Osnove Markovljevih i skrivenih Markovljevih modela i njihova primjena u obradbi informacija

Prema izvatku iz knjige „Speech and Language Processing“, 3. izdanje,

autora: Dan Jurafsky i James H. Martin, sveučilište Stanford, SAD

Prijevod: prof.dr.sc. Davor Petrinović

Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva

Zagreb, studeni 2022.

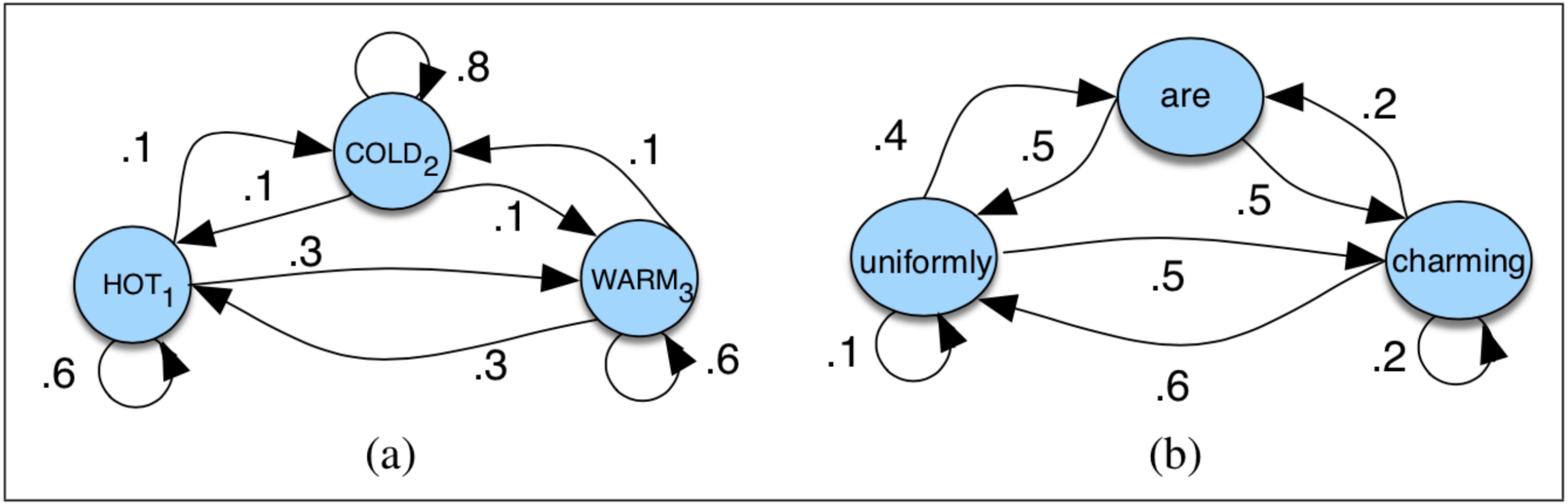
Izvor korišten u pripremi materijala za OI

- Tematici Markovljevih i skrivenih Markovljevim modela posvećene su stotine knjiga i preglednih radova pa je raspoloživa literatura uistinu iznimno opsežna i bogata.
- Kao najpogodniji izvor za ovaj predmet odabrana je elektronička knjiga pod naslovom „*Speech and Language Processing*“, 3rd ed. Draft, autora: Dan Jurafsky i James H. Martin sa sveučilišta Stanford. Radna inačica trećeg izdanja ove knjige raspoloživa je na poveznici:
- https://web.stanford.edu/~jurafsky/slp3/
- Pri tome, konkretni materijal koji se odnosi na Markovljeve modele i koji je korišten za pripremu ovih materijala je *Appendix A* ove knjige. https://web.stanford.edu/~jurafsky/slp3/A.pdf

Markovljevi lanci

Markovljev lanac - ilustracija

- Markovljev lanac za vrijeme (a) i za riječi (b), prikazuje stanja i prijelaze.
- Potrebna je i početna distribucija π;
  - π = [0.1, 0.7, 0.2] za model (a) značila bi vjerojatnost 0.7 pokretanja u stanju 2 (hladno), vjerojatnost 0.1 pokretanja u stanju 1 (vruće), itd.



Markovljev lanac

- Andrey Andreyevich Markov (14. 6. 1856 – 20. 7. 1922.)
- Ruski matematičar
- Najpoznatiji po svojem radu na stohastičkim procesima, a posebno na istraživanju modela koji će kasnije biti poznat upravo po njegovom prezimenu kao Markovljev lanac.

Markovljeva pretpostavka

- Zamislimo da ovaj lanac prolazi kroz niz stanja q_1 q_2 ... q_i
- Čime je određena vjerojatnost da se u *i*-tom koraku nalazimo u nekom odabranom stanju *a*, npr. q_i=*a* (npr. *a*=HOT za slučaj primjera lanca s vremenom)?

$$\textbf{Markov Assumption:} \quad P(q_i = a|q_1...q_{i-1}) = P(q_i = a|q_{i-1}) \qquad \text{(A.1)}$$

- Markov pretpostavlja da je vjerojatnost određena isključivo prvim prethodnim stanjem q_{i-1} (iz kojeg je došao), a ne i ranijim stanjima modela: $q_{i-2,}$ $q_{i-3,}$ .. q_1

Formalni opis Markovljevog lanca

- $Q = q_1\ q_2 \ldots q_N$ je skup od N mogućih stanja modela;
- $A = a_{11}\ a_{12} \ldots a_{N1}, \ldots a_{NN}$ je matrica vjerojatnosti prijelaza A, gdje svaki njen član a_{ij} predstavlja vjerojatnost prelaska iz stanja i u stanje j, pri čemu suma članova a_{ij} u svakom retku (tj. za stupce $j=1$ do N) je jednaka jedinici, jer sustav mora ili ostati u istom stanju ($i \to i$), ili mora prijeći u bilo koje novo stanje ($i \to j$);
- $\pi = \pi_1, \pi_2, \ldots, \pi_N$ je vektor početne raspodjele vjerojatnosti po stanjima. Vrijednost π_i je vjerojatnost da će Markovljev lanac započeti u stanju i.

