Markovljevi modeli i primjene

Jozinović, Leonarda

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Leonarda Jozinović

MARKOVLJEVI MODELI I PRIMJENE

Diplomski rad

Voditelj rada:
dr. sc. Petra Lazić

Zagreb, veljača 2024.

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

Uvod

Markovljevi modeli vrsta su statističkih modela koji se koriste za modeliranje slučajnih procesa koji zadovoljavaju Markovljevo svojstvo. Ono kaže da budućnost ovisi samo o sadašnjosti, a ne prošlosti. Preciznije, vjerojatnost prijelaza iz jednog stanja u drugo uvjetovana je samo trenutnim stanjem, a ne svim prethodnim stanjima. Ruski matematičar Andrej Andrejevič Markov prvi je uveo Markovljeve procese 1906. godine, a tijekom 20. stoljeća Markovljevi modeli postali su ključni alati u teoriji vjerojatnosti i statistici. Najjednostavniji model je Markovljev lanac koji služi za prikazivanje procesa u kojima su sva stanja uočljiva. U slučaju kada stanja nisu direktno vidljiva, koristimo skrivene Markovljeve lance. Cilj je rekonstruirati skriveno stanje procesa na temelju promatranja. Markovljev proces odlučivanja koristi se za modeliranje procesa odlučivanja u uvjetima neizvjesnosti, a posebno je važan u području robotike i umjetne inteligencije kako bi se modelirale složene interakcije s okolinom i donosile optimalne odluke. Može se proširiti na djelomično vidljiv Markovljev proces odlučivanja koji u obzir uzima situacije u kojima donositelj odluka nema potpune informacije o svojoj okolini nego samo opažanja koja su povezana sa stvarnim stanjem okoline. Njihova sposobnost modeliranja složenih stohastičkih procesa i pružanja dubljeg uvida u dinamiku sustava čini ih neizostavnim alatom u mnogim znanstvenim, inženjerskim i primijenjenim područjima. Markovljevi modeli pokazali su se korisnima u biologiji, posebno u analizi sekvenciranja DNA i proteina te modeliranju bioloških sustava. U ekonomiji se koriste za modeliranje dinamičkih procesa na tržištima, a kroz razvoj računalne znanosti sve se više koriste u područjima prepoznavanja govora i računalnog vida.

Rad je podijeljen na četiri poglavlja. U prvom poglavlju prisjećamo se definicije i osnovnih svojstava Markovljevih lanaca. U drugom poglavlju uvodimo skrivene Markovljeve lance, navodimo tri glavna problema i njihova rješenja, a zatim neke primjene u svakodnevnom životu. Treće poglavlje obrađuje Markovljev proces odlučivanja i problem maksimizacije nagrade, a četvrto poglavlje bavi se djelomično vidljivim Markovljevim procesima kojima modeliramo složenije probleme.

Poglavlje 1

Markovljevi lanci

1.1 Definicija i osnovna svojstva

Markovljev lanac matematički je model koji služi za predviđanje vjerojatnosti niza događaja temeljenih na najnovijem događaju. Markovljevi lanci nazivaju se tako jer slijede pravilo poznato kao Markovljevo svojstvo. Markovljevo svojstvo kaže da ono što će se dogoditi sljedeće u procesu ovisi samo o trenutnom stanju, odnosno ono nema memoriju o tome što je bilo prije. Čest primjer upotrebe Markovljevog lanca je način na koji Google predviđa sljedeću riječ u rečenici prilikom pisanja *maila* temeljem prethodnog unosa unutar *maila*.

Definicija 1.1.1. *Neka je S skup. Slučajan proces s diskretnim vremenom i prostorom stanja S je familija $X = (X_n : n \geq 0)$ slučajnih varijabli definiranih na nekom vjerojatnosnom prostoru $(\Omega, \mathcal{F}, \mathbb{P})$ s vrijednostima u S.*

Definicija 1.1.2. *Slučajni proces $X = (X_n : n \geq 0)$ s vrijednostima u skupu S je Markovljev lanac ako vrijedi*

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

za svaki $n \geq 0$ i za sve $i_0, \ldots, i_{n-1} i, j \in S$ za koje su obje uvjetne vjerojatnosti dobro definirane.

Napomena 1.1.3. *Svojstvo u relaciji (1.1) naziva se Markovljevo svojstvo. Drugi način na* *koji možemo iskazati to svojstvo je sljedeće:*

$$\begin{aligned}
&\mathbb{P}(X_{n+1} = j, X_{n-1} = i_{n-1}, \dots, X_0 = i_0 \mid X_n = i) \\
&= \frac{\mathbb{P}(X_{n+1} = j, X_n = i, X_{n-1} = i_{n-1} \dots, X_0 = i_0)}{\mathbb{P}(X_n = i)} \\
&= \frac{\mathbb{P}(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0)\,\mathbb{P}(X_n = i, X_{n-1} = i_{n-1}, \dots, X_0 = i_0)}{\mathbb{P}(X_n = i)} \\
&= \mathbb{P}(X_{n+1} = j \mid X_n = i)\,\mathbb{P}(X_{n-1} = i_{n-1}, \dots, X_0 = i_0 \mid X_n = i),
\end{aligned} \tag{1.2}$$

odnosno budućnost i prošlost uvjetno su nezavisne uz danu sadašnjost.

Markovljevi lanci u kojima desna strana u definiciji 1.1.2 ne ovisi o vremenu n zovu se homogeni Markovljevi lanci.

Definicija 1.1.4. *Neka je S najviše prebrojiv skup. Matrica $A = (a_{ij} : i, j \in S)$ naziva se stohastičkom matricom ako je $a_{ij} \geq 0$ za sve $i, j \in S$, te*

$$\sum_{j \in S} a_{ij} = 1, \quad \text{za sve } i \in S.$$

Definicija 1.1.5. *Neka je $\pi = (\pi_i : i \in S)$ vjerojatnosna distribucija na S, te neka je $A = (a_{ij} : i, j \in S)$ stohastička matrica. Homogen Markovljev lanac $X = (X_n : n \geq 0)$ imat će početnu dsitribuciju π i prijelaznu matricu A ako vrijedi*

(i)

$$\mathbb{P}(X_0 = i) = \pi_i, \text{ za sve } i \in S$$

(ii)

$$\mathbb{P}(X_{n+1} = j \mid X_n = i) = a_{ij} \tag{1.3}$$

za svaki $n \geq 0$ i za sve $i_0, \dots, i_{n-1}, i, j \in S$.

Takav Markovljev lanac nazivamo (π, A)-Markovljevim lancom. Spomenut ćemo još pitanje konačnodimenzionalnih distribucija, odnosno zanimaju nas vjerojatnosti s kojima se slučajni proces u danim vremenskim trenutcima nalazi u danim stanjima. Odgovor na to pitanje daje sljedeći teorem.

Teorem 1.1.6. *Neka je X (π, A)-Markovljev lanac. Tada za sve $n \geq 0$ i za sva stanja $i_0, i_1, \dots, i_{n-1}, i_n$ vrijedi*

$$\mathbb{P}(X_0 = i_0, X_1 = i_1, \dots, X_{n-1} = i_{n-1}, X_n = i_n) = \pi_{i_0} a_{i_0 i_1} \cdots a_{i_{n-1} i_n}. \tag{1.4}$$

Obratno, pretpostavimo da je $X = (X_n : n \geq 0)$ slučajan proces s konačnodimenzionalnim distribucijama danim formulom (1.4), gdje je π neka vjerojatnosna distribucija na S, a A neka stohastička matrica na S. Tada je X (π, A)-Markovljev lanac.

Dokaz. Koristeći poopćenje formule za uvjetnu vjerojatnost

$$\mathbb{P}(D_0 \cap D_1 \cap \ldots \cap D_n) = \mathbb{P}(D_0)\,\mathbb{P}(D_1 \mid D_0)\,\mathbb{P}(D_2 \mid D_0 \cap D_1) \ldots \mathbb{P}(D_n \mid D_0 \cap \ldots \cap D_{n-1})$$

dobivamo

$$\begin{aligned}
&\mathbb{P}(X_0 = i_0, X_1 = i_1, \ldots, X_{n-1} = i_{n-1}, X_n = i_n) \\
&= \mathbb{P}(X_0 = i_0)\,\mathbb{P}(X_1 = i_1 \mid X_0 = i_0) \ldots \mathbb{P}(X_n = i_n \mid X_0 = i_0, \ldots, X_{n-1} = i_{n-1}) \\
&\overset{(1.3)}{=} \pi_{i_0} a_{i_0 i_1} \cdots a_{i_{n-1} i_n}.
\end{aligned}$$

Obratno, uzimanjem $n = 0$ u (1.4) slijedi da je π početna distribucija. Dokažimo još (1.3). Pretpostavimo da je $\mathbb{P}(X_0 = i_0, X_1 = i_1, \ldots, X_n = i) > 0$. Tada je

$$\begin{aligned}
&\mathbb{P}(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) \\
&= \frac{\mathbb{P}(X_{n+1} = j, X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0)}{\mathbb{P}(X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0)} \\
&= \frac{\pi_{i_0} a_{i_0 i_1} \cdots a_{i_{n-1} i} a_{ij}}{\pi_{i_0} a_{i_0 i_1} \cdots a_{i_{n-1} i}} \\
&= a_{ij}
\end{aligned}$$

□

Množenje beskonačnih stohastičkih matrica definira se analogno kao množenje konačnih matrica. Ako su $P = (p_{ij} : i, j \in S)$ i $Q = (q_{ij} : i, j \in S)$ matrice, definiramo matricu $PQ = (r_{ij} : i, j \in S)$ s

$$r_{ij} = \sum_{k \in S} p_{ik} q_{kj}.$$

Specijalno, n-ta potencija matrice P dana je s $P^{(n)} = (p_{ij}^{(n)} : i, j \in S)$, gdje je

$$p_{ij}^{(n)} = \sum_{i_1 \in S} \cdots \sum_{i_{n-1} \in S} p_{ii_1} i_{i_1 i_2} \cdots p_{i_{n-1} j}. \tag{1.5}$$

Nultu potenciju matrice P definiramo kao $P^0 = I = (\delta_{ij})$, gdje je $\delta_{ii} = 1$, a $\delta_{ij} = 1$ za $i \neq j$. Iz formule za konačnodimenzionalne distribucije možemo izračunati jednodimenzionalne. Koristeći gornju formulu imamo

$$\begin{aligned}
\mathbb{P}(X_n = j) &= \sum_{i_0 \in S} \sum_{i_1 \in S} \cdots \sum_{i_{n-1} \in S} \pi_{i_0} a_{i_0 i_1} \cdots a_{i_{n-1} j} \\
&= \sum_{i_0 \in S} \pi_{i_0} a_{i_0 j}^{(n)} = (\pi A^n)_j.
\end{aligned} \tag{1.6}$$

πA^n označava produkt vektor-retka π i matrice A, a $(\pi A^n)_j$ j-ti element rezultirajućeg vektor-retka. Definirajmo vjerojatnost da Markovljev lanac koji kreće iz stanja i nakon n koraka bude u stanju j kao

$$\mathbb{P}_i(X_n = j) = \mathbb{P}(X_n = j \mid X_0 = i). \tag{1.7}$$

Iz formule (1.6), uzimanjem $\pi = (0, \ldots, 0, 1, 0, \ldots, 0)$, gdje se 1 nalazi na i-tom mjestu, slijedi da je za sve $n \geq 0$,

$$\mathbb{P}_i(X_n = j) = a_{ij}^{(n)}. \tag{1.8}$$

Ta formula nam govori da ako Markovljev lanac kreće iz stanja i, vjerojatnost da nakon n koraka bude u stanju j jednaka je ij-tom elementu n-te potencije prijelazne matrice A.

Spomenimo još jedan važan primjer Markovljevog lanca.

Primjer 1.1.7. *(Slučajna šetnja) Neka je $Y = (Y_n : n \geq 1)$ niz nezavisnih, jednakodistribuiranih slučajnih varijabli s vrijednostima u $\mathbb{Z}$ s distribucijom $\mathbb{P}(Y_1 = k) = a_k, k \in \mathbb{Z}$. Definiramo slučajnu šetnju $X = (X_n : n \geq 0)$ sa*

$$X_0 = 0, X_n = \sum_{k=1}^{n} Y_k, n \geq 1.$$

Za $n \geq 0$ vrijedi $X_{n+1} = X_n + Y_{n+1}$. Slučajna šetnja X je Markovljev lanac:

$$\begin{aligned}
&\mathbb{P}(X_{n+1} = i_{n+1} \mid X_n = i_n, \ldots, X_1 = i_1, X_0 = i_0) \\
&\quad = \mathbb{P}(Y_{n+1} + i_n = i_{n+1} \mid X_n = i_n, \ldots, X_1 = i_1, X_0 = i_0) \\
&\quad = \mathbb{P}(Y_{n+1} = i_{n+1} - i_n) \\
&\quad = a_{i_{n+1} - i_n} \\
&\mathbb{P}(X_{n+1} = i_{n+1} \mid X_n = i_n) \\
&\quad = \mathbb{P}(Y_{n+1} + i_n = i_{n+1} \mid X_n = i_n) \\
&\quad = \mathbb{P}(Y_{n+1} = i_{n+1} - i_n) \\
&\quad = a_{i_{n+1} - i_n},
\end{aligned}$$

gdje treći i sedmi redak slijede iz nezavisnosti Y_{n+1} sa $X_0, X_1, \ldots, X_n$.

