucije CpG mjesta u genomu su tzv. CpG otoci, genomska područja izuzetno bogata ovim dinukleotidima. CpG otoci definirani su kao odsječci DNA dulji od 200 parova baza gdje suma gvanina i citozina iznosi više od 55% ukupnog broja baza (Dobrinić [12]). Ova područja se često pojavljuju u biološki značajnijim dijelovima genoma (mjestima gdje se geni prepisuju) pa je od velike važnosti identificirati ih iz slijeda nukleotidnih baza.

Model možemo zadati pomoću osam skrivenih stanja $A+, A-, C+, C-, T+, T-, G+, G-$ gdje + predstavlja područje CpG otoka, a − područje izvan. Svako od stanja s vjerojatnosti 1 generira pripadni simbol, tj. pripadnu nukleotidnu bazu (npr. $A+$ i $A-$ sigurno emitiraju adenin). Na ovaj način možemo iskoristiti činjenicu da su vjerojatnosti pojavljivanja nukleotidnih baza bitno drugačije u CpG otoku od ostatka DNK. Cilj je pomoću niza nukleotidnih baza A, C, T, G vidjeti u kojem se području najvjerojatnije nalazimo. Dakle, model zadajemo sa skupom stanja $S = \{A+, A-, C+, C-, T+, T-, G+, G-\}$ i skupom simbola $O = \{A, C, G, T\}$.

Slika 3.3: Skrivena stanja modela. Slika je preuzeta iz Macauley [10]



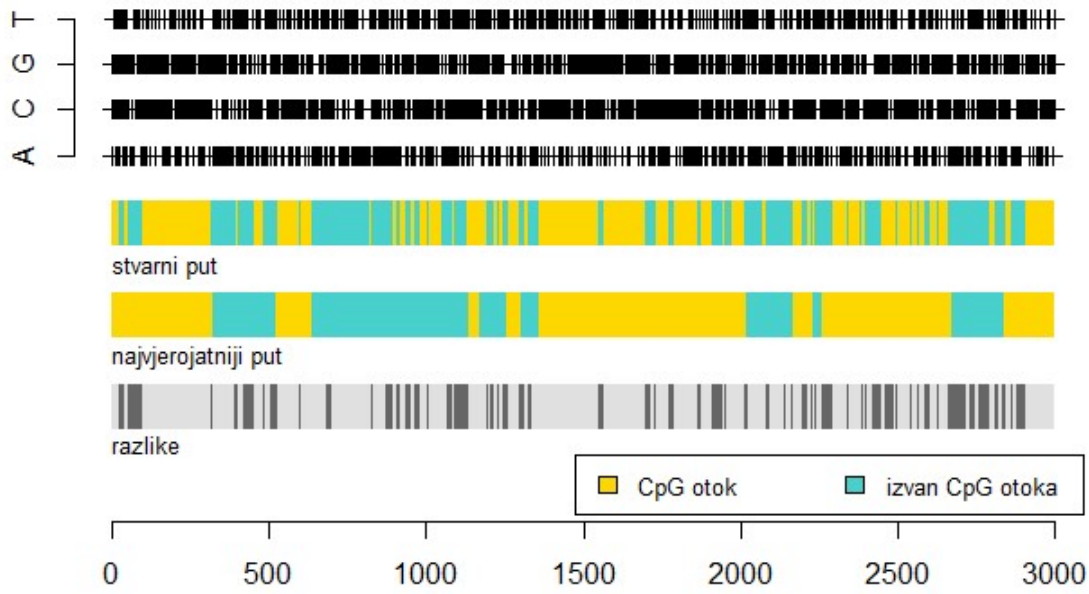
Prijelazna matrica i početna distribucija preuzete su iz Shamir [11]. Prijelazne vjerojatnosti procijenjene su iz uzorka DNK duljine $\approx$ 60,000. Početni parametri bili su procijenjeni na način da se iz uzorka (od 60,000 nukleotida) tražila frekvencija dinukleotida. Npr.

$$p_{A+C+} = \frac{\text{broj pojave ApC dinukleotida u CpG otoku}}{\text{broj svih dinukleotida}}.$$