Primjer izračuna vjerojatnosti niza osmatranja za zadani model

- Za raniji primjer modela izmjene vremenskih prilika koji je prikazan dijagramom stanja s N=3 stanja: **Hot** (1)**, Cold** (2) **i Warm** (3) i s vjerojatnostima prijelaza A i inicijalnom distribucijom π prikazanom na istoj slici potrebno je odrediti vjerojatnosti osmatranja ova dva niza:

- Hot, Hot, Hot, Hot
- Cold, Hot, Cold, Hot
  - Što vam razlika u tim vjerojatnostima govori o modelu vremenskih prilika koje su kodirane mrežom na slici?

Što ako stanja modela nisu izravno osmotriva?

- Markovljev lanac je koristan je kada moramo izračunati vjerojatnost za slijed osmotrivih događaja.
- Međutim, u mnogim su slučajevima događaji koji nas zanimaju skriveni: ne promatramo ih izravno, nego samo posredno.
- Npr. oznake riječi u rečenici (pisanom tekstu) nisu nam vidljive.
- Riječi se po **vrstama riječi** dijele na: *imenice, glagole, pridjeve, zamjenice, brojeve, priloge, prijedloge, veznike i uzvike*, ali
- riječi se dijele i s obzirom na zadatak kojeg nose u rečenici, što je definiranom službom riječi. **Prema službi riječi** u rečenici dijelimo ih na: *predikat, subjekt, objekt, priložnu oznaku, atribut i apoziciju*.

Primjer skrivenih oznaka riječi u tekstu

- Analiziraj ove rečenice i dodaj oznake svake riječi obzirom na vrstu i službu:
  - *Naša nova škola ima golemu sportsku dvoranu.*
  - *Sutra ujutro idem autobusom na more.*
  - *Posvađali su se zbog utakmice.*
  - *Stari profesor Matić uvijek zaboravi dnevnik.*
- Za pomoć, …
- https://matemilas.weebly.com/uploads/1/2/4/7/124781779/sintaksa_-_za_ponavljanje.pdf

Što ako stanja modela nisu izravno osmotriva?

- Prilikom čitanja, oznake vrsta riječi nisu navedene u tekstu, a ipak ga uspješno razumijemo. Umjesto toga, vidimo samo riječi i **moramo sami zaključiti oznake iz niza riječi**. Stoga **oznake nazivamo skrivenim stanjima** jer nisu izravno osmotrive.

- Skriveni Markovljev model (engl. **H**idden **M**arkov **M**odel, **HMM**) omogućuje nam da istovremeno razgovaramo o opaženim događajima (poput niza riječi koje vidimo) kao i o skrivenim događajima (poput oznaka dijela govora) za koje smatramo da su uzročni čimbenici osmotrenih riječi u našem vjerojatnosnom modelu.

Skriveni Markovljev model

Skriveni Markovljev model (engl. *Hidden Markov model, HMM*)

- Formalno je opisan ovim elementima:

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} \ldots a_{ij} \ldots a_{NN}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i$
$O = o_1 o_2 \ldots o_T$	a sequence of T **observations**, each one drawn from a vocabulary $V = v_1, v_2, \ldots, v_V$
$B = b_i(o_t)$	a sequence of **observation likelihoods**, also called **emission probabilities**, each expressing the probability of an observation o_t being generated from a state i
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	an **initial probability distribution** over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. Also, $\sum_{i=1}^{n} \pi_i = 1$

Skriveni Markovljev model

- Skup stanja modela *Q*, prijelazna matrica *A* i vektor inicijalne razdiobe vjerojatnosti stanja π jednaki su kao i za raniji opis Markovljevog lanca.
- Međutim za skriveni Markovljev model imamo i **niz osmotrenih izlaznih simbola** $\boldsymbol{O} = o_1\, o_2 \ldots o_T$ kojeg još možemo nazvati slijedom opažanja duljine *T*, pri čemu je svako opažanje izvučeno iz **konačnog rječnika izlaznih simbola**, $\boldsymbol{V} = v_1, v_2, \ldots, v_V$.
- Dodatni parametar HMM modela je i **matrica vjerojatnosti pojedinog opažanja** $\boldsymbol{B}=b_i(o_t)$ koja opisuje slijed izvjesnosti osmatranja, koje se još nazivaju i emisijskim vjerojatnostima. Svaki element $b_i(o_t)$ ove matrice iskazuje vjerojatnost generiranja osmotrenog simbola o_t u *i*-tom stanju modela.
- Broj stanja modela *N*, duljina osmotrenog niza *T* i veličina rječnika izlaznih simbola *V* su općenito različiti (cijeli) brojevi.