1.2 Primjene

Primjer 1.2.1. *(PageRank [2]) PageRank jedna je od metoda koju Google koristi za određivanje relevantnosti web stranice kako bi osigurao da se važnije stranice pojave ranije u rezultatima pretraživanja. Web možemo zamisliti kao usmjereni graf u kojem su stranice* *vrhovi, a poveznice bridovi. Brid* (u, v) *upućuje da stranica u sadrži poveznicu koji vodi na stranicu v. Ideja je modelirati slučajnog surfera koji slijedi poveznice u grafu, odnosno izvodi slučajnu šetnju. Nakon dovoljnog broja koraka možemo rangirati web stranice prema posjećenosti. Pretpostavljamo da su stranice češće posjećene ako su povezane s puno drugih stranica što smatramo dobrim mjerilom važnosti stranice. Pretpostavimo da se sufer kreće po grafu na sljedeći način: većinu vremena će sa stranice i slijediti izlazne veze i kretati se prema jednom od susjeda stranice i. No manji postotak vremena će mu trenutna stranica dosaditi te će ju napustiti i slučajno izabrati novu stranicu na koju će se teleportirati. Zato uvodimo faktor d koji predstavlja vjerojatnost da će surfer napustiti trenutnu stranicu i teleportirati se na novu. Za d se najčešće uzima* 0.85. *Može se teleportirati na bilo koju stranicu pa svaka stranica ima vjerojatnost* $\frac{1}{N}$ *da bude izabrana, gdje je N ukupan broj stranica. Definirajmo pojednostavljenu verziju PageRank algoritma na sljedeći način. Pretpostavimo da stranica* p_j *sadrži* $L(p_j)$ *poveznica. Ako s* $M(p_i)$ *označimo skup svih stranica koje sadrže poveznicu na stranicu* p_i*, onda ocjenu važnosti, odnosno PageRank* $PR(p_i)$ *stranice* p_i *dobivamo formulom*

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

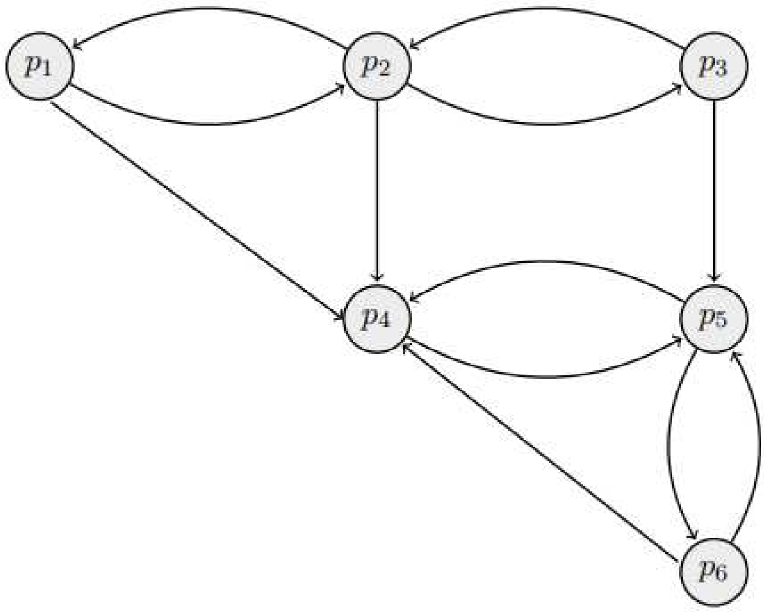
Pretpostavimo da na webu postoji samo šest stranica. Svakoj stranici ćemo pridružiti početni PageRank $\frac{1}{6}$. *Izračunajmo ocjenu važnosti za stranicu* p_2.

$$\begin{aligned} PR(p_2) &= \frac{1-0.85}{6} + 0.85\left(\frac{PR(p_1)}{2} + \frac{PR(p_2)}{3}\right) \\ &= \frac{1-0.85}{6} + 0.85\left(\frac{1}{6\cdot 2} + \frac{1}{6\cdot 2}\right) \\ &= 0.16 \end{aligned}$$

Računanjem ostalih PageRankova dobijemo:

$$\begin{aligned} PR(p_1) &= 0.072 \\ PR(p_2) &= 0.16 \\ PR(p_3) &= 0.072 \\ PR(p_4) &= 0.285 \\ PR(p_5) &= 0.308 \\ PR(p_6) &= 0.096 \end{aligned}$$

Vidimo da stranica p_5 *ima najveći PageRank pa će se najranije pojaviti u rezultatima pretraživanja, a nakon nje stranica* p_4. *Taj rezultat ima smisla jer po tri poveznice vode do stranice* p_5 *i* p_4*, no stranica* p_5 *sadrži dvije poveznice, a stanica* p_4 *samo jednu. Dakle stranica* p_5 *je najbolje povezana s ostalim stranicama što ju čini najrelevantnijom.*



Slika 1.1: Graf pojednostavljenog weba. Slika preuzeta iz [2].

Primjer 1.2.2. *(Predviđanje vremenske prognoze [2]) Jedna od najvažnijih primjena Markovljevih lanaca u svakodnevnom životu je predviđanje vremenske prognoze. Pogledajmo primjer u kojem želimo predvidjeti vrijeme u gradu Bemidji u državi Minnesota u sjevernom dijelu SAD-a. Radi jednostavnosti, pretpostavimo da su moguća samo tri stanja: vrijeme može biti oblačno, sunčano ili može padati snijeg. Pretpostavit ćemo i da sutrašnje vrijeme ovisi samo o današnjem. Uzimajući podatke o vremenu u podne svaki dan u studenome 2019. godine, možemo konstruirati prijelaznu matricu A:*

$$A = \begin{pmatrix} 0.57 & 0.22 & 0.5 \\ 0.21 & 0.44 & 0.1 \\ 0.21 & 0.33 & 0.33 \end{pmatrix}$$

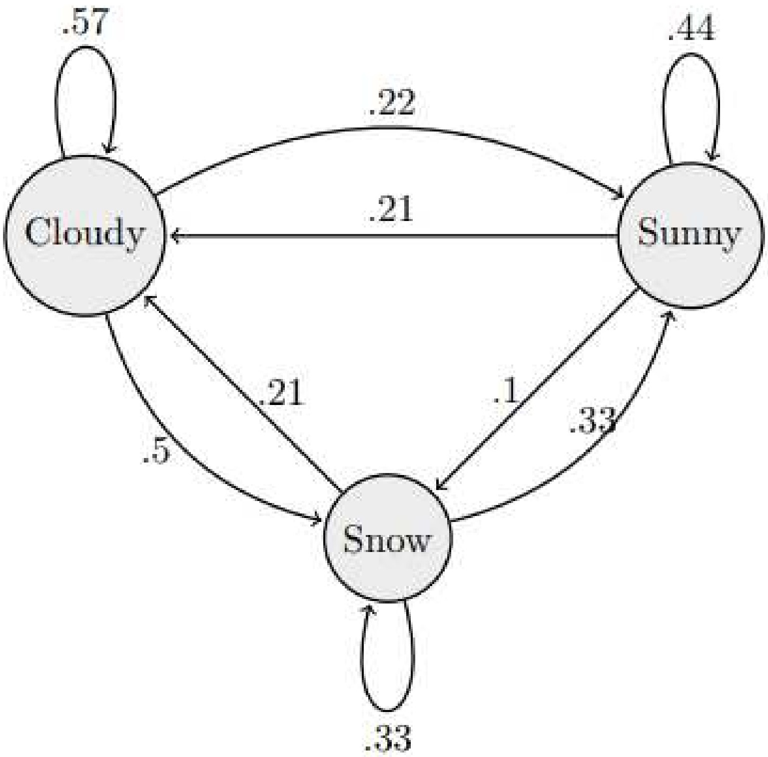
Pretpostavimo da je 12. studenog 2020. godine u podne oblačno i želimo predvidjeti kakvo vrijeme će biti za tri dana, 15. studenog. Ako oblačno vrijeme prikažemo vektorom

$$x_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},$$

onda imamo

$$x_3 = A^3 \cdot x_0 = \begin{pmatrix} 0.57 & 0.22 & 0.5 \\ 0.21 & 0.44 & 0.1 \\ 0.21 & 0.33 & 0.33 \end{pmatrix}^3 \cdot \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0.44 \\ 0.30 \\ 0.23 \end{pmatrix}$$

Dakle, vjerojatnost da će za tri dana biti oblačno je 0.44*, da će biti sunčano* 0.30*, a da će padati snijeg* 0.23.



Slika 1.2: Dijagram stanja. Slika preuzeta iz [2].

Poglavlje 2

Skriveni Markovljevi modeli

2.1 Definicija i osnovna svojstva

Skriveni Markovljev model dvostruki je slučajan proces koji se sastoji od Markovljevog lanca čija stanja su skrivena i opservacijskog procesa koji generira niz opažanja. S obzirom da nam stanja Markovljevog lanca nisu direktno dostupna, glavna ideja je moći izvesti zaključke o lancu promatrajući samo niz opažanja. Na primjer, u teoriji komunikacija, signal koji se prenosi preko komunikacijskog kanala možemo zamisliti kao skriveno stanje, no kako je kanal bučan, korisnik prima izmijenjenu verziju tog signala i želi što preciznije rekonstruirati originalni signal iz svog opažanja. Model se prvotno pojavio u području prepoznavanja govora, a danas ima sve veću primjenu i u ostalim poljima, poput bioinformatike, ekonomije, termodinamike i slično.

Definicija 2.1.1. *Neka su* $X = (X_n : n \geq 0)$ *i* $Y = (Y_n : n \geq 0)$ *slučajni procesi s diskretnim vremenom i vrijednostima u skupu* S*, odnosno* V*. Par* $(X, Y) = \{(X_n, Y_n) : n \geq 0\}$ *je skriveni Markovljev model (HMM) ako vrijedi:*

(i) *X je Markovljev lanac*

(ii)

$$\begin{aligned} &\mathbb{P}(Y_n = v \mid X_n = s, X_{n-1} = s_{n-1}, Y_{n-1} = v_{n-1} \ldots, X_0 = s_0, Y_0 = v_0) \\ &= \mathbb{P}(Y_n = v \mid X_n = s) \end{aligned} \tag{2.1}$$

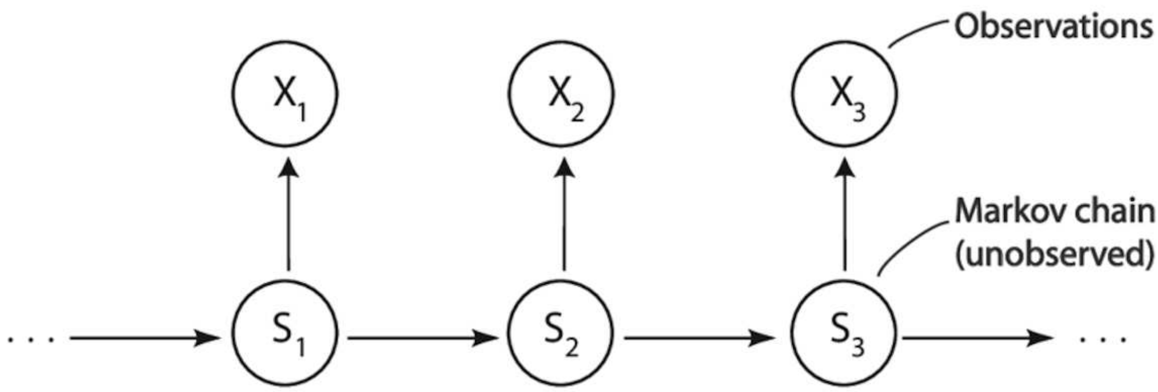
za sve $n \geq 0$ *i za sve* $s_0, \ldots, s_{n-1}, s \in S, v_0, \ldots, v_{n-1}, v \in V$.

Ako uvjetna vjerojatnost (2.1) ne ovisi o vremenu n, kažemo da je skriveni Markovljev model homogen.

Homogen skriveni Markovljev model može se okarakterizirati i pomoću sljedećih elemenata:

- skupa stanja $S = \{s_0, s_1, \dots, s_N\}$ skrivenog procesa $X = (X_n : n \geq 0)$
- skupa stanja $V = \{v_0, v_1, \dots, v_M\}$ opservacijskog procesa $Y = (Y_n : n \geq 0)$. Ta stanja nekad zovemo i opservacijski simboli, emisije ili opažanja.
- matrice prijelaznih vjerojatnosti $A = \{a_{ij}\}$ skrivenog procesa X, gdje je a_{ij} vjerojatnost da je proces u trenutku $n+1$ u stanju s_j, ako je u trenutku n u stanju s_i, tj. $a_{ij} = \mathbb{P}\Big(X_{n+1} = s_j \mid X_n = s_i\Big)$. Matrica A je stohastička matrica, tj. vrijedi $a_{ij} \geq 0$ za sve $0 \leq i, j \leq N$, te
$$\sum_{j=0}^{N} a_{ij} = 1, \ 0 \leq i \leq N. \tag{2.2}$$
- matrice opservacijskih vjerojatnosti $B = \{b_j(k)\}$, gdje je $b_j(k)$ vjerojatnost da se simbol v_k emitira u stanju s_j, tj. $b_j(k) = \mathbb{P}\Big(Y_n = v_k \mid X_n = s_j\Big)$, $\forall n, 0 \leq j \leq N, 0 \leq k \leq M$. Matrica zadovoljava stohastička svojstva $b_j(k) \geq 0$, za sve $0 \leq j \leq N$, $0 \leq k \leq M$, te
$$\sum_{k=0}^{M} b_j(k) = 1, \ 0 \leq j \leq N. \tag{2.3}$$
- početne distribucije $\pi = \{\pi_i\}$ skrivenog procesa $X = (X_n : n \geq 0)$, gdje je π_i vjerojatnost da je model u stanju s_i u trenutku $n = 0$, tj. $\pi_i = \mathbb{P}(X_0 = s_i)$.

Takav skriveni Markovljev model označavat ćemo s $\lambda = (A, B, \pi)$.



Slika 2.1: Grafički prikaz HMM

Slučajne varijable $(Y_0, Y_1, \dots, Y_n)$ uvjetno su nezavisne uz dano $(X_0, X_1, \dots, X_n)$:

$$\begin{aligned}
&\mathbb{P}(Y_0 = v_0, Y_1 = v_1, \dots, Y_n = v_n \mid X_0 = s_0, X_1 = s_1, \dots, X_n = s_n) \\
&= \mathbb{P}(Y_0 = v_0 \mid X_0 = s_0)\,\mathbb{P}(Y_1 = v_1, \dots, Y_n = v_n \mid X_0 = s_0, X_1 = s_1, \dots, X_n = s_n, Y_0 = v_0) \\
&= \mathbb{P}(Y_0 = v_0 \mid X_0 = s_0)\,\mathbb{P}(Y_1 = v_1 \mid X_0 = s_0, Y_0 = v_0, X_1 = s_1) \\
&\cdot \mathbb{P}(Y_2 = v_2, \dots, Y_n = v_n \mid X_0 = s_0, X_1 = s_1, \dots, X_n = s_n, Y_0 = v_0, Y_1 = v_1) \\
&= \dots \\
&= \prod_{t=0}^{n} \mathbb{P}(Y_t = v_t \mid X_0 = s_0, \dots, X_t = s_t, Y_0 = v_0, \dots, Y_{t-1} = v_{t-1}) \\
&= \prod_{t=0}^{n} \mathbb{P}(Y_t = v_t \mid X_t = s_t)
\end{aligned} \tag{2.4}$$

Koristeći činjenicu da X zadovoljava svojstva (1.4) i (2.4), možemo izračunati vjerojatnost produkta, tj. vjerojatnost da HMM generira niz opservacija $\mathbf{v}_T = (v_{j_o}, v_{j_1}, \dots, v_{j_T})$ s nizom skrivenih stanja $\mathbf{s}_T = (s_{i_o}, s_{i_1}, \dots, s_{i_T})$, $0 \leq i_0, i_1, \dots, i_T \leq N$, $0 \leq j_0, j_1, \dots, j_T \leq M$. Radi jednostavnosti označimo $\mathbf{X}_T = (X_0, X_1, \dots, X_T)$ i $\mathbf{Y}_T = (Y_0, Y_1, \dots, Y_T)$. Sada je:

$$\begin{aligned}
\mathbb{P}(\mathbf{X}_T = \mathbf{s}_T, \mathbf{Y}_T = \mathbf{v}_T) &= \mathbb{P}(\mathbf{Y}_T = \mathbf{v}_T \mid \mathbf{X}_T = \mathbf{s}_T) \cdot \mathbb{P}(\mathbf{X}_T = \mathbf{s}_T) \\
&= \prod_{t=0}^{T} \mathbb{P}\left(Y_t = v_{j_t} \mid X_t = s_{i_t}\right) \cdot \pi_{i_0} \prod_{t=1}^{T} a_{i_{t-1} i_t} \\
&= \prod_{t=0}^{T} b_{i_t}(j_t) a_{i_{t-1} i_t}, \text{ gdje je } a_{i_{-1} i_0} = \pi_{i_0}.
\end{aligned} \tag{2.5}$$

Napomena 2.1.2. *Uočimo da (X, Y) zadovoljava ekvivalent Markovljevog svojstva za skriveni model. Iz definicije slijedi:*

$$\begin{aligned}
&\mathbb{P}(X_{n+1} = s_{n+1}, Y_{n+1} = v_{n+1} \mid X_n = s_n, Y_n = v_n, \dots, X_0 = s_0, Y_0 = v_0) \\
&= \mathbb{P}(Y_{n+1} = v_{n+1} \mid X_{n+1} = s_{n+1}, X_n = s_n, Y_n = v_n, \dots, X_0 = s_0, Y_0 = v_0) \\
&\cdot \mathbb{P}(X_{n+1} = s_{n+1} \mid X_n = s_n, Y_n = v_n \dots, X_0 = s_0, Y_0 = v_0) \\
&= \mathbb{P}(Y_{n+1} = v_{n+1} \mid X_{n+1} = s_{n+1}) \cdot \mathbb{P}(X_{n+1} = s_{n+1} \mid X_n = s_n) \\
&= b_{n+1}(n+1) \cdot a_{nn+1}
\end{aligned} \tag{2.6}$$

Važno je napomenuti da $\mathbb{P}(X_{n+1} = s_{n+1} \mid X_n = s_n, Y_n = v_n \dots, X_0 = s_0, Y_0 = v_0) =$ $\mathbb{P}(X_{n+1} = s_{n+1} \mid X_n = s_n)$ ne slijedi iz činjenice da su X i Y nezavisni, već iz činjenice da X_n ne ovisi o prošlom opažanju Y_{n-1}. Dakle, iako su (X, Y) i X Markovljevi procesi, Y ne mora biti pa HMM možemo koristiti i za modeliranje ponašanja koja ne zadovoljavaju Markovljevo svojstvo.