Kod za inicijalizaciju modela, simulaciju niza nukleotida te optimalan put:

```
brojSim = 3000
Stanja = c("A+","C+","G+","T+", "A-","C-","G-","T-")
Simboli = c("A", "C", "G", "T")
pMatrica = matrix(c(0.17,0.26,0.42,0.11,0.01,0.01,0.01,0.01,
                    0.16,0.36,0.26,0.18,0.01,0.01,0.01,0.01,
                    0.15,0.33,0.37,0.11,0.01,0.01,0.01,0.01,
                    0.07,0.35,0.37,0.17,0.01,0.01,0.01,0.01,
                    0.01,0.01,0.01,0.01,0.29,0.2,0.27,0.2,
                    0.01,0.01,0.01,0.01,0.31,0.29,0.07,0.29,
                    0.01,0.01,0.01,0.01,0.24,0.23,0.29,0.2,
                    0.01,0.01,0.01,0.01,0.17,0.23,0.28,0.28),
                    ncol=8, byrow = TRUE)
eMatrica = matrix(c(1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,1,0,0,0,
           0,1,0,0,0,0,1,0,0,0,0,1),ncol=4,byrow=TRUE)
hmm = initHMM(Stanja, Simboli, transProbs = pMatrica,
      emissionProbs = eMatrica)
hsimCPG = simHMM(hmm, brojSim)
vitCPG = viterbi(hmm, hsimCPG$observation)
```

Slika 3.4: Grafički prikaz podataka



Provodena je i Baum-Welchova metoda za procjenu parametara pomoću funkcije *baumWelch*. Ona vraća novi model s procijenjenim parametrima i vektor razlika izračunat iz uzastopnih matrica prijelaza i emisije u svakoj iteraciji BW postupka. Razlika je zbroj udaljenosti između uzastopnih matrica prijelaza i emisije. Ove razlike se sa svakom iteracijom smanjuju te upućuju na konvergenciju k maksimumu funkcije vjerodostojnosti. Kod i output:

```
bwCPG=baumWelch(hmm,hsimCPG$observation,maxIterations=30)

-------------------------------------------------------------------

$hmm
$hmm$States
[1] "A+" "C+" "G+" "T+" "A-" "C-" "G-" "T-"

$hmm$Symbols
[1] "A" "C" "G" "T"

$hmm$startProbs
   A+    C+    G+    T+    A-    C-    G-    T-
0.125 0.125 0.125 0.125 0.125 0.125 0.125 0.125

$hmm$transProbs
    to
from           A+           C+          G+          T+
  A+ 0.1366652555 0.3186877924 0.364092623 0.146569332
  C+ 0.1629847928 0.3740277301 0.241277486 0.192778558
  G+ 0.1145039509 0.3655805108 0.351829247 0.125086662
  T+ 0.0821105898 0.3093610933 0.347587047 0.206051075
  A- 0.0234181650 0.0089256421 0.001937136 0.012265608
  C- 0.0009580621 0.0371624780 0.012900523 0.001501802
  G- 0.0041946060 0.0066474126 0.052629005 0.005754638
  T- 0.0707711023 0.0006850653 0.029850243 0.097011715
    to
from          A-           C-          G-          T-
  A+ 0.004754021 0.0009773339 0.019228040 0.009025602
  C+ 0.005356103 0.0128709195 0.008387177 0.002317235
  G+ 0.004456107 0.0077355794 0.014662440 0.016145503
  T+ 0.047369997 0.0024581107 0.001506378 0.003555709
  A- 0.281843991 0.1854442704 0.307387915 0.178777273
  C- 0.426465639 0.1868351865 0.021553529 0.312622781
```

```
  G- 0.369191806 0.0991091528 0.299463704 0.163009675
  T- 0.088975117 0.2727177581 0.215547216 0.224441784

$hmm$emissionProbs
      symbols
states A C G T
    A+ 1 0 0 0
    C+ 0 1 0 0
    G+ 0 0 1 0
    T+ 0 0 0 1
    A- 1 0 0 0
    C- 0 1 0 0
    G- 0 0 1 0
    T- 0 0 0 1


$difference
 [1] 0.080987407 0.034095391 0.025794835 0.020252904
 [5] 0.016428900 0.013949658 0.012567242 0.012006131
 [9] 0.011973533 0.012220075 0.012569221 0.012908561
[13] 0.013170008 0.013314711 0.013324935 0.013200143
[17] 0.012954160 0.012611545 0.012202829 0.011759381
[21] 0.011309017 0.010873185 0.010465967 0.010094596
[25] 0.009760893 0.009463036 0.009197176 0.008958677
[29] 0.008742909 0.008545677
```