Skriveni Markovljev model - pretpostavke

- Slično kao i Markovljev lanac, skriveni Markovljev model prvog reda je temeljen na ove dvije pretpostavke:
- 1) vjerojatnost određenog stanja ovisi isključivo o prethodnom stanju:

$$\textbf{Markov Assumption:} \quad P(q_i|q_1 \ldots q_{i-1}) = P(q_i|q_{i-1}) \tag{A.4}$$

- 2) vjerojatnost izlaznog promatranja o_i ovisi samo o stanju q_i u kojem je proizvedeno to promatranje, a ne o bilo kojem drugom stanju ili bilo kojem drugom promatranju:

$$\textbf{Output Independence:} \quad P(o_i|q_1 \ldots q_i, \ldots, q_T, o_1, \ldots, o_i, \ldots, o_T) = P(o_i|q_i) \tag{A.5}$$

Skriveni Markovljev model – primjer – prema Jason Eisner, 2002.

- Zamislite da ste klimatolog 2799. godine i proučavate povijest globalnog zatopljenja. Ne možete pronaći zapise o vremenu u Baltimoreu u saveznoj državi Maryland za ljeto 2020. godine, ali pronalazite osobni dnevnik Jasona Eisnera u kojem navodi koliko je sladoleda Jason jeo svakog dana tog ljeta.
- Cilj nam je koristiti ta zabilježena opažanja **za procjenu** temperature svakog dana.
- Pojednostavit ćemo maksimalno ovaj zadatak pretpostavljajući da postoje samo dvije vrste dana: hladni (**C**(old)) i vrući (**H**(ot)).

Skriveni Markovljev model – primjer – prema Jason Eisner 2002

- **Zadatak**: s obzirom na slijed zabilježenih promatranja *O* (cijeli broj u nizu predstavlja broj pojedenih sladoleda pojedinog dana) pronađite „skriveni“ slijed *Q* vremenskih stanja (***H*** ili ***C***) zbog kojih je Jason pojeo jedan ili više sladoleda. Pri tome nikada nije pojeo više od tri, ali je uvijek pojeo barem jedan sladoled.
- Slika A.2 prikazuje primjer HMM modela za opisani zadatak „sladoleda”. Dva skrivena stanja (***H*** i ***C***) odgovaraju toplom i hladnom vremenu, a opažanja (izvučena iz abecede O = {1, 2, 3}) odgovaraju broju sladoleda koje je Jason pojeo određenog dana.

Skriveni Markovljev model – primjer – prema Jason Eisner 2002

- HMM model za odnose temperature i broja pojedenih sladoleda

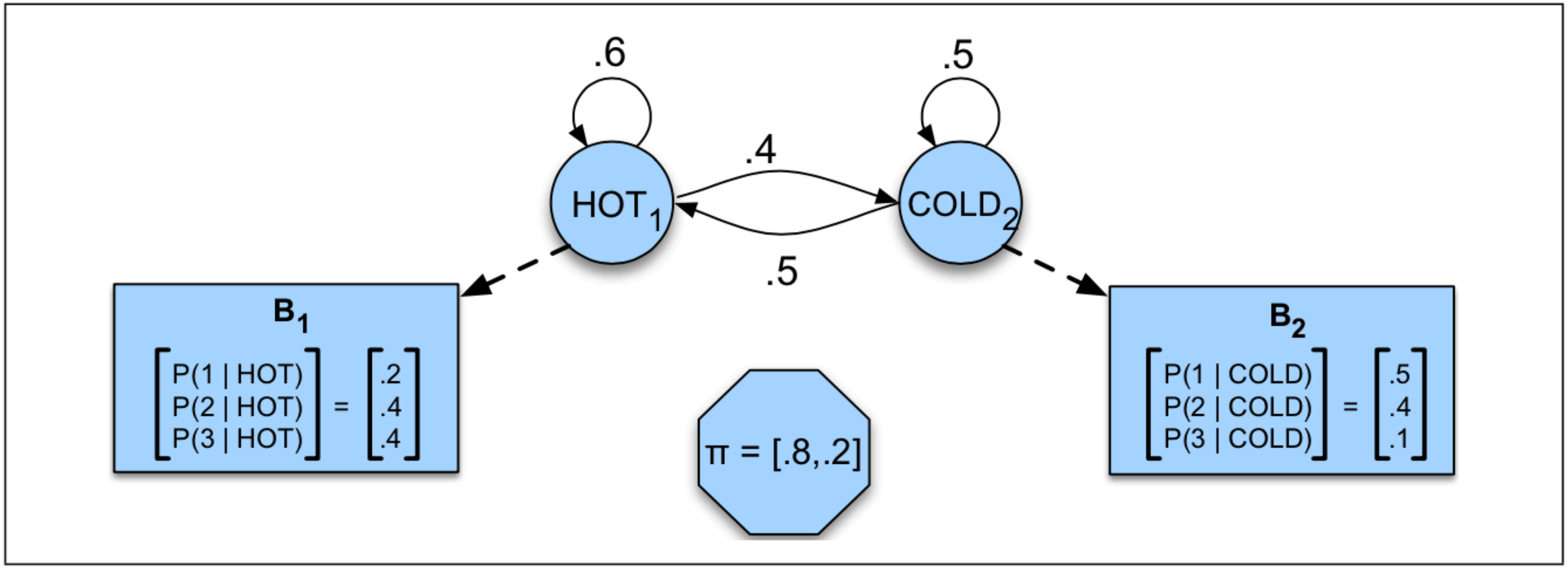


Figure A.2 A hidden Markov model for relating numbers of ice creams eaten by Jason (the observations) to the weather (H or C, the hidden variables).