2.2 Tri osnovna problema i njihova rješenja

2.2.1 Problem evaluacije

Pretpostavimo da nam je dan model $\lambda = (A, B, \pi)$ i niz opažanja $v_{j_0}, v_{j_1}, \ldots, v_{j_T} \in V$. Zanima nas vjerojatnost da je taj niz generiran modelom λ, tj. zanima nas vjerojatnost

$$\mathbb{P}\left(Y_0 = v_{j_0}, Y_1 = v_{j_1}, \ldots, Y_T = v_{j_T} \mid \lambda\right) \tag{2.7}$$

što ćemo dalje označavati s $\mathbb{P}_\lambda\left(Y_0 = v_{j_0}, Y_1 = v_{j_1}, \ldots, Y_T = v_{j_T}\right)$. Ako imamo više modela, rješenje problema evaluacije dat će nam model koji najbolje opisuje dana opažanja, odnosno onaj za kojeg je tražena vjerojatnost najveća. Ovo možemo direktno izračunati koristeći sljedeću lemu.

Lema 2.2.1. *Vrijedi*

$$\mathbb{P}_\lambda\left(Y_0 = v_{j_0}, Y_1 = v_{j_1}, \ldots, Y_T = v_{j_T}\right) = \sum_{0 \le i_0, i_1, \ldots, i_T \le N} \prod_{t=0}^{T} b_{i_t}(j_t) a_{i_{t-1} i_t}, \tag{2.8}$$

gdje je $a_{i_{-1} i_0} = \pi_{i_0}$.

Dokaz. Promatramo sve moguće nizove skrivenih stanja $\mathbf{s}_T$ od vremena 0 do vremena T te vjerojatnost dobijemo sumiranjem produktnih vjerojatnosti niza $\mathbf{v}_T = (v_{j_o}, v_{j_1}, \ldots, v_{j_T})$ i svih mogućih nizova $\mathbf{s}_T$:

$$\begin{aligned} &\mathbb{P}_\lambda\left(Y_0 = v_{j_0}, Y_1 = v_{j_1}, \ldots, Y_T = v_{j_T}\right) \\ &= \mathbb{P}_\lambda\left(\mathbf{Y}_T = \mathbf{v}_T\right) \\ &= \sum_{\textit{svi moguci } \mathbf{s}_T} \mathbb{P}_\lambda\left(\mathbf{X}_T = \mathbf{s}_T, \mathbf{Y}_T = \mathbf{v}_T\right) \\ &\overset{(2.5)}{=} \sum_{0 \le i_0, i_1, \ldots, i_T \le N} \prod_{t=0}^{T} b_{i_t}(j_t) a_{i_{t-1} i_t} \end{aligned} \tag{2.9}$$

□

Iako lema daje direktnu metodu za računanje, ona nije efikasna jer je složenosti reda $(T + 1) \cdot (N + 1)^{T+1}$. Koristimo efikasnije metode koje se zovu *forward* i *backward* metoda.

Forward algoritam

Definiramo pomoćnu *forward* varijablu koja predstavlja vjerojatnost parcijalnog niza opažanja $v_{j_0}, v_{j_1}, \ldots, v_{j_t}$ (do trenutka t) i skrivenog stanja s_{i_t} u trenutku t:

$$\alpha_t(i_t) = \mathbb{P}_\lambda\left(Y_0 = v_{j_0}, Y_1 = v_{j_1}, \ldots, Y_t = v_{j_t}, X_t = s_{i_t}\right)$$

za $t = 0, 1, \dots, T,\ 0 \le i_t \le N$.

Propozicija 2.2.2. *Forward varijable zadovoljavaju rekurziju*

$$\alpha_{t+1}(i_{t+1}) = b_{i_{t+1}}(j_{t+1}) \left[\sum_{i_t=0}^{N} \alpha_t(i_t) a_{i_t i_{t+1}} \right],\ 0 \le i_{t+1} \le N, t = 0, \dots, T-1 \tag{2.10}$$

Dokaz.

$$\begin{aligned}
\alpha_{t+1}(i_{t+1}) &= \mathbb{P}_\lambda \left(Y_0 = v_{j_0}, Y_1 = v_{j_1}, \dots, Y_{t+1} = v_{j_{t+1}}, X_{t+1} = s_{i_{t+1}} \right) \\
&= \sum_{0 \le i_0, \dots, i_t \le N} \mathbb{P}_\lambda \left(Y_0 = v_{j_0}, \dots, Y_{t+1} = v_{j_{t+1}}, X_0 = s_{i_0}, \dots, X_t = s_{i_t}, X_{t+1} = s_{i_{t+1}} \right) \\
&= \sum_{0 \le i_0, \dots, i_t \le N} \mathbb{P}_\lambda \left(Y_{t+1} = v_{j_{t+1}} \mid Y_0 = v_{j_0}, \dots, Y_t = v_{j_t}, X_0 = s_{i_0}, \dots, X_{t+1} = s_{i_{t+1}} \right) \\
&\cdot \mathbb{P}_\lambda \left(Y_0 = v_{j_0}, \dots, Y_t = v_{j_t}, X_0 = s_{i_0}, \dots, X_t = s_{i_t}, X_{t+1} = s_{i_{t+1}} \right) \\
&= \sum_{0 \le i_0, \dots, i_t \le N} \mathbb{P}_\lambda \left(Y_{t+1} = v_{j_{t+1}} \mid X_{t+1} = s_{i_{t+1}} \right) \\
&\cdot \mathbb{P}_\lambda \left(X_{t+1} = s_{i_{t+1}} \mid Y_0 = v_{j_0}, \dots, Y_t = v_{j_t}, X_0 = s_{i_0}, \dots, X_t = s_{i_t} \right) \\
&\cdot \mathbb{P}_\lambda \left(Y_0 = v_{j_0}, \dots, Y_t = v_{j_t}, X_0 = s_{i_0}, \dots, X_t = s_{i_t} \right) \\
&= \mathbb{P}_\lambda \left(Y_{t+1} = v_{j_{t+1}} \mid X_{t+1} = s_{i_{t+1}} \right) \sum_{0 \le i_0, \dots, i_t \le N} \mathbb{P}_\lambda \left(X_{t+1} = s_{i_{t+1}} \mid X_t = s_{i_t} \right) \\
&\cdot \mathbb{P}_\lambda \left(Y_0 = v_{j_0}, \dots, Y_t = v_{j_t}, X_0 = s_{i_0}, \dots, X_t = s_{i_t} \right) \\
&= b_{i_{t+1}}(j_{t+1}) \sum_{i_t=0}^{N} a_{i_t i_{t+1}} \alpha_t(i_t)
\end{aligned} \tag{2.11}$$

□

Forward algoriram sastoji se od tri koraka:

1. Inicijalizacija:
$$\alpha_0(i_0) = \pi_{i_0} b_{i_0}(j_0),\ 0 \le i_0 \le N$$

2. Rekurzija:
$$\alpha_{t+1}(i_{t+1}) = b_{i_{t+1}}(j_{t+1}) \left[\sum_{i_t=0}^{N} \alpha_t(i_t) a_{i_t i_{t+1}} \right],\ t = 0, \dots, T-1, 0 \le i_{t+1} \le N$$

3. Kraj:
$$\mathbb{P}_\lambda(\mathbf{Y}_T = \mathbf{v}_T) = \sum_{i_T=0}^{N} \mathbb{P}_\lambda(\mathbf{Y}_T = \mathbf{v}_T, X_T = s_{i_T}) = \sum_{i_T=0}^{N} \alpha_T(i_T)$$

Vidimo da *forward* metoda ima složenost reda $(N+1)^2 \cdot (T+1)$ što je puno efikasnije od direktnog računanja.

Primjer 2.2.3. *Ivana ima puno različitih hobija. Tijekom dana bavi se jednim od svoja četiri hobija: svira klavir, peče kolače, vodi psa u šetnju ili biciklira. Izbor aktivnosti za određeni dan i vremenska prognoza povezani su:*

- *Ako Ivana svira, vjerojatnost da je sunčano je* 0.4*, a vjerojatnost da je oblačno je* 0.3.
- *Ako Ivana peče kolače, vjerojatnost da je sunčano je* 0.1*, a vjerojatnost da je oblačno je* 0.45.
- *Ako Ivana ide u šetnju, vjerojatnost da je sunčano je* 0.2*, a vjerojatnost da je oblačno je* 0.2.
- *Ako Ivana biciklira, vjerojatnost da je sunčano je* 0.3*, a vjerojatnost da je oblačno je* 0.05.

Znamo i da sutrašnje vrijeme ovisi o današnjem:

- *Ako je danas sunčano, vjerojatnost da će i sutra biti sunčano je* 0.8*, a vjerojatnost da će biti oblačno* 0.2.
- *Ako je danas oblačno, vjerojatnost da će i sutra biti oblačno je* 0.6*, a vjerojatnost da će biti sunčano* 0.4.
- *Također znamo da je vjerojatnost da je vrijeme sunčano na prvi dan promatranja Ivaninog ponašanja* 0.6*, a da je oblačno* 0.4.

Pretpostavimo da smo četiri dana promatrali Ivanin izbor hobija i dobili smo niz: sviranje, pečenje, šetnja i bicikliranje. Pretpostavimo da nemamo pristup vremenskoj prognozi, tada su stanja sunčano i oblačno skrivena. Ovaj problem možemo modelirati koristeći skriveni Markovljev model.

Neka je S *skup skrivenih stanja, odnosno* $S = \{$*sunčano, oblačno*$\}$ *i neka je* V *skup opservacija, odnosno* $V = \{$*sviranje, pečenje, šetnja, bicikliranje*$\}$*. Također imamo zadane matrice prijelaznih i opservacijskih vjerojatnosti te početnu distribuciju.*

$$A = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \end{bmatrix}, \ B = \begin{bmatrix} 0.4 & 0.1 & 0.2 & 0.3 \\ 0.3 & 0.45 & 0.2 & 0.05 \end{bmatrix}, \ \pi = \begin{bmatrix} 0.6 & 0.4 \end{bmatrix}$$

Zanima nas vjerojatnost da je zadani model emitirao niz opažanja (sviranje, pečenje, šetnja, bicikliranje). Koristeći forward algoritam imamo:

1. *Inicijalizacija:*

$$\alpha_0(sunčano) = \pi_{sunčano} \cdot b_{sunčano}(sviranje) = 0.6 \cdot 0.4 = 0.24$$
$$\alpha_0(oblačno) = \pi_{oblačno} \cdot b_{oblačno}(sviranje) = 0.4 \cdot 0.3 = 0.12$$

2. *Rekurzija:*

$t = 0$:

$$\begin{aligned}
\alpha_1(sunčano) &= (\alpha_0(sunčano) \cdot a_{sunčano,sunčano} + \alpha_0(oblačno) \cdot a_{oblačno,sunčano}) \\
&\quad \cdot b_{sunčano}(pečenje) \\
&= (0.24 \cdot 0.8 + 0.12 \cdot 0.4) \cdot 0.1 = 0.024 \\
\alpha_1(oblačno) &= (\alpha_0(sunčano) \cdot a_{sunčano,oblačno} + \alpha_0(oblačno) \cdot a_{oblačno,oblačno}) \\
&\quad \cdot b_{oblačno}(pečenje) \\
&= (0.24 \cdot 0.2 + 0.12 \cdot 0.6) \cdot 0.45 = 0.054
\end{aligned}$$

$t = 1$:

$$\begin{aligned}
\alpha_2(sunčano) &= (\alpha_1(sunčano) \cdot a_{sunčano,sunčano} + \alpha_1(oblačno) \cdot a_{oblačno,sunčano}) \\
&\quad \cdot b_{sunčano}(šetnja) \\
&= (0.024 \cdot 0.8 + 0.054 \cdot 0.4) \cdot 0.2 = 0.00816 \\
\alpha_2(oblačno) &= (\alpha_1(sunčano) \cdot a_{sunčano,oblačno} + \alpha_1(oblačno) \cdot a_{oblačno,oblačno}) \\
&\quad \cdot b_{oblačno}(šetnja) \\
&= (0.024 \cdot 0.2 + 0.054 \cdot 0.6) \cdot 0.2 = 0.00744
\end{aligned}$$

$t = 2$:

$$\begin{aligned}
\alpha_3(sunčano) &= (\alpha_2(sunčano) \cdot a_{sunčano,sunčano} + \alpha_2(oblačno) \cdot a_{oblačno,sunčano}) \\
&\quad \cdot b_{sunčano}(bicikliranje) \\
&= (0.00816 \cdot 0.8 + 0.00744 \cdot 0.4) \cdot 0.3 = 0.0028512 \\
\alpha_3(oblačno) &= (\alpha_2(sunčano) \cdot a_{sunčano,oblačno} + \alpha_2(oblačno) \cdot a_{oblačno,oblačno}) \\
&\quad \cdot b_{oblačno}(bicikliranje) \\
&= (0.00816 \cdot 0.2 + 0.00744 \cdot 0.6) \cdot 0.05 = 0.0003048
\end{aligned}$$

3. *Kraj:*

$$\begin{aligned}
&\mathbb{P}_\lambda\,(sviranje,\ pečenje,\ šetnja,\ bicikliranje) = \alpha_3(sunčano) + \alpha_3(oblačno) \\
&= 0.0028512 + 0.0003048 = 0.003156
\end{aligned}$$

Backward algoritam

Druga metoda je *backward* metoda. Postupak je sličan *forward* metodi, a razlika je što ovdje računamo unatrag. Ponovno definiramo pomoćnu *backward* varijablu koja predstavlja vjerojatnost parcijalnog niza opažanja (od trenutka $t+1$ do kraja) uz uvjet da je lanac u skrivenom stanju s_{i_t} u trenutku t:

$$\beta_t(i_t) = \mathbb{P}_\lambda\left(Y_{t+1} = v_{j_{t+1}}, Y_{t+2} = v_{j_{t+2}}, \ldots, Y_T = v_{j_T} \mid X_t = s_{i_t}\right),$$

za $0 \le t \le T-1, 0 \le i_t \le N$.