3.3 Analiza teksta

U ovom poglavlju bavit ćemo se pitanjem raspoznavanja vrste teksta. Proučavamo dokument koji se sastoji od programskih zadataka i njihovih rješenja. Rješenja zadataka su napisana u programskom jeziku C. Konkretno, opažanje čine četiri zadatka s drugog kolokvija 2019. godine iz kolegija Programiranje 2 [14] zajedno s rješenjima koja su napisana nakon pripadnog zadatka. Opažanje se sastoji od 12238 znakova bez praznina. Cilj je dobiti model koji može prepoznati kada se radi o zadatku tj. običnom tekstu, a kada o kodu. Skup stanja je zato zadan s $S = \{tekst, kod\}$, a skup simbola čine sva slova (osim č,ć,š,ž i đ koja su pretvorena u redom c,c,s,z i d), pravopisni znakovi, dodatni znakovi te znamenke. Dakle, $O = \{a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z,$!, (,), <, > , {, , , =, ; , %, :, +, }, −, |, ., \\, /, &, [,], ∗, _, ′ , \, #, \n, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9}.

Početna distribucija je $\Pi = (0.55, 0.45)$. Prijelaznu matricu zadajemo s:

$$P = \begin{pmatrix} 0.8 & 0.2 \\ 0.1 & 0.9 \end{pmatrix},$$

a emisijske vjerojatnosti jednoliko:

$$\psi_{jk} = \frac{1}{|O|}, \text{ za svaki } k \in O, j \in S.$$

Na ovaj način definiramo model, a zatim funkcijom *baumWelch* procjenjujemo parametre. Ispod koda su ispisane prijelazne matrice i transponirane emisijske matrice (na desnoj strani su prikazane nove procijenjene matrice, a s lijeve strane početne za usporedbu). Vjerojatnosti su zaokružene na 7 decimala radi jasnijeg prikaza. Početna distribucija je ostala nepromijenjena.

```
data=read.delim("zadaci\_kodovi.txt",header=FALSE, quote="",sep="")
tekst=paste(data)
uzorak_lista=tokenize_characters(tekst,lowercase=TRUE,
             strip_non_alphanum=FALSE)
uzorak=uzorak_lista[[1]]
n=length(uzorak)
simboli=read.table("simboli.txt", header=FALSE, sep=" ")
m=length(simboli)
stanja=c("tekst","kod")
pi=c(0.55,0.45)
p=matrix(c(0.8, 0.2, 0.1, 0.9), byrow=TRUE, c(2,2))
e=matrix(numeric(m*2),2,m)
e[1,1:m]=1/m
e[2,1:m]=1/m
hmm_text= initHMM(stanja,simboli,transProbs=p,emissionProbs=e,
          startProbs=pi)
bw_text=baumWelch(hmm_text,uzorak,maxIterations=50)
hmm_procj=bw_text$hmm
log_alfe1=forward(hmm_text,uzorak)
logP1= sum(log_alfe1[,n])
log_alfe2=forward(hmm_procj,uzorak)
logP2 = sum(log_alfe2[,n])
```

---

	tekst	kod	tekst	kod
tekst	0.8	0.2	0.9941685	0.0058314
kod	0.1	0.9	0.0041901	0.9958098

	tekst	kod	tekst	kod
a	0.015873	0.015873	0.1031025	0.0326475
b	0.015873	0.015873	0.0290208	0.0281441
c	0.015873	0.015873	0.0240504	0.0274212
d	0.015873	0.015873	0.0365898	0.0203867
e	0.015873	0.015873	0.0796124	0.0486694
f	0.015873	0.015873	0.0039583	0.0292426
g	0.015873	0.015873	0.0094086	0.0054011
h	0.015873	0.015873	0.0032543	0.0086903
i	0.015873	0.015873	0.1008512	0.0871499
j	0.015873	0.015873	0.0412018	0.0169614
k	0.015873	0.015873	0.0455416	0.0088967
l	0.015873	0.015873	0.0260027	0.0345152
m	0.015873	0.015873	0.0268419	0.0186551
n	0.015873	0.015873	0.050354	0.0446249
o	0.015873	0.015873	0.0729349	0.0389615
p	0.015873	0.015873	0.0344871	0.0343371
q	0.015873	0.015873	0.0007635	0.0001449
r	0.015873	0.015873	0.0601224	0.0557397
s	0.015873	0.015873	0.0434405	0.0353772
t	0.015873	0.015873	0.0667139	0.0549738
u	0.015873	0.015873	0.0454263	0.0236495
v	0.015873	0.015873	0.0249648	0.0082301
w	0.015873	0.015873	0	0.0012816
x	0.015873	0.015873	0	0.0052688
y	0.015873	0.015873	0	0.0009968
z	0.015873	0.015873	0.0249504	0.0146487
,	0.015873	0.015873	0.0123731	0.0231354
.	0.015873	0.015873	0.0110326	0.0064734
!	0.015873	0.015873	0	0.0018512
(	0.015873	0.015873	0.0080321	0.0253626
)	0.015873	0.015873	0.0080318	0.0255052
<	0.015873	0.015873	0	0.0039872
>	0.015873	0.015873	0	0.0102528
{	0.015873	0.015873	0	0.0075472