Koje probleme uopće želimo/trebamo rješavati za HMM modele?

- Lawrence R. Rabiner je u vrlo utjecajnom preglednom članku iz 1989. pod naslovom *„A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition“*, (koji je zasnovan na tutorijalima Jacka Fergusona iz 1960-ih), uveo ideju da skriveni Markovljevi modeli trebaju biti karakterizirani s tri temeljna problema:
- **Određivanje izvjesnosti**,
- **Dekodiranje stanja**, i
- **Treniranje modela**.

- https://web.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/tutorial%20on%20hmm%20and%20applications.pdf

Koje probleme uopće želimo/trebamo rješavati za HMM modele?

- **Problem 1 (izvjesnost)**: S obzirom na zadani HMM model $\lambda = (A, B)$ i slijed promatranja O, odredite izvjesnost $P(O|\lambda)$, tj. koliko je izvjesno da je osmotreni niz O nastao upravo kao izlaz zadanog modela λ.
- **Problem 2 (dekodiranje)**: S obzirom na slijed osmatranja O i HMM model $\lambda = (A, B)$, otkrijte najbolji skriveni slijed stanja Q koji je generirao to osmatranje.
- **Problem 3 (treniranje)**: S obzirom na slijed osmatranja O i skup mogućih stanja HMM-a, naučite HMM parametre A i B, kako bi taj model najbolje opisivao odnos slijeda osmatranja i skupa stanja.

Kako ćemo riješiti te probleme?

- U svrhu rješavanja problema 1 i 3 (izvjesnost osmatranja za zadani model i treniranje modela), koristit ćemo se pomoću dva algoritma kojima je moguće značajno umanjiti složenost izračuna izvjesnosti osmatranja, a to su:
- algoritam **„Unaprijed”**,
- algoritam **„Unazad”**, i
- njihova kombinacija, algoritam **„Unaprijed-unazad”**, a
- za utvrđivanje najboljeg slijeda skrivenih stanja, koristit ćemo se: **„Viterbijevim”** algoritmom.

Izračun izvjesnosti korištenjem algoritma “Unaprijed”

Algoritam „Unaprijed” za izračun izvjesnosti osmatranja

- Naš prvi problem je izračunati izvjesnost određenog slijeda osmatranja za zadani model.
- Na primjer, za HMM model na slici A.2 koji statistički opisuje konzumaciju sladoleda, odredimo kolika je vjerojatnost osmatranja slijeda „3 1 3“?
- Matematički zapisano …
- **Izračun izvjesnosti**: S obzirom na zadani HMM $\lambda = (A, B)$ i niz osmatranja O, odredite izvjesnost $P(O|\lambda)$.

Algoritam „Unaprijed” za izračun izvjesnosti osmatranja

- Za Markovljev lanac, gdje su vanjska osmatranja identična kao i skriveni događaji, mogli bismo izračunati vjerojatnost „3 1 3“ jednostavno sljedeći stanja označena s 3 1 3 i množeći vjerojatnosti uzduž lukova.
- Za skriveni Markovljev model stvari nisu tako jednostavne. Želimo utvrditi vjerojatnost slijeda promatranja pojedenih sladoleda poput „3 1 3“, ali ne znamo koji je bio skriveni slijed stanja!
- Kako riješiti ovaj problem … ?
- Moramo pretpostaviti da „ipak poznamo” slijed stanja, jer je tada moguće naći ovu vjerojatnost … pa to onda ponoviti za sve moguće sljedove stanja i pribrojiti.

Algoritam „Unaprijed” – uz pretpostavljeni i fiksirani slijed skrivenih stanja

- S obzirom na takvo preslikavanje „jedan-na-jedan” i na Markovljevu pretpostavku izraženu jednadžbom A.4, za određeni slijed skrivenog stanja $Q = q_0, q_1, q_2, \ldots, q_T$ i pripadajući slijed izlaznih osmatranja $O = o_1, o_2, \ldots, o_T$, izvjesnost osmotrenog slijeda se nalazi kao:

$$P(O|Q) = \prod_{i=1}^{T} P(o_i|q_i) \qquad \text{(A.6)}$$

- Izračun *unaprijedne vjerojatnosti* za naše osmatranje sladoleda „3 1 3“ iz jednog mogućeg niza skrivenih stanja „vruće vruće hladno“ prikazano je u jednadžbi A.7. i na slici A.3 koja grafički prikazuje ovaj izračun.

Algoritam „Unaprijed” – uz pretpostavljeni i fiksirani slijed skrivenih stanja

- Za jedan pretpostavljeni slijed skrivenih stanja (Hot, Hot, Cold):

$$P(3\ 1\ 3|\text{hot hot cold}) = P(3|\text{hot}) \times P(1|\text{hot}) \times P(3|\text{cold}) \qquad \text{(A.7)}$$