Propozicija 2.2.4. *Backward varijable zadovoljavaju rekurziju*

$$\beta_t(i_t) = \sum_{i_{t+1}=0}^{N} a_{i_t i_{t+1}} b_{i_{t+1}}(j_{t+1}) \beta_{t+1}(i_{t+1}),\ 0 \le i_t \le N, t = 0, \ldots, T-1 \tag{2.12}$$

Dokaz.

$$\begin{aligned}
\beta_t(i_t) &= \mathbb{P}_\lambda\left(Y_{t+1} = v_{j_{t+1}}, Y_{t+2} = v_{j_{t+2}}, \ldots, Y_T = v_{j_T} \mid X_t = s_{i_t}\right)\\
&= \sum_{i_{t+1}=0}^{N} \mathbb{P}_\lambda\left(Y_{t+1} = v_{j_{t+1}}, Y_{t+2} = v_{j_{t+2}}, \ldots, Y_T = v_{j_T}, X_{t+1} = s_{i_{t+1}} \mid X_t = s_{i_t}\right)\\
&= \sum_{i_{t+1}=0}^{N} \mathbb{P}_\lambda\left(Y_{t+1} = v_{j_{t+1}}, Y_{t+2} = v_{j_{t+2}}, \ldots, Y_T = v_{j_T}, X_{t+1} = s_{i_{t+1}}, X_t = s_{i_t}\right) \cdot \frac{1}{\mathbb{P}_\lambda\left(X_t = s_{i_t}\right)}\\
&= \sum_{i_{t+1}=0}^{N} \mathbb{P}_\lambda\left(Y_{t+1} = v_{j_{t+1}} \mid Y_{t+2} = v_{j_{t+2}}, \ldots, Y_T = v_{j_T}, X_{t+1} = s_{i_{t+1}}, X_t = s_{i_t}\right)\\
&\cdot \mathbb{P}_\lambda\left(Y_{t+2} = v_{j_{t+2}}, \ldots, Y_T = v_{j_T}, X_{t+1} = s_{i_{t+1}} \mid X_t = s_{i_t}\right)\\
&= \sum_{i_{t+1}=0}^{N} \mathbb{P}_\lambda\left(Y_{t+1} = v_{j_{t+1}} \mid X_{t+1} = s_{i_{t+1}}\right) \cdot \mathbb{P}_\lambda\left(X_{t+1} = s_{i_{t+1}} \mid X_t = s_{i_t}\right)\\
&\cdot \mathbb{P}_\lambda\left(Y_{t+2} = v_{j_{t+2}}, \ldots, Y_T = v_{j_T} \mid X_{t+1} = s_{i_{t+1}}, X_t = s_{i_t}\right)\\
&= \sum_{i_{t+1}=0}^{N} \mathbb{P}_\lambda\left(Y_{t+1} = v_{j_{t+1}} \mid X_{t+1} = s_{i_{t+1}}\right) \cdot \mathbb{P}_\lambda\left(Y_{t+2} = v_{j_{t+2}}, \ldots, Y_T = v_{j_T} \mid X_{t+1} = s_{i_{t+1}}\right)\\
&\cdot \mathbb{P}_\lambda\left(X_{t+1} = s_{i_{t+1}} \mid X_t = s_{i_t}\right)\\
&= \sum_{i_{t+1}=0}^{N} b_{i_{t+1}}(j_{t+1}) a_{i_t i_{t+1}} \beta_{t+1}(i_{t+1})
\end{aligned}$$

□

Backward algoritam sastoji se od tri koraka:

1. Inicijalizacija:
$$\beta_T(i_T) = 1$$

2. Rekurzija:
$$\beta_t(i_t) = \sum_{i_{t+1}=0}^{N} a_{i_t i_{t+1}} b_{i_{t+1}}(j_{t+1})\beta_{t+1}(i_{t+1}),\ t = 0, \dots, T-1,\ 0 \leq i_t \leq N$$

3. Kraj:
$$\begin{aligned}
\mathbb{P}_\lambda(\mathbf{Y}_T = \mathbf{v}_T) &= \sum_{i_0=0}^{N} \mathbb{P}_\lambda(\mathbf{Y}_T = \mathbf{v}_T, X_0 = s_{i_0}) \\
&= \sum_{i_0=0}^{N} \mathbb{P}_\lambda(\mathbf{Y}_T = \mathbf{v}_T \mid X_0 = s_{i_0}) \cdot \mathbb{P}_\lambda(X_0 = s_{i_0}) \\
&= \sum_{i_0=0}^{N} \mathbb{P}_\lambda\left(Y_1 = v_{j_1}, \dots, Y_T = v_{j_T} \mid X_0 = s_{i_0}\right) \cdot \mathbb{P}_\lambda\left(Y_0 = v_{j_0} \mid X_0 = s_{i_0}\right) \\
&\cdot \mathbb{P}_\lambda(X_0 = s_{i_0}) \\
&= \sum_{i_0=0}^{N} \beta_0(i_0) b_{i_0}(j_0)\pi_{i_0}
\end{aligned}$$

Kao i kod *forward* metode, složenost *backward* metode je reda $(N+1)^2(T+1)$.

2.2.2 Problem dekodiranja

Pretpostavimo da je dan model $\lambda = (A, B, \pi)$ i niz opažanja $v_{j_0}, v_{j_1}, \dots, v_{j_T} \in V$. Zanima nas optimalan niz skrivenih stanja koji je emitirao zadana opažanja. Postoje različiti kriteriji optimalnosti. Jedan od načina biranja optimalnog niza je da za svako opažanje pronađemo skriveno stanje koje ima najveću vjerojatnost emitiranja tog opažanja i sva takva stanja spojimo u jedan niz. No ovaj pristup može dovesti do niza stanja koji je nemoguć, odnosno za koji je neka prijelazna vjerojatnost jednaka 0. Umjesto toga,pronaći ćemo niz skrivenih stanja $\mathbf{X}_T = \mathbf{s}_T$ koji će maksimizirati vjerojatnost $\mathbb{P}_\lambda(\mathbf{X}_T = \mathbf{s}_T \mid \mathbf{Y}_T = \mathbf{v}_T)$, za sve moguće nizove $\mathbf{s}_T$. To je ekvivalentno računanju $\mathbb{P}_\lambda(\mathbf{X}_T = \mathbf{s}_T, \mathbf{Y}_T = \mathbf{v}_T)$, jer je $\mathbb{P}_\lambda(\mathbf{Y}_T = \mathbf{v}_T)$ fiksirano. Za rješenje ovog problema koristimo algoritam baziran na metodama dinamičkog programiranja, takozvani Viterbijev algoritam. Najprije definiramo pomoćnu varijablu koja nam daje najveću vjerojatnost koju parcijalni niz opažanja i niz stanja do trenutka t može imati, ako je trenutno stanje i.

$$\delta_t(i_t) = \max_{0 \leq i_0, i_1, \ldots, i_{t-1} \leq N} \mathbb{P}_\lambda \left(X_0 = s_{i_0}, \ldots, X_{t-1} = s_{i_{t-1}}, X_t = s_{i_t}, Y_0 = v_{j_0}, \ldots, Y_t = v_{j_t} \right)$$

Propozicija 2.2.5. *Vrijedi*

$$\delta_{t+1}(i_{t+1}) = \max_{0 \leq i_t \leq N} [\delta_t(i_t) a_{i_t i_{t+1}}] \, b_{i_{t+1}}(j_{t+1}), \; t = 1, \ldots, T, 0 \leq i_{t+1} \leq N$$

Dokaz.

$$\begin{aligned}
\delta_{t+1}(i_{t+1}) &= \max_{0 \leq i_0, i_1, \ldots, i_t \leq N} \mathbb{P}_\lambda \left(X_0 = s_{i_0}, \ldots, X_t = s_{i_t}, X_{t+1} = s_{i_{t+1}}, Y_0 = v_{j_0}, \ldots, Y_{t+1} = v_{j_{t+1}} \right) \\
&= \max_{0 \leq i_0, i_1, \ldots, i_t \leq N} \mathbb{P}_\lambda \left(Y_{t+1} = v_{j_{t+1}} \mid X_0 = s_{i_0}, \ldots, X_{t+1} = s_{i_{t+1}}, Y_0 = v_{j_0}, \ldots, Y_t = v_{j_t} \right) \\
&\cdot \mathbb{P}_\lambda \left(X_0 = s_{i_0}, \ldots, X_t = s_{i_t}, X_{t+1} = s_{i_{t+1}}, Y_0 = v_{j_0}, \ldots, Y_t = v_{j_t} \right) \\
&= \max_{0 \leq i_0, i_1, \ldots, i_t \leq N} \mathbb{P}_\lambda \left(Y_{t+1} = v_{j_{t+1}} \mid X_{t+1} = s_{i_{t+1}} \right) \\
&\cdot \mathbb{P}_\lambda \left(X_{t+1} = s_{i_{t+1}} \mid X_0 = s_{i_0}, \ldots, X_t = s_{i_t}, Y_0 = v_{j_0}, \ldots, Y_t = v_{j_t} \right) \\
&\cdot \mathbb{P}_\lambda \left(X_0 = s_{i_0}, \ldots, X_t = s_{i_t}, Y_0 = v_{j_0}, \ldots, Y_t = v_{j_t} \right) \\
&= \mathbb{P}_\lambda \left(Y_{t+1} = v_{j_{t+1}} \mid X_{t+1} = s_{i_{t+1}} \right) \max_{0 \leq i_0, i_1, \ldots, i_t \leq N} \mathbb{P}_\lambda \left(X_{t+1} = s_{i_{t+1}} \mid X_t = s_{i_t} \right) \\
&\cdot \mathbb{P}_\lambda \left(X_0 = s_{i_0}, \ldots, X_t = s_{i_t}, Y_0 = v_{j_0}, \ldots, Y_t = v_{j_t} \right) \\
&= b_{i_{t+1}}(j_{t+1}) \max_{0 \leq i_t \leq N} \max_{0 \leq i_0, i_1, \ldots, i_{t-1} \leq N} a_{i_t i_{t+1}} \mathbb{P}_\lambda \left(X_0 = s_{i_0}, \ldots, X_t = s_{i_t}, Y_0 = v_{j_0}, \ldots, Y_t = v_{j_t} \right) \\
&= b_{i_{t+1}}(j_{t+1}) \max_{0 \leq i_t \leq N} [\delta_t(i_t) a_{i_t i_{t+1}}]
\end{aligned}$$

□

Kako bismo pronašli niz skrivenih stanja, pamtimo argument koji maksimizira $\delta_t(i_t)$, za svaki t i i_t i spremamo u polje $\psi_t(i_t)$.

Viterbijev algoritam sastoji se od tri koraka:

1. Inicijalizacija:

$$\delta_0(i_0) = \pi_{i_0} b_{i_0}(j_0), \; 0 \leq i_0 \leq N$$
$$\psi_0(i_0) = 0$$

2. Rekurzija:

$$\delta_t(i_t) = \max_{0 \leq i_{t-1} \leq N} [\delta_{t-1}(i_{t-1}) a_{i_{t-1} i_t}] \, b_{i_t}(j_t), \; t = 1, \ldots, T, 0 \leq i_t \leq N$$

$$\psi_t(i_t) = \underset{0 \leq i_{t-1} \leq N}{arg \; max} \; [\delta_{t-1}(i_{t-1}) a_{i_{t-1} i_t}], \; t = 1, \ldots, T, 0 \leq i_t \leq N$$

3. Kraj:

$$\delta^* = \max_{0 \le i_T \le N} \delta_T(i_T)$$

$$\psi^* = arg\ max_{0 \le i_T \le N} \delta_T(i_T)$$

Konačno, imamo niz skrivenih stanja:

$$i_T = \psi^*, i_t = \psi_{t+1}(i_{t+1}),\ t = T-1, T-2, \dots, 0$$

Ideja Viterbijevog algoritma slična je *forward* proceduri, razlika je što znak sumacije mijenjamo maksimumom. Spomenimo još jednu metodu koja koristi ideju *backward* metode, $\delta - \theta$ algoritam. Prednost je ušteda vremena jer tražimo maksimum s oba kraja istovremeno. Definiramo pomoćnu varijablu

$$\theta_t(i_t) = \max_{0 \le i_{t+1}, \dots, i_T \le N} \mathbb{P}_\lambda \left(X_{t+1} = s_{i_{t+1}}, \dots, X_T = s_{i_T}, Y_{t+1} = v_{j_{t+1}}, \dots, Y_T = v_{j_T} \mid X_t = s_{i_t} \right)$$

$\delta - \theta$ algoritam sastoji se od tri koraka:

1. Inicijalizacija:

$$\delta_0(i_0) = \pi_{i_0} b_{i_0}(j_0),\ 0 \le i_0 \le N$$

$$\theta_T(i_T) = 1,\ 0 \le i_T \le N$$

2. Rekurzija:

$$\delta_t(i_t) = \max_{0 \le i_{t-1} \le N} [\delta_{t-1}(i_{t-1}) a_{i_{t-1} i_t}]\, b_{i_t}(j_t),\ t = 1, \dots, T, 0 \le i_t \le N$$

$$\psi_t(i_t) = arg\ max_{0 \le i_{t-1} \le N} [\delta_{t-1}(i_{t-1}) a_{i_{t-1} i_t}],\ t = 1, \dots, T, 0 \le i_t \le N$$

$$\theta_t(i_t) = \max_{0 \le i_{t+1} \le N} \theta_{t+1}(i_{t+1}) a_{i_t i_{t+1}} b_{i_{t+1}}(j_{t+1}),\ t = 0, \dots, T-1, 0 \le i_t \le N$$

$$\varphi_t(i_t) = arg\ max_{0 \le i_{t+1} \le N} \theta_{t+1}(i_{t+1}) a_{i_t i_{t+1}} b_{i_{t+1}}(j_{t+1}),\ t = 0, \dots, T-1, 0 \le i_t \le N$$

3. Kraj:

$$\max_{0 \le i_0, \dots, i_T \le N} \mathbb{P}_\lambda \left(X_0 = s_{i_0}, \dots, X_T = s_{i_T}, Y_0 = v_{j_0}, \dots, Y_T = v_{j_T} \right) = \max_{0 \le i_t \le N} [\delta_t(i_t) \theta_t(i_t)]$$

Definiramo $i_t^* = arg\ max_{0 \le i_t \le N} [\delta_t(i_t)\theta_t(i_t)]$ pa je najvjerojatniji niz skrivenih stanja dan s:

$$i_t = \begin{cases} \phi_{t+1}(i_{t+1}) & \text{od } 0 \text{ do } t-1 \\ i_t^* & \text{u trenutku } t \\ \varphi_{t-1}(i_{t-1}) & \text{od } t+1 \text{ do } T \end{cases}$$

Primjer 2.2.6. *Vratimo se na primjer 2.2.3. Pretpostavimo da sada imamo zadan opažanja (šetnja, pečenje, bicikliranje, sviranje). Ako znamo da se Ivana bavila tim hobijima, možemo li zaključiti kakvo je bilo vrijeme ta četiri dana?*
Koristeći Viterbijem algoritam imamo:

1. *Inicijalizacija:*

$$\delta_0(\textit{sunčano}) = \pi_{\textit{sunčano}} b_{\textit{šetnja}}(\textit{sviranje}) = 0.6 \cdot 0.2 = 0.12$$

$$\psi_0(\textit{sunčano}) = 0$$

$$\delta_0(\textit{oblačno}) = \pi_{\textit{oblačno}} b_{\textit{šetnja}}(\textit{sviranje}) = 0.4 \cdot 0.2 = 0.08$$

$$\psi_0(\textit{oblačno}) = 0$$

2. *Rekurzija:*

   $t = 1$*:*

$$\begin{aligned}\delta_1(\textit{sunčano}) = \max[&\delta_0(\textit{sunčano}) \cdot a_{\textit{sunčano,sunčano}} \cdot b_{\textit{sunčano}}(\textit{pečenje}),\\ &\delta_0((\textit{oblačno})) \cdot a_{\textit{oblačno,sunčano}} \cdot b_{\textit{sunčano}}(\textit{pečenje})]\\ =\ &\max[0.12 \cdot 0.8 \cdot 0.1, 0.08 \cdot 0.4 \cdot 0.1]\\ =\ &0.0096\end{aligned}$$

$$\psi_1(\textit{sunčano}) = \textit{sunčano}$$

$$\begin{aligned}\delta_1(\textit{oblačno}) = \max[&\delta_0(\textit{sunčano}) \cdot a_{\textit{sunčano,oblačno}} \cdot b_{\textit{oblačno}}(\textit{pečenje}),\\ &\delta_0(\textit{oblačno}) \cdot a_{\textit{oblačno,oblačno}} \cdot b_{\textit{oblačno}}(\textit{pečenje})]\\ =\ &\max\,[0.12 \cdot 0.2 \cdot 0.45, 0.08 \cdot 0.6 \cdot 0.45]\\ =\ &0.0216\end{aligned}$$

$$\psi_1(\textit{oblačno}) = \textit{oblačno}$$

   $t = 2$*:*

$$\begin{aligned}\delta_2(\textit{sunčano}) = \max[&\delta_1(\textit{sunčano}) \cdot a_{\textit{sunčano,sunčano}} \cdot b_{\textit{sunčano}}(\textit{bicikliranje}),\\ &\delta_1(\textit{oblačno}) \cdot a_{\textit{oblačno,sunčano}} \cdot b_{\textit{sunčano}}(\textit{bicikliranje})]\\ =\ &\max\,[0.0096 \cdot 0.8 \cdot 0.3, 0.0216 \cdot 0.4 \cdot 0.3]\\ =\ &0.002592\end{aligned}$$

$$\psi_2(\textit{sunčano}) = \textit{oblačno}$$

$$\begin{aligned}\delta_2(obla\check{c}no) &= \max[\delta_1(sun\check{c}ano)\cdot a_{sun\check{c}ano,obla\check{c}no}\cdot b_{obla\check{c}no}(bicikliranje),\\ &\quad \delta_1(obla\check{c}no)\cdot a_{obla\check{c}no,obla\check{c}no}\cdot b_{obla\check{c}no}(bicikliranje)]\\ &= \max\,[0.0096\cdot 0.2\cdot 0.05, 0.0216\cdot 0.6\cdot 0.05]\\ &= 0.000648\end{aligned}$$

$$\psi_2(obla\check{c}no) = obla\check{c}no$$

$t = 3$:

$$\begin{aligned}\delta_3(sun\check{c}ano) &= \max[\delta_2(sun\check{c}ano)\cdot a_{sun\check{c}ano,sun\check{c}ano}\cdot b_{sun\check{c}ano}(sviranje),\\ &\quad \delta_2(obla\check{c}no)\cdot a_{obla\check{c}no,sun\check{c}ano}\cdot b_{sun\check{c}ano}(sviranje)]\\ &= \max\,[0.002592\cdot 0.8\cdot 0.4, 0.000648\cdot 0.4\cdot 0.4]\\ &= 0.00082944\end{aligned}$$

$$\psi_3(sun\check{c}ano) = sun\check{c}ano$$

$$\begin{aligned}\delta_3(obla\check{c}no) &= \max[\delta_2((sun\check{c}ano))\cdot a_{sun\check{c}ano,obla\check{c}no}\cdot b_{obla\check{c}no}(sviranje),\\ &\quad \delta_2((obla\check{c}no))\cdot a_{obla\check{c}no,obla\check{c}no}\cdot b_{obla\check{c}no}(sviranje)]\\ &= \max\,[0.002592\cdot 0.2\cdot 0.3, 0.000648\cdot 0.6\cdot 0.3]\\ &= 0.00015552\end{aligned}$$

$$\psi_3(obla\check{c}no) = sun\check{c}ano$$

3. *Kraj:*

$$\phi^* = arg\ max\ [\delta_3(sun\check{c}ano), delta_3(obla\check{c}no)] = sun\check{c}ano$$

Dobili smo polja $\psi(sun\check{c}ano)$ = [0, *sunčano, oblačno, sunčano, sunčano*] *i* $\psi(obla\check{c}no)$ = [0, *oblačno, oblačno, sunčano, sunčano*]. *Sada možemo pronaći traženi niz skrivenih stanja:*

$$\begin{aligned}i_3 &= sun\check{c}ano\\ i_2 &= \psi_3(sun\check{c}ano) = sun\check{c}ano\\ i_1 &= \psi_2(sun\check{c}ano) = obla\check{c}no\\ i_0 &= \psi_1(obla\check{c}no) = obla\check{c}no\end{aligned}$$

Dakle dobilo smo niz skrivenih stanja (oblačno, oblačno, sunčano, sunčano).

2.2.3 Problem učenja

Pretpostavimo da je dan model λ i niz opažanja $v_{j_0}, v_{j_1}, \dots, v_{j_T} \in V$. Želimo procijeniti parametre modela (A, B, π) koji maksimiziraju vjerojatnost $\mathbb{P}_\lambda(\mathbf{Y}_T = \mathbf{v}_T)$. Za rješenje ovog problema koristimo iterativni *Baum - Welch* algoritam poznat još i kao *Forward - Backward* algoritam. Uz *forward* i *backward* varijable, definirat ćemo još dvije pomoćne varijable

$$\xi_t(i_t, i_{t+1}) = \mathbb{P}_\lambda(X_t = s_{i_t}, X_{t+1} = s_{i_{t+1}} \mid \mathbf{Y}_T = \mathbf{v}_T)$$

$$\gamma_t(i_t) = \mathbb{P}_\lambda(X_t = s_{i_t} \mid \mathbf{Y}_T = \mathbf{v}_T)$$

Koristeći *forward* i *backward* varijable, možemo ih zapisati na sljedeći način

$$\begin{aligned}
\xi_t(i_t, i_{t+1}) &= \frac{\mathbb{P}_\lambda(X_t = s_{i_t}, X_{t+1} = s_{i_{t+1}}, \mathbf{Y}_T = \mathbf{v}_T)}{\mathbb{P}_\lambda(\mathbf{Y}_T = \mathbf{v}_T)} \\
&= \frac{\alpha_t(i_t) a_{i_t i_{t+1}} b_{i_{t+1}}(j_{t+1}) \beta_{t+1}(i_{t+1})}{\mathbb{P}_\lambda(\mathbf{Y}_T = \mathbf{v}_T)} \\
&= \frac{\alpha_t(i_t) a_{i_t i_{t+1}} b_{i_{t+1}}(j_{t+1}) \beta_{t+1}(i_{t+1})}{\sum_{i_t=0}^N \sum_{i_{t+1}=0}^N \alpha_t(i_t) a_{i_t i_{t+1}} b_{i_{t+1}}(j_{t+1}) \beta_{t+1}(i_{t+1})}
\end{aligned}$$

$$\gamma_t(i_t) = \frac{\mathbb{P}_\lambda(X_t = s_{i_t}, \mathbf{Y}_T)}{\mathbb{P}_\lambda(\mathbf{Y}_T = \mathbf{v}_T)} = \frac{\alpha_t(i_t)\beta_t(i_t)}{\sum_{i_t=0}^N \alpha_t(i_t)\beta_t(i_t)},$$

pa je veza između $\xi_t(i_t, i_{t+1})$ i $\gamma_t(i_t)$ dana s

$$\gamma_t(i_t) = \sum_{t_{t+1}=0}^{N} \xi_t(i_t, i_{t+1}).$$

Sada možemo opisati *Baum-Welch* postupak. Iz početnog modela odredimo α i β, a zatim ξ i γ. Sljedeće jednadžbe zovemo formule reestimacije i koristimo ih za nove procjene parametara modela:

$$\begin{aligned}
\overline{\pi_i} &= \gamma_0(i) \\
\overline{a_{ij}} &= \frac{\sum_{t=0}^{T-1} \xi_t(i, j)}{\sum_{t=0}^{T-1} \gamma_t(i)} \\
\overline{b_j(k)} &= \frac{\sum_{t=0, Y_t = v_k}^{T-1} \gamma_t(j)}{\sum_{t=0}^{T-1} \gamma_t(j)},
\end{aligned}$$

gdje $\sum_{t=0}^{T-1} \xi_t(i, j)$ predstavlja očekivani broj tranzicija iz stanja s_i u s_j, $\sum_{t=0}^{T-1} \gamma_t(i)$ očekivani broj posjeta stanja s_i u trenutku t=0, a $\overline{\pi_i}$ očekivani broj puta u stanju s_i. Dobijemo novi model te iterativno ponavljamo postupak poboljšavajući traženu vjerojatnost dok ne dođemo do nekog kriterija zaustavljanja.

Primjer 2.2.7. *Ponovno se vratimo na primjer 2.2.3. Želimo prilagoditi parametre modela kako bi maksimizirali vjerojatnost da je zadani model emitirao niz opažanja (sviranje, pečenje, šetnja, bicikliranje). Prvo računamo backward varijable.*

1. *Inicijalizacija:*

$$\beta_3(sunčano) = 1$$

$$\beta_3(oblačno) = 1$$

2. *Rekurzija:*

$t = 2$*:*

$$\begin{aligned}\beta_2(sunčano) &= a_{sunčano,sunčano} b_{sunčano}(bicikliranje)\beta_3(sunčano) + \\ &\quad a_{sunčano,oblačno} b_{oblačno}(bicikliranje)\beta_3(oblačno) \\ &= 0.8 \cdot 0.3 \cdot 1 + 0.2 \cdot 0.05 \cdot 1 = 0.25\end{aligned}$$

$$\begin{aligned}\beta_2(oblačno) &= a_{oblačno,sunčano} b_{sunčano}(bicikliranje)\beta_3(sunčano) + \\ &\quad a_{oblačno,oblačno} b_{oblačno}(bicikliranje)\beta_3(oblačno) \\ &= 0.4 \cdot 0.3 \cdot 1 + 0.6 \cdot 0.05 \cdot 1 = 0.15\end{aligned}$$

$t = 1$*:*

$$\begin{aligned}\beta_1(sunčano) &= a_{sunčano,sunčano} b_{sunčano}(šetnja)\beta_2(sunčano) + \\ &\quad a_{sunčano,oblačno} b_{oblačno}(šetnja)\beta_2(oblačno) \\ &= 0.8 \cdot 0.2 \cdot 0.25 + 0.2 \cdot 0.2 \cdot 0.15 = 0.046\end{aligned}$$

$$\begin{aligned}\beta_1(oblačno) &= a_{oblačno,sunčano} b_{sunčano}(šetnja)\beta_2(sunčano) + \\ &\quad a_{oblačno,oblačno} b_{oblačno}(šetnja)\beta_2(oblačno) \\ &= 0.4 \cdot 0.2 \cdot 0.25 + 0.6 \cdot 0.2 \cdot 0.15 = 0.038\end{aligned}$$

$t = 0$*:*

$$\begin{aligned}\beta_0(sunčano) &= a_{sunčano,sunčano} b_{sunčano}(pečenje)\beta_1(sunčano) + \\ &\quad a_{sunčano,oblačno} b_{oblačno}(pečenje)\beta_1(oblačno) \\ &= 0.8 \cdot 0.1 \cdot 0.046 + 0.2 \cdot 0.45 \cdot 0.038 = 0.0071\end{aligned}$$

$$\begin{aligned}\beta_0(oblačno) &= a_{oblačno,sunčano} b_{sunčano}(pečenje)\beta_1(sunčano) + \\ &\quad a_{oblačno,oblačno} b_{oblačno}(pečenje)\beta_1(oblačno) \\ &= 0.4 \cdot 0.1 \cdot 0.046 + 0.6 \cdot 0.45 \cdot 0.038 = 0.0121\end{aligned}$$

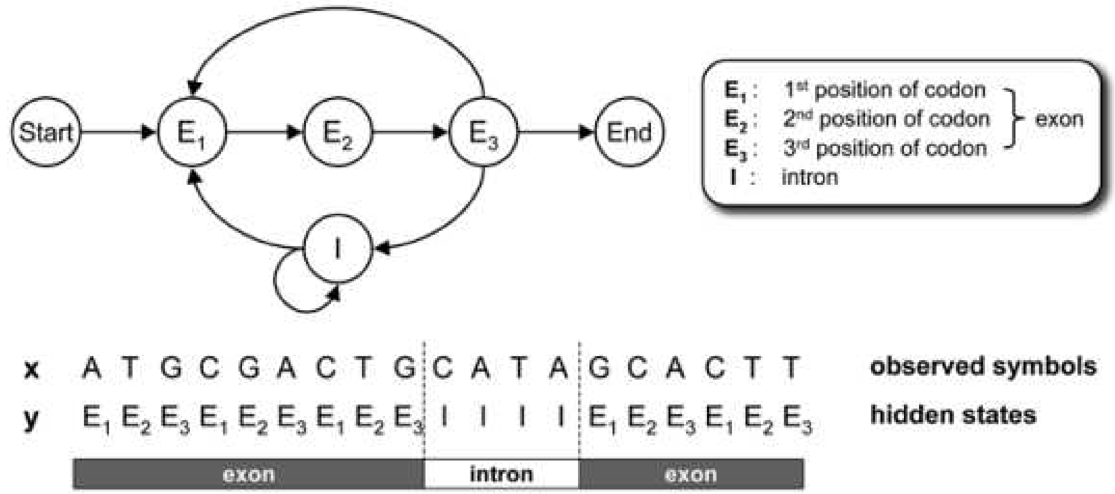
3. Kraj:

$$
\begin{aligned}
&\mathbb{P}_\lambda\,(\textit{sviranje, pečenje, šetnja, bicikliranje})\\
&= \beta_0(\textit{sunčano})b_{\textit{sunčano}}(\textit{sviranje})\pi_{\textit{sunčano}} + \beta_0(\textit{oblačno})b_{\textit{oblačno}}(\textit{sviranje})\pi_{\textit{oblačno}}\\
&= 0.0071 \cdot 0.4 \cdot 0.6 + 0.0121 \cdot 0.3 \cdot 0.4 = 0.003156
\end{aligned}
$$

Dobili smo isto rješenje kao u primjeru 2.2.3. Preostaje izračunati pomoćne varijable $\xi_t(i, j)$ $\gamma_t(i)$, *za skrivena stanja i i j i* $0 \leq t \leq 3$, *koristeći zapis pomoću forward i backward varijabli. Nove parametre modela dobijemo jednadžbama reestimacije, a računanjem vjerojatnosti da je novi model emitirao zadani niz opažanja možemo provjeriti jesmo li dobili bolji model, odnosno veću vjerojatnost.*

2.3 Primjene

Primjer 2.3.1. *(Primjena u biologiji [5]) Skriveni Markovljevi modeli mogu se koristiti za modeliranje gena za kodiranje proteina u eukariota. Različiti dijelovi gena koji sadrže kodirajuće informacije za proteine pokazuju preferenciju prema određenim kodonima. Ova nejednolika upotreba kodona rezultira različitim statistikama simbola na različitim pozicijama kodona. Ove razlike nisu prisutne u intronima, koji ne sudjeluju u prijenosu informacija za sintezu aminokiselina. Stoga je ključno uključiti ove statistike kodona prilikom modeliranja gena koji kodiraju proteine te u razvoju alata za identifikaciju gena. Prikazani HMM model za eukariotske gene nastoji zabilježiti statističke razlike između egzona (kodirajućih sekvenci koje sadrže eukariotski gen) i introna (nekodirajućeg DNA). Skriveni Markovljev model ima četiri stanja, pri čemu se stanja označena s* E_1, E_2, E_3 *koriste za modeliranje baznih statistika u egzonima. Svako stanje* E_k *koristi različite vjerojatnosti emisije kako bi odrazilo statistike simbola na k-tom položaju kodona. Stanje I se koristi za modeliranje baznih statistika introna. Važno je napomenuti da ovaj Markovljev model može predstavljati gene s više egzona, gdje respektivni egzoni mogu imati različit broj kodona, a introni različite duljine. Dakle, ako poznajemo strukturu i karakteristike određenog biološkog niza, lako možemo konstruirati Markovljev model i koristiti ga za analizu novog niza opservacija. Na primjer, pretpostavimo da imamo novi DNA niz* $\mathbf{v} = v_1 \ldots v_{19}$=*ATGCGACTGCATAGCACTT. Kako ćemo znati je li ovaj niz kodirajući gen? Ako pretpostavimo da je* $\mathbf{v}$ *protein za kodiranje gena, kako možemo predvidjeti lokaciju egzona i introna u danom nizu? Odgovor na prvo pitanje dobivamo računanjem vjerojatnosti da je taj niz generiran konstruiranim modelom, dakle riječ je o problemu evaluacije. Velika vjerojatnost može implicirati da je zadani DNA niz kodirajući gen. Inače možemo zaključiti da je malo vjerojatno da je niz kodirajući gen, s obzirom da ne zadovoljava statistička obilježja koja su tipična za takve gene. Odgovor na drugo pitanje, kako predvidjeti lokaciju egzona i introna u danom nizu, uključuje predviđanje niza stanja koji* *najbolje opisuje dane opservacije, što predstavlja problem dekodiranja. Na primjer, pretpostavimo da je optimalni niz stanja y prikazan na slici što implicira da prvih devet baza odgovara prvog egzonu, sljedeća četiri pripadaju intronu, a zadnjih šest drugom egzonu. Dakle, skriveni Markovljev model pruža korisne alate za analizu bioloških nizova.*



Slika 2.2: Jednostavni HMM za modeliranje gena. Slika preuzeta iz [5].