=	0.015873	0.015873	0	0.026344
;	0.015873	0.015873	0	0.0383056
%	0.015873	0.015873	0	0.0052688
:	0.015873	0.015873	0.0018271	0.0019183
+	0.015873	0.015873	0	0.0065504
}	0.015873	0.015873	0	0.0075472
-	0.015873	0.015873	0.0015597	0.0106608
|	0.015873	0.015873	0	0.000712
/	0.015873	0.015873	0	0.0058384
\	0.015873	0.015873	0	0.004272
&	0.015873	0.015873	0	0.0022784
[	0.015873	0.015873	0	0.0082592
]	0.015873	0.015873	0	0.0082592
*	0.015873	0.015873	0	0.0148096
_	0.015873	0.015873	0.0006702	0.0234255
'	0.015873	0.015873	0	0.0019936
"	0.015873	0.015873	0	0.0111072
0	0.015873	0.015873	0.0007733	0.0083969
1	0.015873	0.015873	0.0003907	0.007969
2	0.015873	0.015873	0	0.0068352
3	0.015873	0.015873	0.0017152	0.0044221
4	0.015873	0.015873	0	0.0005696
5	0.015873	0.015873	0	0.0022784
6	0.015873	0.015873	0	0.0008544
7	0.015873	0.015873	0	0.0001424
8	0.015873	0.015873	0	0.0004272
9	0.015873	0.015873	0	0.0002848
\n	0.015873	0.015873	0	0
#	0.015873	0.015873	0	0.0011392

Vrijednost funkcije log-vjerodostojnosti za početni model dobiven pomoću *forward* algoritma je:

$$\log\,[\,\mathbb{P}(Y_1 = k_1, ..., Y_{12238} = k_{12238}\,;\, P, \Psi, \Pi)\,] = -101408.9.$$

Vrijednost funkcije log-vjerodostojnosti nakon 50 iteracija Baum-Welchovog postupka je:

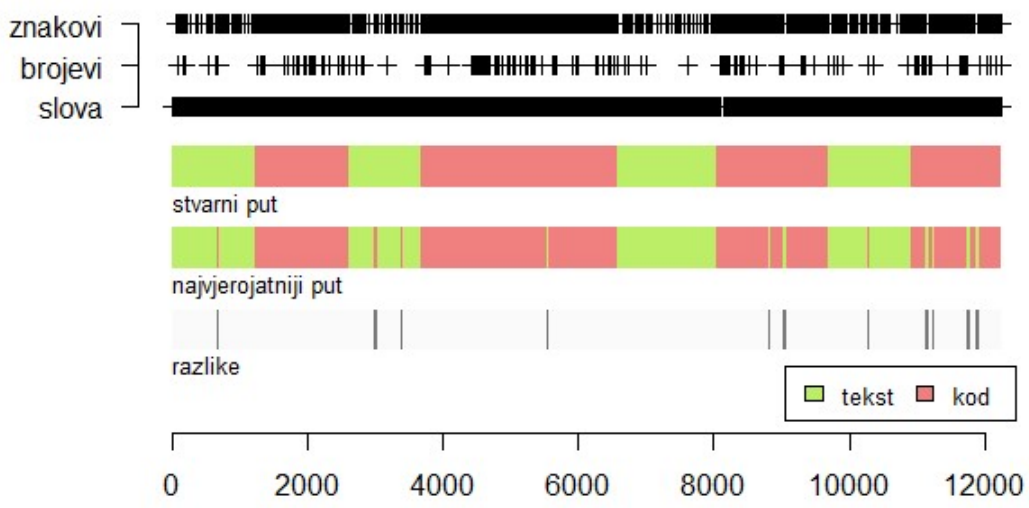
$$\log\,[\,\mathbb{P}(Y_1 = k_1, ..., Y_{12238} = k_{12238}\,;\, \hat{P}, \hat{\Psi}, \hat{\Pi})\,] = -83050.96.$$

Ovo pokazuje da je model poboljšan tj. bolje opisuje dobiven uzorak.