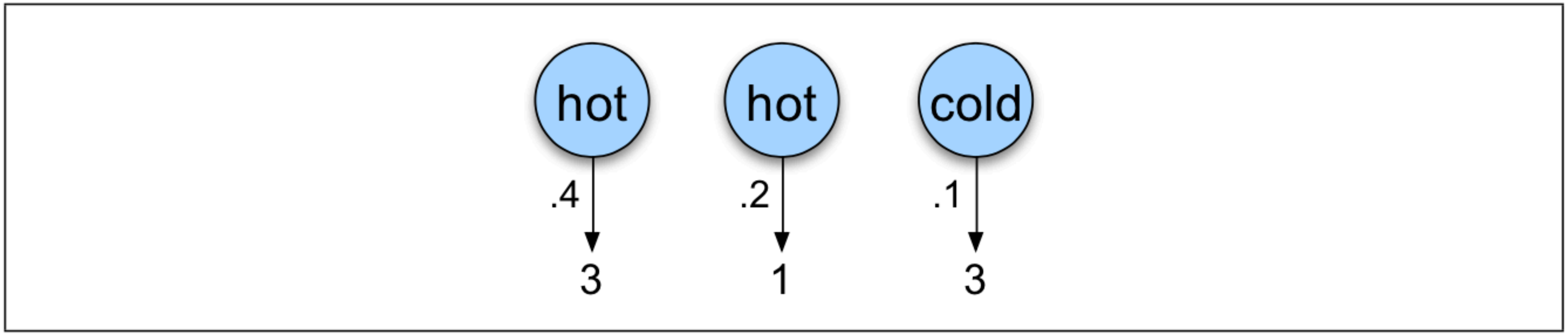


Figure A.3 The computation of the observation likelihood for the ice-cream events *3 1 3* given the hidden state sequence *hot hot cold*.

Algoritam „Unaprijed” – uz kombiniranje svih mogućih sljedova skrivenih stanja

- Mi zapravo ne znamo kakav je bio skriveni slijed stanja (promjene vremena), pa stoga moramo izračunati vjerojatnost sladolednih događaja „3 1 3“ zbrajanjem svih mogućih vremenskih sljedova stanja, ponderiranih njihovom vjerojatnošću.
- Izračunajmo prvo **zajedničku vjerojatnost** da se nalazimo u određenom vremenskom slijedu stanja *Q* i da pri tome generiramo određeni izlazni slijed *O* događaja sladoleda. Općenito, to možemo zapisati ovim izrazom:

$$P(O,Q) = P(O|Q) \times P(Q) = \prod_{i=1}^{T} P(o_i|q_i) \times \prod_{i=1}^{T} P(q_i|q_{i-1}) \qquad \text{(A.8)}$$

Algoritam „Unaprijed” – zajednička vjerojatnosti niza stanja *Q* i osmatranja *O*

- Za jedan pretpostavljeni slijed skrivenih stanja:

$$
\begin{aligned}
P(3\ 1\ 3, \text{hot hot cold}) &= P(\text{hot}|\text{start}) \times P(\text{hot}|\text{hot}) \times P(\text{cold}|\text{hot}) \\
&\quad \times P(3|\text{hot}) \times P(1|\text{hot}) \times P(3|\text{cold})
\end{aligned}
\tag{A.9}
$$

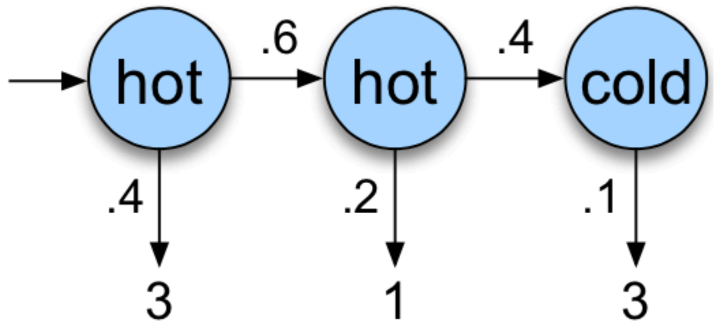


Figure A.4 The computation of the joint probability of the ice-cream events *3 1 3* and the hidden state sequence *hot hot cold*.

Algoritam „Unaprijed” – uz kombiniranje svih mogućih sljedova skrivenih stanja

- Sad kad znamo izračunati zajedničku vjerojatnost opažanja s određenim slijedom skrivenih stanja, možemo izračunati ukupnu vjerojatnost opažanja jednostavnim zbrajanjem svih mogućih sljedova skrivenih stanja:

$$P(O) = \sum_{Q} P(O,Q) = \sum_{Q} P(O|Q)P(Q) \qquad \text{(A.10)}$$

- Za ovaj konkretni primjer, broj mogućih kombinacija za slijed vremenskih prilika je 8, tj. potrebno je zbrojiti slučajeve: „hladno hladno hladno“, „hladno hladno vruće“, „vruće vruće hladno“, i tako sve do osmog mogućeg slijeda stanja „vruće vruće vruće“.

Algoritam „Unaprijed” – složenost kombiniranja svih mogućih sljedova skrivenih stanja

- Na prvi pogled zvuči jednostavno, … ali je li to uopće izvedivo?
- Za HMM model s N skrivenih stanja i sa slijedom osmatranja duljine T postoje N^T mogućih skrivenih nizova stanja.
- Za stvarne primjene, gdje su vrijednosti N i T velike, **N^T je iznimno velik broj**, tako da ne možemo izračunati ukupnu vjerojatnost osmatranja računanjem zasebne vjerojatnosti osmatranja za svaku pretpostavljenu skrivenu sekvencu stanja, a zatim ih zbrojiti.
- Ima li rješenja … ?
- Odgovor je upravo u **učinkovitom rekurzivnom načinu izvedbe** algoritma „**Unaprijed**”.