Primjer 2.3.2. *(Obrada prirodnog jezika) Obrada prirodnog jezika grana je umjetne inteligencije koja računalima omogućava razumijevanje teksta i izgovorenih riječi, odnosno glavna ideja je naučiti računalo da razumije puno značenje jezika kojim ljudi komuniciraju, uključujući namjeru i ton. Važan dio obrade jezika je označavanje vrsta riječi ili POS označavanje. To je postupak u kojem svakoj riječi pridružujemo odgovarajuću oznake vrste riječi s obzirom na njeno značenje i vezu s ostalim riječima u rečenici. Na primjer, u rečenicama "Danas sam naporno radio." i "U autu uvijek slušam radio." riječ radio je glagol, odnosno imenica. Postoje razne tehnike označavanja vrste riječi, a najjednostavija je ona koja daje najčešće korištenu oznaku za određenu riječ u već označenim podacima za treniranje modela i koristi tu informaciju za označavanje riječu u neoznačenom tekstu. No ova metoda nekad može dati nizove oznaka koji nisu prihvatljivi po gramatičkim pravilima jezika. Zato koristimo stohastički pristup u kojem izračunavamo vjerojatnost različitih nizova oznaka koji su mogući u rečenici i oznake dodjeljujemo iz niza s najvećom vjerojatnosti. U takvom pristupu veliku ulogu imaju skriveni Markovljevi modeli. Skrivena stanja su oznake vrste riječi, opažena stanja riječi u rečenici, primjer tranzicijske vjerojatnosti je vjerojatnost da je trenutna riječ imenica ako znamo da je prethodna bila zamjenica, a primjer emisijske vjerojatnost je vjerojatnost da je riječ "radio" imenica. Želimo riješiti* *problem dekodiranja, odnosno pronaći niz oznaka koji je najvjerojatniji s obzirom na niz promatranih riječi. Pokažimo postupak na jednostavnom primjeru.*

Pretpostavimo da imamo sljedeće podatke na engleskom jeziku.

1. *Mary Jane can see Will.*
2. *Spot will see Mary.*
3. *Will Jane spot Mary?*
4. *Mary will pat Spot.*

Tablica prikazuje riječi i frekvenciju oznaka vrste riječi.

	IMENICA	MODALNI GLAGOL	GLAGOL
MARY	4	0	0
JANE	2	0	0
WILL	1	3	0
SPOT	2	0	1
CAN	0	1	0
SEE	0	0	2
PAT	0	0	1

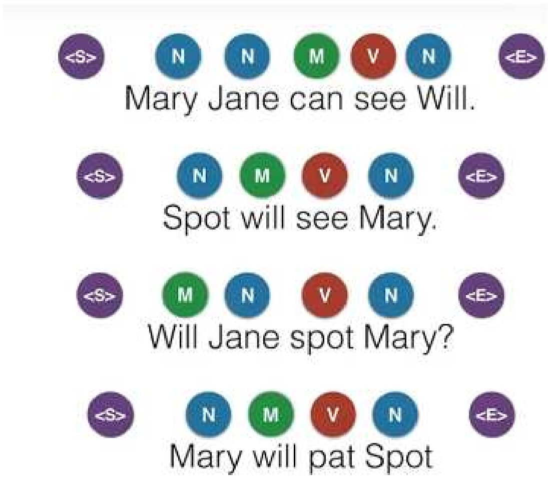
Tablica 2.1: Oznake vrste riječi

Na primjer, riječ "will" se pojavljuje kao imenica u prvoj rečenici, a kao modalni glagol u drugoj, trećoj i četvrtoj rečenici. U sljedećem koraku ćemo svaki stupac podijeliti s ukupnim brojem pojavljivanja te oznake kako bismo dobili emisijske vjerojatnosti. Iz tablice vidimo da je vjerojatnost da je riječ "Mary" imenica $\frac{4}{9}$*, a vjerojatnost da je glagol* 0*.*

	IMENICA	MODALNI GLAGOL	GLAGOL
MARY	4/9	0	0
JANE	2/9	0	0
WILL	1/9	3/4	0
SPOT	2/9	0	1/4
CAN	0/9	1/4	0
SEE	0/9	0	2/4
PAT	0/9	0	1/4

Tablica 2.2: Tablica opservacijskih vjerojatnosti

Preostaje nam izračunati prijelazne vjerojatnosti. Uvest ćemo dodatne oznake START i END. Sada naši podaci izgledaju ovako:



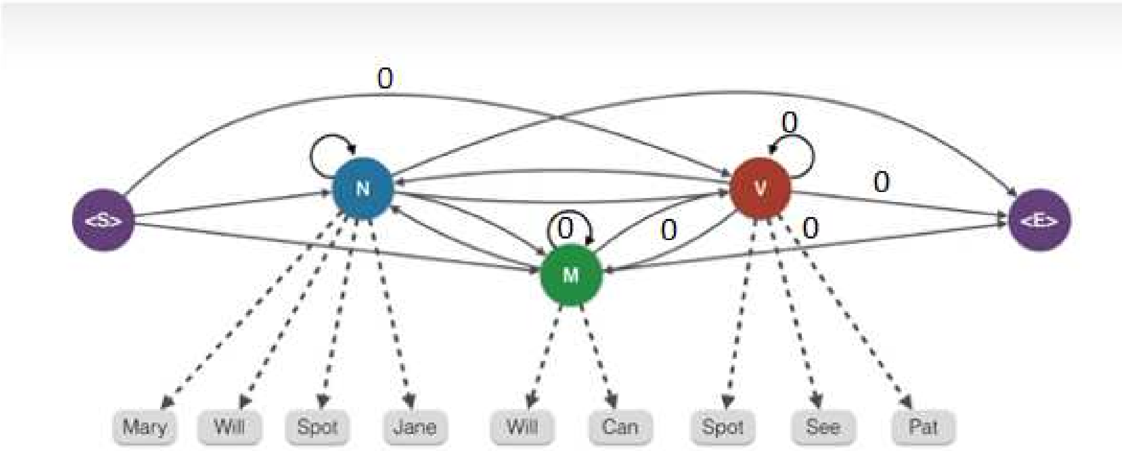
Slika 2.3: Novi podaci. Slika preuzeta iz [6].

Konstruiramo novu tablicu u kojoj upisujemo frekvenciju pojavljivanja pojedine oznake iz svakog stupca nakon oznaka iz svakog retka podijeljenju s brojem ponavljanja oznake iz retka. Na primjer, imenica slijedi nakon oznake START u prvoj, drugoj i četvrtoj rečenici pa broj 3 *dijelimo s brojem ponavljanja oznake START i dobivamo* $\frac{3}{4}$.

	IMENICA	MODALNI GLAGOL	GLAGOL	END
START	3/4	1/4	0	0
IMENICA	1/9	3/9	1/9	4/9
MODALNI GLAGOL	1/4	0	3/4	0
GLAGOL	4/4	0	0	0

Tablica 2.3: Matrica prijelaznih vjerojatnosti

Uzmimo sad novu rečenicu "Will can spot Mary." i pogledajmo sve moguće kombijacije oznaka.



Slika 2.4: Sve kombinacije. Slika preuzeta iz [6].

Pogledajmo moguće oznake riječi u rečenici "Will can spot Mary.".

1. *START→IMENICA→MODALNI GLAGOL→IMENICA→IMENICA →END*
2. *START→MODALNI GLAGOL→MODALNI GLAGOL→IMENICA→IMENICA→END*
3. *START→IMENICA→MODALNI GLAGOL→GLAGOL→IMENICA →END*
4. *START→MODALNI GLAGOL→MODALNI GLAGOL→GLAGOL→IMENICA →END*

S obzirom da je vjerojatnost da modalni glagol slijedi nakon modalnog glagola 0*, ostaju nam samo prva i treća opcija. Množenjem prijelaznih i emisijskih vjerojatnosti dobivamo:*

1. *START→IMENICA→MODALNI GLAGOL →IMENICA →IMENICA →END*

$$\frac{3}{4}\cdot\frac{1}{9}\cdot\frac{3}{9}\cdot\frac{1}{4}\cdot\frac{1}{4}\cdot\frac{2}{9}\cdot\frac{1}{9}\cdot\frac{4}{9}\cdot\frac{4}{9} = 0.00000846754$$

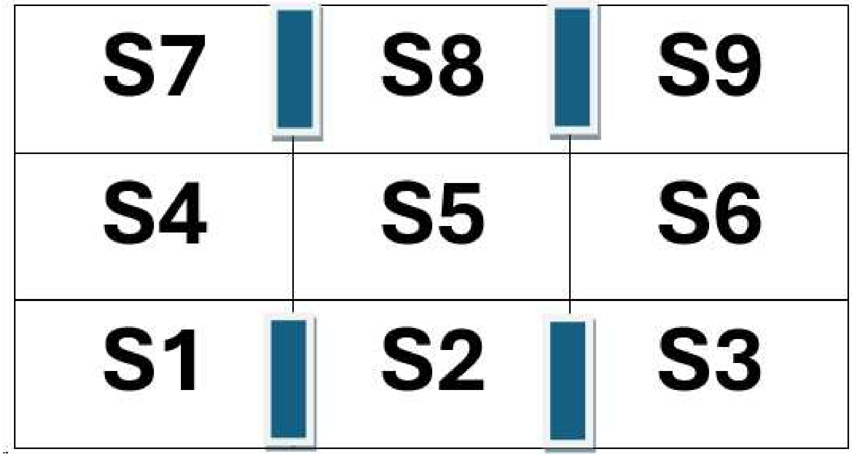
2. *START→IMENICA→MODALNI GLAGOL→GLAGOL→IMENICA→END*

$$\frac{3}{4}\cdot\frac{1}{9}\cdot\frac{3}{9}\cdot\frac{1}{4}\cdot\frac{3}{4}\cdot\frac{1}{4}\cdot\frac{4}{4}\cdot\frac{4}{9}\cdot\frac{4}{9}\cdot = 0.00025720164$$

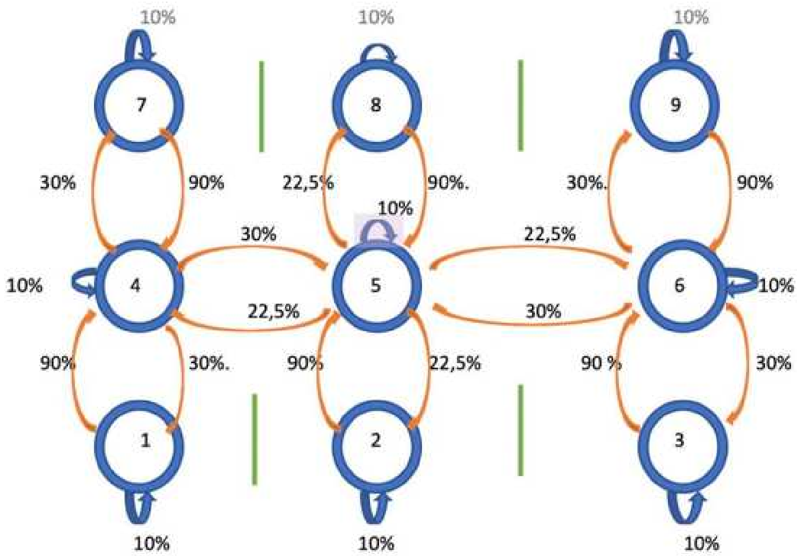
Vjerojatnost drugog niza je veća pa će model svakoj riječi iz zadane rečenice pridružiti oznake koje odgovaraju drugom nizu.

Primjer 2.3.3. *(Lokalizacija robota) 2003. godine tim znanstvenika stvorio je robota pod nazivom Groundhog koji je mogao istraživati i stvoriti mapu napuštenog rudnika koji je zbog niskih razina kisika bio opasan za ljude. Ovaj težak zadatak navigiranja robota u nepoznatoj okolini i istovremeno stvaranje mape i dalje je aktualna tema proučavanja. Pretpostavimo da imamo robota u skladištu koji je opremljen kompasom i senzorom blizine pomoću kojeg može prepoznati prepreke u četiri smjera: sjever, jug, zapad i istok, s greškom* $e = 0.25$*. Zamislimo da nemamo pristup skladištu i jedine informacije kojima raspolažemo su očitanja senzora robota. Pojednostavimo problem i pretpostavimo da se robot može kretati po skladištu koje se sastoji od devet polja, a između nekih polja nalaze se police. U svakom koraku, robot je programiran da promijeni svoju poziciju s vjerojatnosti* 0.9 *i pomakne se na slučajno odabrano susjedno mjesto. Za modeliranje ovog problema koristimo skriveni Markovljev model koji se sastoji od skrivenih stanja* $X_1, \ldots, X_9$ *koja predstavljaju lokaciju robota u skladištu i opservacija* $E_1, \ldots, E_5$ *koja predstavljaju očitanja senzora. Senzor može očitati sljedeće:*

- *SWE - vrijedi za polja* $S1, S2, S3$ *jer ta polja imaju prepreke na južnoj (S), zapadnoj (W) i istočnoj (E) strani*
- *NWE - vrijedi za polja* $S7, S8, S9$
- *E - vrijedi za polje* $S6$
- *W - vrijedi za polje* $S4$
- *No - vrijedi za polje* $S5$ *jer ono nema nikakvih prepreka*



Slika 2.5: Skladište



Slika 2.6: Grafički prikaz vjerojatnosti. Slika preuzeta iz [7].