Viterbijevim algoritmom za novi model i dani uzorak tražimo optimalan put:

```
vit_text=viterbi(hmm_procj,uzorak)
```

Slika 3.5: Grafički prikaz podataka

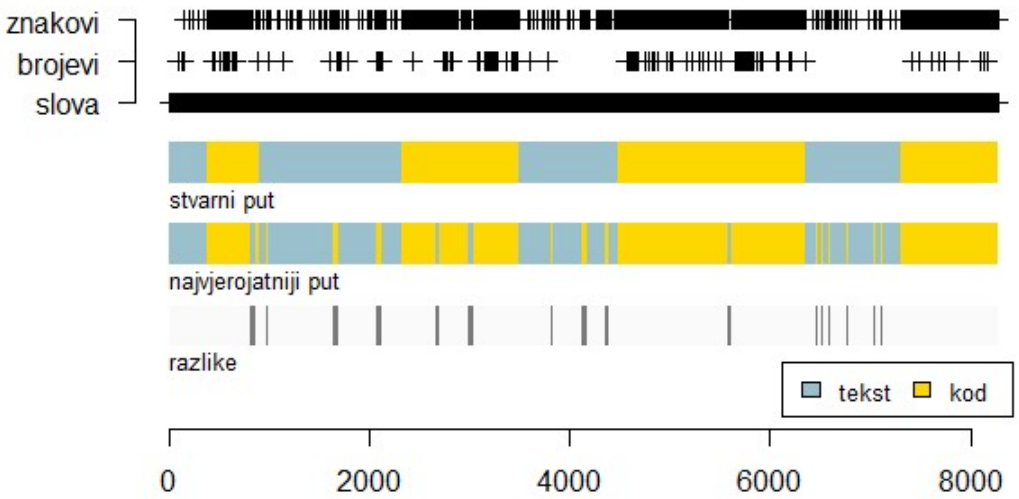


Na slici 3.5 su crticama obilježeni simboli iz uzorka. Grupirani su u tri kategorije (slova, znakovi, brojevi) radi jasnijeg prikaza. Prikazan je stvarni put tj. koji dio uzorka je zadatak (tekst), a koji rješenje (kod). Ispod je prikazan optimalan put dobiven Viterbijevim algoritmom i vidimo da je procjena dobra.

Sada možemo uzeti drugačiji uzorak i vidjeti hoće li ovaj model naučen na prvom uzorku odgovarati novom. Novo opažanje čine četiri zadatka s drugog kolokvija 2020. iz kolegija Programiranje 2 [14] zajedno s rješenjima koja su napisana nakon pripadnog zadatka. Ovaj uzorak sadrži 8272 znakova. Kod:

```
data2=read.delim("zadaci_kodovi2.txt",header=FALSE,quote="",sep="")
tekst2=paste(data2)
uzorak_lista2=tokenize_characters(tekst2,lowercase=TRUE,
              strip_non_alphanum=FALSE)
uzorak2=uzorak_lista2[[1]]
vit_text2=viterbi(hmm_procj,uzorak2)
```

Slika 3.6: Grafički prikaz podataka za novi uzorak



Na slici 3.6 vidimo da je optimalan put dobiven Viterbijevim algoritmom sličan stvarnom putu što ukazuje na to da model uistinu može otprilike odrediti koji dio uzorka je običan tekst, a koji dio kod.

Poglavlje 4

Zaključak

Skriveni Markovljevi modeli nam omogućuju modeliranje procesa sa skrivenim stanjima na temelju određenog uočljivog niza. Pomoću primjera danih u radu (u poglavlju Primjene) pokazano je da su ovi modeli vrijedan alat za prepoznavanje uzoraka. U prvom smo primjeru pomoću niza brojeva od 1 do 6 otkrili koja kocka je najvjerojatnije bila bačena da se dobije takav niz. Drugi primjer je pojednostavljenje primjene modela za određivanje CpG otoka u lancu DNK. Za jasnije i bolje rezultate trebalo bi uvesti još bitnih parametara (dodatnih stanja, dodatnih simbola), te se čak često koristi i poseban oblik skrivenih Markovljevih modela kojeg se u ovom radu nismo dotaknuli. U trećem primjeru pokazana je uspješnost modela u raspoznavanju dva oblika tekstualnog zapisa. Postoje općenitije verzije ovih modela te razni oblici koji su se pokazali djelotvorni u različitim područjima, a u ovom radu smo pokazali kako je i najjednostavniji tip skrivenog Markovljevog modela potencijalno vrlo koristan.