Algoritam „Unaprijed” – učinkovita rekurzivna izvedba

- Umjesto izravnog kombiniranja svih sljedova, čija je složenost ekstremno eksponencijalna, koristimo učinkoviti $O(N^2T)$ algoritam koji se naziva algoritam „**Unaprijed**“ (engl. *forward algorithm*).
- On pripada skupini dinamičkog programiranja, tj. to je algoritam koji koristi tablicu za pohranu među-vrijednosti dok gradi vjerojatnost slijeda promatranja.
- Algoritam „**Unaprijed**“ izračunava vjerojatnost promatranja zbrajanjem vjerojatnosti svih mogućih putova skrivenih stanja koji bi mogli generirati osmotrenu sekvencu, ali to čini učinkovito implicitnim preklapanjem svake od mogućih staza u jednu „**rešetku unaprijed**“.

Algoritam „Unaprijed” – pomoću rešetke stanja

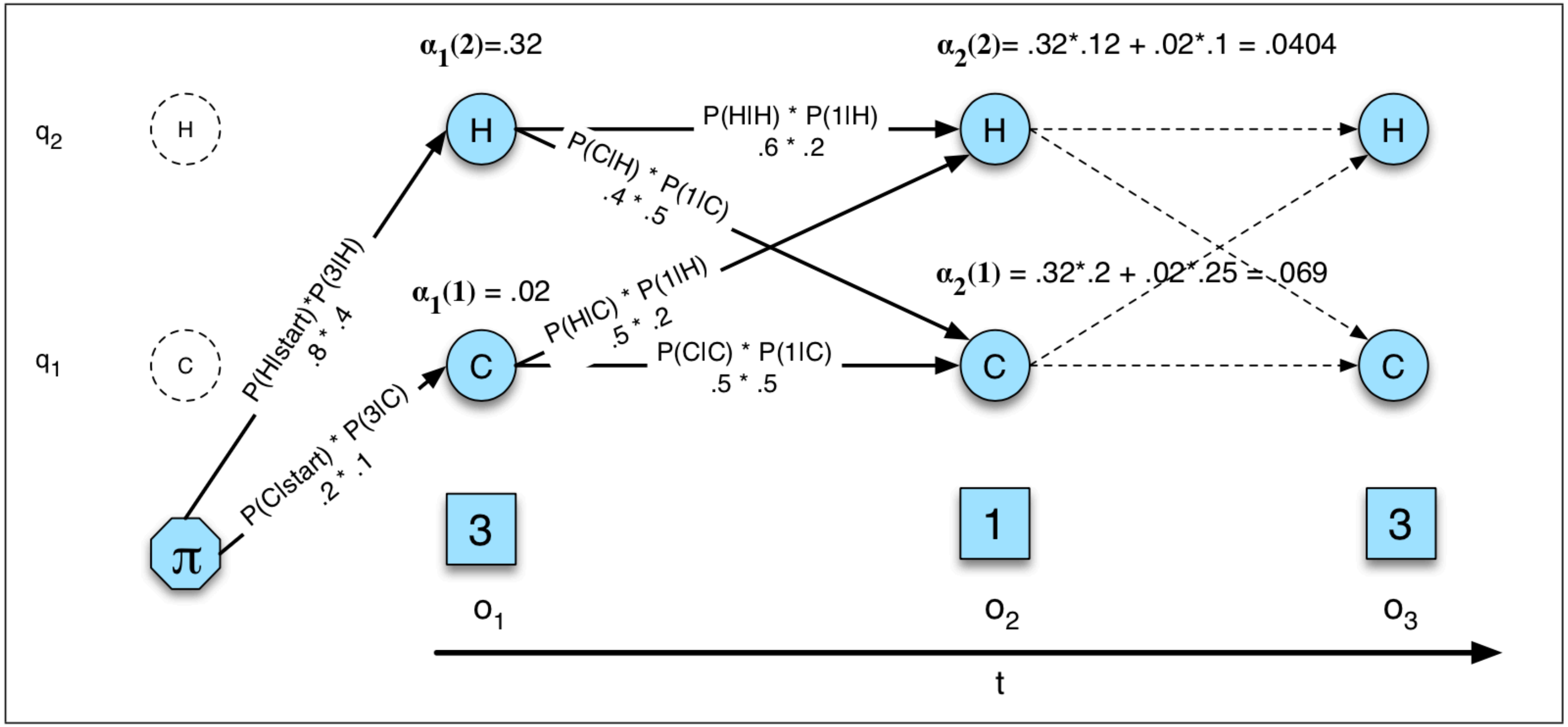


Figure A.5 The forward trellis for computing the total observation likelihood for the ice-cream events *3 1 3*. Hidden states are in circles, observations in squares. The figure shows the computation of $\alpha_t(j)$ for two states at two time steps. The computation in each cell follows Eq. A.12: $\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i)a_{ij}b_j(o_t)$. The resulting probability expressed in each cell is Eq. A.11: $\alpha_t(j) = P(o_1, o_2 \ldots o_t, q_t = j|\lambda)$.

Algoritam „Unaprijed” – pomoću rešetke stanja

- Slika A.5 prikazuje primjer takve „**rešetke unaprijed**“ za izračunavanje izvjesnosti osmotrenog niza „3 1 3“ s obzirom na sve moguće skrivene sljedove stanja.
- Radi jednostavnosti prikaza, izračun se ilustrira samo za prva dva osmotrena izlazna simbola o_1=3 i o_2=1, a analogno se proširuje i za zadnji osmotreni simbol o_3=3.
- Svaki stupac u ovoj rešetci odgovara pojedinom osmotrenom simbolu, dok svaki redak označava jedno od mogućih stanja u kojem je HMM model bio u tom trenutku (**H**, q_t=2 ili **C**, q_t=1).
- Strelice ukazuju na sve moguće prijelaze iz bilo kojeg zatečenog stanja u trenutačnom koraku osmatranja *t* u bilo koje novo (ili isto) stanje narednom koraku *t*+1. Horizontalne strelice označavaju zadržavanje istog stanja, a dijagonalne označavaju njegovu promjenu.