Ako pretpostavimo da ne znamo početni položaj robota, za početnu distribuciju skrivenog procesa možemo uzeti uniformnu distribuciju pa imamo prijelaznu matricu i početnu *distribuciju:*

$$A = \begin{pmatrix} 0.1 & 0 & 0 & 0.3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 & 0.225 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.1 & 0 & 0 & 0.3 & 0 & 0 & 0 \\ 0.9 & 0 & 0 & 0.1 & 0.225 & 0 & 0.9 & 0 & 0 \\ 0 & 0.9 & 0 & 0.3 & 0.1 & 0.3 & 0 & 0.9 & 0 \\ 0 & 0 & 0.9 & 0 & 0.225 & 0.1 & 0 & 0 & 0.9 \\ 0 & 0 & 0 & 0.3 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.225 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0 & 0 & 0.1 \end{pmatrix}$$

$$\pi = \begin{pmatrix} 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 & 1/9 \end{pmatrix}$$

Izračunajmo vjerojatnost da senzor očita j*, ako se nalazimo u stanju* i*:*

$$\mathbb{P}(E = j \mid X = i) = (1 - e)^{4-d} \cdot e^d,$$

gdje d predstavlja broj signala koji su različiti od stvarnog stanja. Na primjer, ako senzor očita stanje SWE, a robot se nalazi na polju $S7$*, to znači da je detektirao zid za zapadu i policu na istoku, ali nije detektirao zid na sjeveru. Također je detektirao prepreku na jugu koje u stvarnosti nema pa dva senzora daju krivu informaciju, odnosno* $d = 2$*. Tada je* $\mathbb{P}(E = SWE \mid X = 7) = (1 - 0.25)^{4-2} \cdot 0.25^2 = 0.04$*. Sada možemo izračunati sve emisijske vjerojatnosti i dobivamo sljedeće matrice:*

$$SWE = \begin{bmatrix} 0.32 & & & & & & & & \\ & 0.32 & & & & & & & \\ & & 0.32 & & & & & & \\ & & & 0.04 & & & & & \\ & & & & 0.01 & & & & \\ & & & & & 0.04 & & & \\ & & & & & & 0.04 & & \\ & & & & & & & 0.04 & \\ & & & & & & & & 0.04 \end{bmatrix}$$

$$NEW = \begin{bmatrix} 0.04 & & & & & & & & \\ & 0.02 & & & & & & & \\ & & 0.04 & & & & & & \\ & & & 0.04 & & & & & \\ & & & & 0.0 & & & & \\ & & & & & 0.04 & & & \\ & & & & & & 0.32 & & \\ & & & & & & & 0.32 & \\ & & & & & & & & 0.32 \end{bmatrix}$$

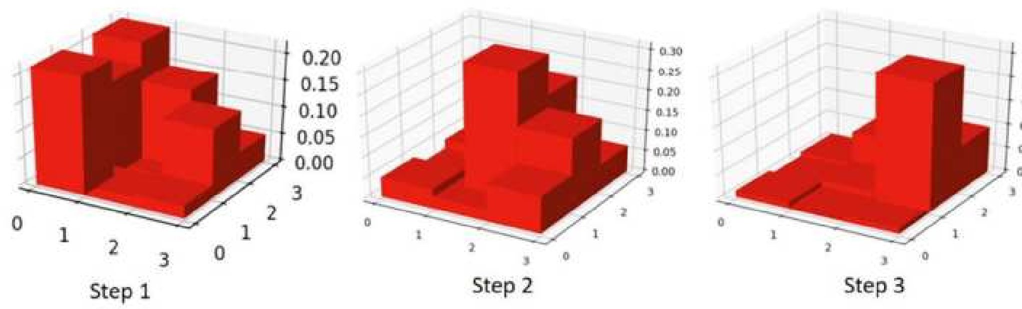
$$E = \begin{bmatrix} 0.04 & & & & & & & & \\ & 0.04 & & & & & & & \\ & & 0.04 & & & & & & \\ & & & 0.04 & & & & & \\ & & & & 0.11 & & & & \\ & & & & & 0.32 & & & \\ & & & & & & 0.04 & & \\ & & & & & & & 0.04 & \\ & & & & & & & & 0.04 \end{bmatrix}$$

$$W = \begin{bmatrix} 0.04 & & & & & & & & \\ & 0.04 & & & & & & & \\ & & 0.04 & & & & & & \\ & & & 0.32 & & & & & \\ & & & & 0.11 & & & & \\ & & & & & 0.04 & & & \\ & & & & & & 0.04 & & \\ & & & & & & & 0.04 & \\ & & & & & & & & 0.04 \end{bmatrix}$$

$$No = \begin{bmatrix} 0.01 & & & & & & & & \\ & 0.01 & & & & & & & \\ & & 0.01 & & & & & & \\ & & & 0.11 & & & & & \\ & & & & 0.32 & & & & \\ & & & & & 0.11 & & & \\ & & & & & & 0.01 & & \\ & & & & & & & 0.01 & \\ & & & & & & & & 0.01 \end{bmatrix}$$

Daljnjom analizom mogu se izračunati vjerojatnosti da se robot nalazi u nekom polju u svakom koraku. Na primjer, za prva tri koraka dobije se sljedeće:

- *U prvom koraku, robot ima vjerojatnost* 20% *da se nalazi na polju 1 ili 7.*
- *U drugom koraku ima veliku vjerojatnost da se nalazi na polju 5, s vjerojatnosti skoro* 30%*.*
- *U trećem koraku robot ima veliku vjerojatnost* (50%) *da se nalazi u stanju 6.*



Slika 2.7: Grafički prikaz vjerojatnosti za prva tri koraka. Slika preuzeta iz [7].

Koristeći senzorske podatke, možemo zaključiti najvjerojatniju lokaciju robota u svakom vremenskom trenutku.

Poglavlje 3

Markovljev proces odlučivanja

3.1 Definicija i osnovna svojstva

Markovljeve lance možemo proširiti na Markovljev proces odlučivanja koji služi za modeliranje donošenja odluka u situacijama gdje su ishodi djelomično slučajni, a djelomično pod kontrolom donositelja odluka, tj. agenta. Model uzima u obzir kako trenutačne odluke utječu na kratkoročne ishode, ali i prilike za donošenje odluka u budućnosti. Razlika između Markovljevih lanaca i procesa odlučivanja je dodatak akcija i nagrada. Akcije omogućuju agentu da odabere što učiniti sljedeće, nagrađujući ih odgovarajućom nagradom nakon promjene stanja. Preciznije, Markovljev proces odlučivanja možemo definirati na sljedeći način:

Definicija 3.1.1. *Markovljev proces odlučivanja s diskretnim vremenom je četvorka* (S, A, tr, R)*, gdje su:*

- S *konačan skup stanja* (λ, P) *- Markovljevog lanca* $X = (X_n : n \geq 0)$
- A *konačan skup akcija koje agent može poduzeti*
- tr *prijelazna funkcija, gdje s* $tr(s, a, s')$ *označavamo vjerojatnost da će agent koji izvršava akciju* a *u stanju* s *doći do stanja* s'*, tj. prijelaznu funkciju definiramo kao* $tr(s, a, s') = \mathbb{P}(X_{n+1} = s' \mid X_n = s, A_n = a)$
- R *realna funkcija nagrade, gdje s* $R(s, a)$ *označavamo očekivanu nagradu* r *za izvršenje akcije* a *u stanju* s*, tj.* $R(s, a) = \mathbb{E}(r_n \mid X_n = s, A_n = a)$*.*

Napomena 3.1.2. *U ovom radu prostor stanja je diskretan i konačan, ali postoje proširenja na kontinuirane i beskonačne prostore stanja.*

Markovljev proces odlučivanja zadovoljava Markovljevo svojstvo, odnosno vjerojatnost da završimo u nekom stanju s', ako se nalazimo u stanju s i poduzmemo akciju a, ovisi samo o trenutnom stanju s i akciji a.

Primjer 3.1.3. *Marsov rover motorno je vozilo koje putuje površinom planeta Mars. Jedan od njegovih zadataka je prikupljanje uzoraka različitih stijena koje uočava oko sebe. Rover treba planirati najefikasniju rutu koja će ga dovesti do svih zanimljivih stijena, s ciljem minimiziranja potrebnog vremena i gubitka energije. Za donošenje odluka o tome kamo dalje ići, robot treba poznavati samo trenutno stanje okoline - vlastitu lokaciju, lokaciju stijena, informaciju o tome koje su stijene već uzorkovane, vlastiti energetski nivo i slično. Robot ne mora pamtiti svoje prethodne lokacije kako bi odlučio o sljedećoj akciji. Za modeliranje takvih autonomnih agenata (agenata čije je donošenje odluka u većoj mjeri zasnovano na vlastitoj percepciji negoli na ugrađenom prethodnom znanu) koji djeluju u Markovljevog okruženju korisni su Markovljevi procesi odlučivanja. U problemu Marsovog rovera, skup stanja bi trebao sadržavati informacije poput lokacije robota, lokacije stijena, informaciju o tome koje stijene su već uzorkovane i stanja baterije. Primjer akcija koje robot može poduzeti su kretanje unaprijed, okretanje ulijevo ili udesno, uzorkovanje stijene i punjenje baterije. Prilikom kretanja u određenom smjeru, robot može proklizati ili promašiti smjer zbog nekih netočnosti u motoru. Zbog toga su učinci akcija stohastički pa je prikladno definirati prijelaznu funkciju. U našem primjeru rover prima nagradu svaki put kada uzorkuje stijenu koja nije bila uzorkovana prije, dok bi trebao biti kažnjen (primiti negativnu nagradu) kada uzorkuje stijenu koja je već uzorkovao. Također je moguće modelirati troškove akcija kao negativnu nagradu, primjerice, nametnuti trošak za svaku akciju robota povezan s potrošnjom energije koja proizlazi iz te radnje.*

3.2 Rješenje problema Markovljevog procesa odlučivanja

Cilj agenta je maksimizirati niz dolaznih nagrada ili neku njezinu funkciju. Agent može djelovati pod konačnim horizontom, tj. unaprijed definiranim brojem koraka unutar kojih mora izvršiti svoje zadatke. U tom scenariju optimalna akcija koju agent može izvršiti ovisi o preostalim vremenskim koracima. Na primjer, ako agent ne može postići glavni cilj unutar preostalih koraka, može pokušati izvršiti sekundarni cilj koji donosi manju nagradu, ali se brže ostvaruje. Stoga odabir akcije ne ovisi samo o trenutnom stanju, već i o broju preostalih vremenskih koraka. U beskonačnom horizontu, agent neprestano izvršava zadatke pa optimalna akcija ovisi isključivo o trenutnom stanju. Postavlja se pitanje kako vrednovati dugoročnu budućnost i kako usmjeriti agenta ka cilju koji nije dostižan u jednom koraku? U slučajnu beskonačnog horizonta, uvodimo faktor umanjenja $\gamma \in \langle 0, 1 \rangle$ koji koristimo kako bi s vremenom umanjili važnost budućih nagrada, a agent treba maksimizirati beskonačnu sumu umanjenih nagrada, odnosno očekivanu buduću umanjenu nagradu

$$\mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t r_t\right), \tag{3.1}$$

gdje r_t predstavlja nagradu u trenutku t. Ako je vrijednost γ bliža nuli, prednost se daje trenutačnim nagradama, a ako je bliža jedinici, tada se naglasak stavlja na dugoročnu nagradu. Suma (3.1) je ograničena s $\frac{r_{max}}{1-\gamma}$, gdje r_{max} predstavlja maksimalnu moguću nagradu, što osigurava konvergenciju algoritma kojeg navodimo u sljedećem koraku.

Politika agenta je svako preslikavanje $\pi : S \rightarrow A$ koje stanju agenta pridružuje neku akciju. Želimo pronaći optimalnu politiku koja maksimizira ukupnu očekivanu nagradu. Politiku agenta možemo definirati pomoću funkcije korisnosti $V : S \rightarrow \mathbb{R}$ koja svakom stanju pridružuje neku vrijednost, što je veća korisnost stanja, stanje je "poželjnije" za agenta. Vrijedi

$$\pi_V(s) = arg \underset{a \in A}{max} \left(R(s,a) + \gamma \sum_{s' \in S} tr(s,a,s')V(s')\right) \tag{3.2}$$

Optimalna funkcija korisnosti V^* je jedistvena i vrijedi

$$V^*(s) = \max_{a \in A}\left(R(s,a) + \gamma \sum_{s' \in S} tr(s,a,s')V(s')\right) \tag{3.3}$$

Jednadžbu (3.3) zovemo *Bellmanova* jednadžba koja uzima u obzir da korisnost stanja ne ovisi samo o neposrednoj nagradi nego i o budućim umanjenim nagradama. Politika π_{V^*} koja odgovara funkciji korisnosti V^* je optimalna politika. Umjesto definiranja korisnosti za svako stanje, možemo definirati funkciju $Q : S \times A \rightarrow \mathbb{R}$, poznatiju kao Q-vrijednost. $Q(s,a)$ predstavlja očekivanu nagradu koju agent treba primiti ukoliko poduzme akciju a u stanju s. Sada možemo definirati algoritam za računanje optimalnog rješenja Markovljevog procesa odlučivanja pomoću iteracija vrijednosti. Algoritam inicijalizira funkciju korisnosti na maksimalnu nagradu u jednom koraku. U svakoj iteraciji korisnost stanja popravlja se na temelju *Bellmanove* jednadžbe ažuriranja. Nastavljamo ažurirati korisnosti stanja dok ne konvergiraju. Tipično zaustavljamo algoritam kada maksimalna razlika između dviju uzastopnih funkcija korisnosti (poznata kao *Bellmanova* rezidualna vrijednost) padne ispod nekog ϵ.

Algoritam 1 Iteracije vrijednosti

Inicijaliziraj $V(s) = \max_{a \in A} R(s, a)$
while V ne konvergira **do**
  for $s \in S$ **do**
    for $a \in A$ **do**
      $Q(s, a) \leftarrow R(s, a) + \gamma \sum_{s' \in S} tr(s, a, s')V(s')$
    end for
    $V(s) \leftarrow \max_{a \in A} Q(s, a)$
  end for
end while
$\pi(s) \leftarrow \underset{a \in A}{arg\ max}\ Q(s, a)$

3.3 Primjene

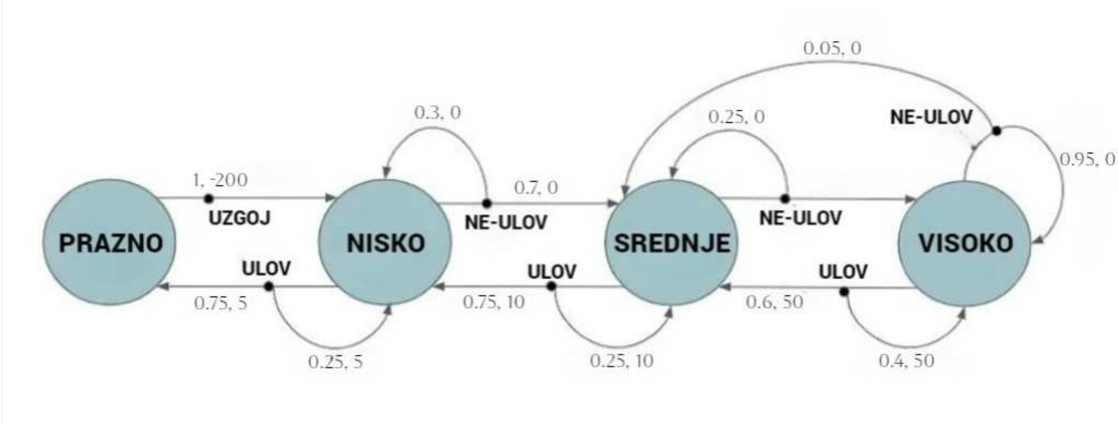
Primjer 3.3.1. *(Ulov lososa) Imamo problem u kojem želimo odrediti optimalan udio lososa za ulov tijekom godine na određenom području s ciljem maksimiziranja dugoročnog povrata. Nakon ulova, losos možemo prodati i svaki pojedini losos generira fiksan iznos dolara. No, ako se ulovi previše lososa, to može rezultirati smanjenim prinosom sljedeće godine. Ovaj problem možemo modelirati kao Markovljev proces odlučivanja na sljedeći način:*

- *Skup stanja je broj dostupnih lososa u određenoj godini. Pojednostavit ćemo problem i pretpostaviti da razlikujemo četiri stanja, PRAZNO (područje je bez lososa), NISKO (dostupan broj lososa je ispod određenog praga t_1), SREDNJE (dostupan broj lososa između t_1 i t_2) i VISOKO (dostupan broj lososa iznad t_2).*

- *Skup akcija čine samo dvije akcije, ULOV i NE-ULOV lososa.*

- *Lovljenje u svakom stanju donosi neku nagradu. Ako lovimo u stanju NISKO nagrada iznosi* 5$, *u stanju SREDNJE* 10$, *a u stanju VISOKO* 50$. *Ako nas neka akcija dovede do stanja PRAZNO, onda je kazna* 200$, *jer uzgoj novih lososa zahtijeva vrijeme i novac.*

- *Prijelaznu funkciju možemo shvatiti na sljedeći način: Akcija ULOV lososa u određenom stanju ima veću vjerojatnost prelaska u stanje s manjim brojem lososa, a akcija NE-ULOV ima veću vjerojatnost prelaska u stanje s većim brojem lososa.*



Slika 3.1: Prijelazni graf za problem ulova lososa.