Bibliografija

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

[2] F. Caron, *Lecture notes: Hidden Markov Models*, University of Oxford, 2019., dostupno na linku `http://www.stats.ox.ac.uk/~caron/teaching/sb1b/lecturehmm.pdf` (listopad 2020.)

[3] D. Jurafsky, J.H. Martin, *Speech and Language Processing*, 2019., dostupno na linku `https://web.stanford.edu/~jurafsky/slp3/A.pdf` (listopad 2020.)

[4] R. von Handel, *Hidden Markov Models, Lecture Notes*, 2008., dostupno na linku `https://web.math.princeton.edu/~rvan/orf557/hmm080728.pdf` (listopad 2020.)

[5] C. Kohlschein, *An introduction to Hidden Markov Models*, dostupno na linku `https://www.tcs.rwth-aachen.de/lehre/PRICS/WS2006/kohlschein.pdf` (studeni 2020.)

[6] Z. Vondraček, *Markovljevi lanci, predavanja*, Prirodoslovno-matematički fakultet, Zagreb, 2008.

[7] M. Karuza, *Izvedba funkcija za skrivene Markovljeve modele*, završni rad, Fakultet elektrotehnike i računarstva, Zagreb, 2011.

[8] M. Tepić, *Kompleksnost skrivenih Markovljevih modela*, diplomski rad, Prirodoslovno-matematički fakultet, Zagreb, 2015.

[9] *Hidden Markov Models - the Unfair Casino*, dostupno na linku: `https://web.stanford.edu/class/stats366/hmmR2.html` (studeni 2020.)

[10] M. Macauley, *Identifying CpG islands using hidden Markov models*, Clemson University, 2016., dostupno na linku `http://www.math.clemson.edu/~macaule/classes/f16_math4500/slides/f16_math4500_cpg-islands_handout.pdf` (listopad 2020.)

[11] R. Shamir, *Algorithms for Molecular Biology*, 1999., dostupno na linku `http://www.cs.tau.ac.il/~rshamir/algmb/98/scribe/pdf/lec06.pdf` (listopad 2020.)

[12] P. Dobrinić *Konstrukcija epigenetičkoga CRISPR/Cas9 sustava za ciljanu metilaciju specifičnih CpG mjesta*, doktorski rad, Prirodoslovno-matematički fakultet, Zagreb, 2016.

[13] Wikipedia, *Hidden Markov model*, dostupno na linku: `https://en.wikipedia.org/wiki/Hidden_Markov_model#Applications` (studeni 2020.)

[14] *Programiranje 2*, dostupno na linku: `http://degiorgi.math.hr/prog2/kolokviji.php` (studeni 2020.)

Sažetak

U ovom radu su obrađeni skriveni Markovljevi modeli s diskretnim skupom stanja i diskretnim skupom opažanja. Uvedena je formalna definicija te su zatim objašnjeni glavni problemi modela. Opisana su njihova rješenja i algoritmi pomoću kojih dolazimo do istih. Pomoću primjera nepoštenog kasina i jednostavnog primjera iz bioinformatike pokazana je inicijalizacija modela, simulacija niza opažanja te upotreba Viterbijevog i Baum-Welchovog algoritma koristeći programski jezik R. Provedena je i primjena modela za raspoznavanje razlike između teksta i koda te zaključak o modelu i njegovim primjenama.

Summary

This thesis starts with an introduction to Markov Chains which is followed by a formal definition of Hidden Markov models with discrete state and discrete observation space. The three main problems and their solutions are described along with an algorithm for each. The example of Dishonest Casino and a simple bioinformatics application are used to show the model initialization, sequence simulation and usage of Viterbi and Baum-Welch algorithms in the programming language R. An application in text analysis is also showed in this thesis leading to a conclusion on Hidden Markov models and their usage.

Životopis

Rođena sam 22. veljače 1995. godine u Zagrebu. 2009. godine završavam OŠ Josipa Račića te potom opisujem Klasičnu gimnaziju. Nakon mature 2013. godine upisujem preddiplomski sveučilišni studij Matematika na Matematičkom odsjeku Prirodoslovno-matematičkog fakulteta u Zagrebu. Tijekom studija sam bila član studentske udruge Mladi nadareni matematičari "Marin Getaldić". Preddiplomski studij završavam 2017. godine, a 2018. godine upisujem diplomski studij Financijska i poslovna matematika na istom fakultetu.