Algoritam „Unaprijed” – pomoću rešetke stanja

- Kako riječima tumačimo prikazane vrijednosti u rešetci stanja?
- HMM mreža kreće iz početnog stanja π (start) u kojem se sustav nalazi prije pojave prvog izlaznog simbola, te prelazi u stanje **H** (q_1=2) s vjerojatnosti 0.8 ili u stanje **C** (q_1=1) s vjerojatnosti 0.2.
- U prvom vremenskom trenutku t=1 osmotren je simbol o_1=3, pa će vjerojatnost da se mreža zatekla u stanju **H** biti $\pi_2 * b_2(o_1=3) = 0.8*0.4=0.32$, a vjerojatnost da se po isteku tok događaja zatekla u stanju **C** biti jednaka $\pi_1 * b_1(o_1=3) = 0.2*0.1=0.02$.
- Te su vjerojatnosti na slici označene kao $\alpha_1(2)$=0.32, odnosno $\alpha_1(1)$=0.02, pri čemu subskript 1 označava prvi vremenski trenutak t=1.
- U ovom primjeru je stanje **H** indeksirano kao drugo stanje, a stanje **C** kao prvo stanje, pa je taj indeks stanja označen kao argument unutar zagrada.

Algoritam „Unaprijed” – pomoću rešetke stanja

- Kako sada idemo dalje?
- Sada ove dvije vrijednosti koristimo za izračun $\alpha_2(2)$ i $\alpha_2(1)$ temeljem matrice prijelaza *A* (koja opisuje vjerojatnosti svih mogućih unaprijednih puteva u mreži), te poznavanjem vjerojatnosti osmatranja drugog izlaznog simbola o_2=1 u svakom od dva moguća stanja (**H** i **C**) koje su definirane u matrici izlaznih vjerojatnosti *B*.
- Konkretno, u stanju **H** u koraku *t*=2 možemo završiti preko dva moguća puta:
  - ravnog puta s vjerojatnošću $\alpha_1(2) * a_{22} * b_2(o_2=1) = 0.32 * 0.6 * 0.2$, ili preko
  - dijagonalnog puta iz stanja **C** s vjerojatnosti $\alpha_1(1) * a_{12} * b_2(o_2=1) = 0.02 * 0.5 * 0.2$.
- Zbrajanjem ove dvije vjerojatnosti dobivamo $\alpha_2(2)=0.0404$.
- Analogno zbrajanjem vjerojatnosti za dva moguća puta koji završavaju u stanju **C** u koraku *t*=2 dobivamo vjerojatnost $\alpha_2(1)=0.069$.
- Znači, da je uz zadani model vjerojatnost opažanja prva dva simbola „3 1“ jednaka $\alpha_2(2) + \alpha_2(1)=0.1094$. Isti postupak sada ponavljamo i za treći osmotreni simbol o_3=3, kako bi dobili izvjesnost osmatranja cijelog niza za zadani model.

Algoritam „Unaprijed” – unaprijedna vjerojatnost

- Općenito svaki čvor $\alpha_t(j)$ u rešetki ovog algoritma „Unaprijed“ predstavlja vjerojatnost da se mreža zatekne u stanju *j* nakon što osmotri prvih *t* opažanja, i to s obzirom na zadane parametre automata λ, pa je stoga zovemo „**unaprijedna vjerojatnost**”.
- Vrijednost svakog čvora $\alpha_t(j)$ izračunava se zbrajanjem vjerojatnosti svakog puta koji bi nas mogao dovesti do tog čvora. Formalno, svaki čvor izražava sljedeću vjerojatnost:

$$\alpha_t(j) = P(o_1, o_2 \dots o_t, q_t = j|\lambda) \qquad \text{(A.11)}$$

.. ovdje q_t=*j* označava „da je *t*-to stanje u nizu stanja upravo stanje *j*“.

Algoritam „Unaprijed” – unaprijedna vjerojatnost

- Dakle, ovu vjerojatnost $\alpha_t(j)$ izračunavamo zbrajanjem produljenja svih puteva koji vode do trenutnog čvora. Za dano stanje q_j u trenutku t, vrijednost $\alpha_t(j)$ izračunavamo kao:

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t) \tag{A.12}$$

- Tri faktora umnoška u svakom članu ove sume koje koristimo za izračun unaprijedne vjerojatnosti u trenutku t temeljem produljenja svih prethodnih puteva su:
- $\alpha_{t-1}(i)$ … **prethodna vjerojatnost puta unaprijed** iz prethodnog vremenskog koraka,
- a_{ij} … **vjerojatnost prijelaza** iz prethodnog stanja q_i u trenutno stanje q_j,
- $b_j(o_t)$ … **izvjesnost osmatranja** izlaznog simbola o_t u stanju j.