Pogledajmo graf sa slike. Tranzicije između stanja prikazane su strelicama iznad koji je uređen par (prijelazna vjerojatnost, nagrada). Na primjer, od stanja SREDNJE akcija ULOV može voditi prema stanju NISKO s vjerojatnosti 0.75 *i nagradom* 10$ *ili može ostati u stanju SREDNJE s vjerojatnosti* 0.25 *i istom nagradom.*

Osim navedenog primjera, Markovljev proces odlučivanja može se koristiti u nekim poznatijim problemima poput problema putujućeg putnika. U tom slučaju, agent je trgovac, akcije su rute koje se mogu odabrati iz trenutnog stanja, nagrade predstavljaju troškove svake rute, a cilj je minimizirati funkciju troška tijekom trajanja putovanja. Markovljevi procesi odlučivanja korišteni su i za učenje računala kako igrati igre poput *Ponga, Pacmana* ili *AlphaGO*.

Poglavlje 4

Djelomično vidljiv Markovljev proces odlučivanja

4.1 Definicija i osnovna svojstva

U Markovljevom procesu odlučivanja agent uvijek ima potpuno znanje o svom položaju i okruženju. U stvarnosti informacije o okolini su samo djelomične i neprecizne pa uvodimo djelomično vidljiv Markovljev proces odlučivanja za situacije u kojima agent nema potpune informacije o trenutnom stanju okoline. Agent prima nejasna ili ograničena opažanja koja su posljedica njegovih akcija i stanja okoline. Na primjer, u scenariju s Marsovim roverom okolne stijene mogu sadržavati zanimljive minerale. Rover može aktivirati senzor kako bi provjerio je li stijena zanimljiva prije nego joj se približi, no točnost senzora opada s povećanjem udaljenosti od stijene. Okolina u kojem rover djeluje je i dalje Markovljeva, ali rover ima samo djelomične informacije o njoj. Ovaj tip procesa tek se nedavno pojavio u literaturi pa ćemo dati kratki pregled dosada dobivenih rezultata, a detaljnije je objašnjeno u [8], [9] i [10].

Definicija 4.1.1. *Djelomično vidljiv Markovljev proces odlučivanja s diskretnim vremenom je n-torka* $(S, A, tr, R, \Omega, O, b_0)$*, gdje su:*

- S, A, tr, R *parametri iz definicije 3.1.1 osnovnog Markovljevog procesa odlučivanja koji opisuju okolinu u kojoj agent djeluje*
- Ω *skup opažanja, najčešće skup svih mogućih izlaza senzora*
- $O(a, s', o)$ *vjerojatnost da će agent opaziti* $o \in \Omega$ *nakon izvršavanja radnje* a *i dolaska u stanje* s'*. Funkcija* O *modelira grešku senzora.*

Djelomično vidljiv Markovljev proces odlučivanja ne zadovoljava Markovljevo svojstvo s obzirom na opažanja. Budući da nemamo izravan pristup trenutnom stanju, naše odluke zahtijevaju praćenje moguće cijele povijesti procesa. Način na koji ga možemo učiniti Markovljevim je definiranje stanja vjerovanja - vektora vjerojatnosti gdje je $b(s)$ vjerojatnost da je agent u stanju s. Ovakav način pruža iste informacije kao da smo pratili cijelu povijest procesa. Stanje vjerovanja oponaša osnovno stanje pa je Markovljevo, stoga su potrebne samo informacije o prethodnom stanju vjerovanja, akcija koja je poduzeta i trenutno opažanje. b_0 predstavlja početno vjerovanje, prije nego agent izvrši neku akciju ili primi opažanje. Vjerovanja moramo ažurirati - kada je agent u stanju vjerovanja b, izvršavajući radnju a i promatrajući o, možemo odrediti novo stanje vjerovanja $b'(s) = \tau(b, a, o)$:

$$\begin{aligned}
b'(s) &= \mathbb{P}(s \mid a, o, b) \\
&= \frac{\mathbb{P}(s, a, o, b)}{\mathbb{P}(a, o, b)} \\
&= \frac{\mathbb{P}(o \mid s, a, b)\,\mathbb{P}(s \mid a, b)\,\mathbb{P}(a, b)}{\mathbb{P}(o \mid a, b)\,\mathbb{P}(a, b)} \\
&= \frac{\mathbb{P}(o \mid s, a, b)\,\mathbb{P}(s \mid a, b)}{\mathbb{P}(o \mid a, b)} \\
&= \frac{\mathbb{P}(o \mid s, a)\,\mathbb{P}(s \mid a, b)}{\mathbb{P}(o \mid a, b)} \\
&= \frac{O(a, s, o) \sum_{s' \in S} b(s') tr(s', a, s)}{\mathbb{P}(o \mid a, b)} \\
\mathbb{P}(o \mid a, b) &= \sum_{s \in S} b(s) \sum_{s' \in S} tr(s, a, s') O(a, s', o)
\end{aligned} \tag{4.1}$$

4.2 Rješenje problema djelomično vidljivog Markovljevog procesa odlučivanja

Rješenje djelomično vidljivog procesa odlučivanja je politika koja propisuje koju radnju poduzeti u svakom stanju vjerovanja kako bi maksimizirao

$$\mathbb{E}\left(\sum_{t=0}^{\infty} \gamma^t r_t\right). \tag{4.2}$$

Stanja vjerovanja su kontinuirana, rezultirajući beskonačnom skupom stanja što ovaj problem čini puno težim za rješavanje u usporedbi s osnovnim Markovljevim procesom odlučivanja. Spomenut ćemo nekoliko metoda aproksimacije koje se oslanjaju na rješenje procesa odlučivanja. Metoda najvjerojatnijeg stanja bira optimalnu akciju za stanje s najvećom vjerojatnoti u trenutnom stanju vjerovanja, no ne uzima u obzir stupanj sigurnosti trenutnog stanja niti vrijednost optimalne funkcije. Metoda glasovanja uzima u obzir te nesigurnosti odabirom akcije koja ima najveću vjerojatnost prema trenutnoj distribuciji stanja vjerovanja, a Q-aproksimacija ju dodatno poboljšava jer u obzir uzima Q-vrijednosti. U toj metodi biramo akciju koja maksimizira $\sum_s Q(s,a)b(s)$. Optimalno ponašanje često može uključivati akcije sakupljanje informacija koje se poduzimaju isključivo zato što poboljšavaju procjenu trenutačnog stanja agenta, omogućujući mu da donese bolje odluke u budućnosti. S obzirom da je u Markovljevom procesu odlučivanja trenutno stanje uvijek poznato, ove metode ne uzimaju u obzir akcije koje prikupljaju opažanja. Te akcije nemaju nikakvu vrijednost i neće biti odabrane navedenim metodama, no takve metode možemo koristiti za brzu inicijalizaciju funkcije korisnosti.

Tradicionalan način za rješavanje djelomično vidljivih procesa odlučivanja je putem osnovnog procesa odlučivanja nad prostorom vjerovanja djelomično vidljivog procesa. Definiramo:

$$\begin{aligned} R(b,a) &= \sum_{s\in S} b(s)R(s,a) \\ tr(b,a,b') &= \sum_{o\in\Omega} \mathbb{P}\left(b' \mid b,a,o\right)\mathbb{P}\left(o \mid b,a\right) \\ \mathbb{P}\left(b' \mid b,a,o\right) &= \begin{cases} 1 & \tau(b,a,o) = b' \\ 0 & \text{inače} \end{cases} \\ \mathbb{P}\left(o \mid b,a\right) &= \sum_{s\in S} b(s) \sum_{s'\in S} tr(s,a,s')O(a,s',o) \end{aligned} \tag{4.3}$$

Jednadžba iteracije vrijednost dana je sljedećim izrazom:

$$\begin{aligned} V_o(b) &= \max_{a\in A} \sum_{s\in S} b(s)R(s,a) \\ V_{t+1}(b) &= \max_{a\in A}\left(\sum_{s\in S} b(s)R(s,a) + \gamma \sum_{o\in\Omega} \mathbb{P}\left(o \mid b,a\right) V_t(\tau(b,a,o))\right) \end{aligned} \tag{4.4}$$

Možemo je rastaviti kao:

$$\begin{aligned} V_{t+1}(b) &= \max_{a\in A} V^a(b) \\ V^a(b) &= \sum_{o\in\Omega} V^a_o(b) \\ V^a_o(b) &= \frac{\sum_{s\in S} b(s)R(s,a)}{|\Omega|} + \gamma\cdot\mathbb{P}\left(o \mid b,a\right) V_t(\tau(b,a,o)) \end{aligned} \tag{4.5}$$

Sada je

$$\pi_V(b) = arg\max_{a \in A} \left(\sum_{s \in S} b(s)R(s,a) + \gamma \sum_{o \in \Omega} \mathbb{P}\left(o \mid b, a\right) V_t(\tau(b,a,o)) \right). \tag{4.6}$$

Djelomično vidljivi Markovljevi procesi odlučivanja imaju široku primjenu u različitim područjima gdje se donose odluke u uvjetima neizvjesnosti i djelomične informiranosti. Neke od najčešćih primjena su:

- robotika - za navigaciju i planiranje optimalnih putanja robota u nepoznatim okruženjima, uzimajući u obzir nepredvidljive prepreke i ograničenja, npr. autonomni automobili. Osim navigacije, koriste se i za manipulaciju objektima i pomažu robotima u donošenju odluka o tome kako optimalno uhvatiti, premjestiti ili manipulirati objektima u okruženju.
- računalni vid - prepoznavanje objekata na slikama ili videozapisima, uzimajući u obzir varijabilnost u slikama. Važni su i u razumijevanju scene, odnosno segmentaciji slika na različite dijelove od kojih svaki odgovara različitom objektu.
- industrija - za optimizaciju upravljanja zalihama, određivanje optimalnih količina naručivanja i minimiziranje troškova zaliha.
- zdravstveni sustav - za personalizirano planiranje liječenja, uzimajući u obzir individualne karakteristike pacijenata i njihove specifične potrebe kao i za donošenje odluka o upućivanju na dodatne pretrage ili liječenje.

Bibliografija

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

[2] F. Mahfuz, *Markov Chains and their applications*, 2021., dostupno na http://hdl.handle.net/10950/3704, (pristupljeno: veljača 2024.)

[3] P. Dymarski, *Hidden Markov Models, Theory and Applications*, InTech, 2011.

[4] L. Yang, *A Study of Hidden Markov Model*, University of Tennessee, 2004., dostupno na https://trace.tennessee.edu/utk_gradthes/2326, (pristupljeno: siječanj 2024.)

[5] B. Yoon, *Hidden Markov Models and their Applications in Biological Sequence Analysis*, Current genomics, 2009., dostupno na https://www.researchgate.net/publication/41623535, (pristupljeno: siječanj 2024.)

[6] https://www.linkedin.com/pulse/part-speech-tagging-using-hidden-markov-model-gim-pei-ng, (pristupljeno: veljača 2024.)

[7] J. Boudnaya, A. Haytoumi, O. Eddayer, A. Mkhida, *Prediction of Robot Localization States Using Hidden Markov Models*, 2021., dostupno na https://www.researchgate.net/publication/344046988, (pristupljeno: siječanj 2024.)

[8] G. Shani, *Learning and Solving Partially Observable Markov Decision Processes*, 2007., dostupno na https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=2051d7d9b99495dc6686ab8f94257c8eb6ef8dd3, (pristupljeno: veljača 2024.)

[9] https://www.pomdp.org/, (pristupljeno: veljača 2024.)

[10] I. Chadès, L.V. Pascal, S. Nicol, C.S. Fletcher, J. Ferrer-Mestres, *A primer on partially observable Markov decision processes (POMDPs)*, Methods in Ecology and Evolution, 2021., dostupno na https://doi.org/10.1111/2041-210X.13692, (pristupljeno: veljača 2024.)

Sažetak

U ovom radu napravljen je pregled četiri tipa Markovljevih modela. Opisana je struktura Markovljevih lanaca kao osnovog tipa Markovljevih procesa te je spomenut *PageRank* algoritam i primjena u predviđanju vremenske prognoze. Nakon toga je definirano što su skriveni Markovljevi lanci te su objašnjeni glavni problemi modela. Autorov doprinos ovoj temi je uvođenje primjera koji ilustriraju *backward*, *forward* i *Viterbijev* algoritam te omogućuju razumijevanje praktične primjene algoritama i tema obrađenih u radu. Navedene su i primjene u biologiji, obradi prirodnog jezika i robotici koje pokazuju važnost modela. Zatim je definiran Markovljev proces odlučivanja i dano je rješenje problema maksimizacije nagrade, a pomoću primjera pokazano je kako koristiti model za modeliranje donošenja odluka. Na kraju je dan kratak osvrt na djelomično vidljive Markovljeve procese odlučivanja.

Summary

In this paper, an overview of four types of Markov models is provided. The structure of Markov chains, as the basic type of Markov processes, is described, and the *PageRank* algorithm is mentioned along with its application in weather forecasting. Subsequently, hidden Markov chains are defined, and the main problems of the model are explained. The author's contribution to this topic lies in the introduction of examples illustrating the *backward*, *forward* and *Viterbi* algorithms, enabling understanding of the practical application of the algorithms and topics discussed in the paper. Applications in biology, natural language processing, and robotics are also mentioned, highlighting the importance of the models. Next, the Markov decision process is defined, and a solution to the reward maximization problem is provided, demonstrating how to use the model to model decision-making through examples. Finally, a brief overview of partially observable Markov decision processes is given.

Životopis

Rođena sam 12. srpnja 1999. godine u Zagrebu. Pohađala sam osnovnu školu Dragutina Tadijanovića u Zagrebu od 2006. do 2014. godine, a svoje srednjoškolsko obrazovanje sam nastavila u V. gimnaziji u Zagrebu koju sam završila 2018. godine. Te iste godine upisujem preddiplomski studij Matematika na Prirodoslovno-matematičkom fakultetu u Zagrebu, a 2021. sam stekla zvanje prvostupnika matematike. Svoje obrazovanje nastavljam na Prirodoslovno-matematičkom fakultetu upisom diplomskog studija Matematička statistika 2021. godine.