Algoritam „Unaprijed” – unaprijedna vjerojatnost

- Razmotrimo izračunavanje $\alpha_2(2)$ na slici A.5, tj. vjerojatnost da ćemo se nalaziti u koraku 2 u stanju 2 generirajući djelomično promatranje „3 1“, primjenom ovog **indukcijskog koraka**.
- Izračunavamo ovu vrijednost proširujući α-vjerojatnosti iz vremenskog koraka 1, kroz dva moguća puta , gdje se svako proširenje sastoji od tri navedena faktora:

$$\alpha_1(1) \times P(\mathbf{H}|\mathbf{C}) \times P(1|\mathbf{H}) + \alpha_1(2) \times P(\mathbf{H}|\mathbf{H}) \times P(1|\mathbf{H})$$

- Slika A.6 prikazuje općenitu vizualizaciju ovog indukcijskog koraka za izračunavanje vrijednosti u jednom novom čvoru rešetke, temeljem unaprijednih vjerojatnosti iz prošlog koraka za HMM s N stanja.

Algoritam „Unaprijed” – indukcija *t*-1 -> *t*

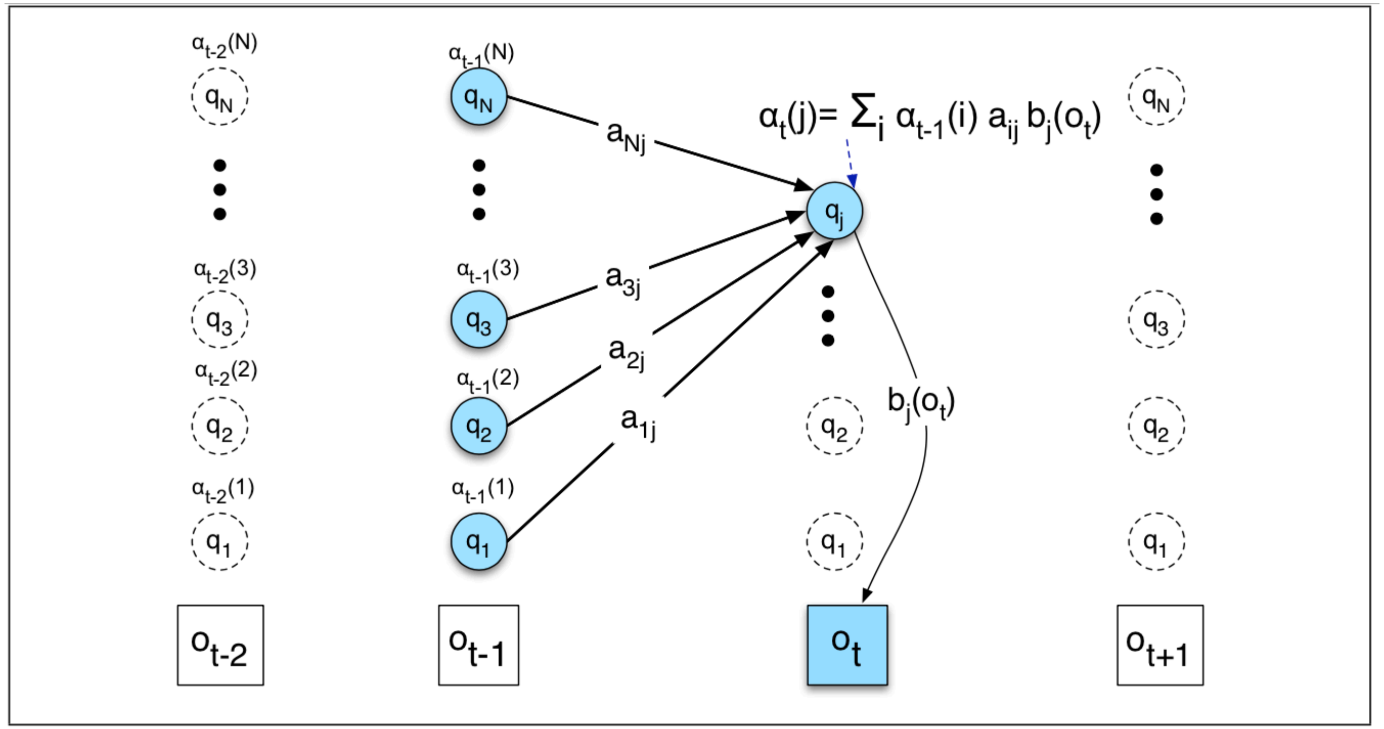


Figure A.6 Visualizing the computation of a single element $\alpha_t(i)$ in the trellis by summing all the previous values α_{t-1}, weighted by their transition probabilities a, and multiplying by the observation probability $b_i(o_t)$. For many applications of HMMs, many of the transition probabilities are 0, so not all previous states will contribute to the forward probability of the current state. Hidden states are in circles, observations in squares. Shaded nodes are included in the probability computation for $\alpha_t(i)$.

Algoritam „Unaprijed” – formalni opis algoritma

function FORWARD(*observations* of len T, *state-graph* of len N) **returns** *forward-prob*

create a probability matrix *forward[N,T]*
for each state s **from** 1 **to** N **do** ; initialization step
$forward[s,1] \leftarrow \pi_s * b_s(o_1)$
for each time step t **from** 2 **to** T **do** ; recursion step
for each state s **from** 1 **to** N **do**

$$forward[s,t] \leftarrow \sum_{s'=1}^{N} forward[s',t-1] * a_{s',s} * b_s(o_t)$$

$$forwardprob \leftarrow \sum_{s=1}^{N} forward[s,T]$$; termination step

return *forwardprob*

Figure A.7 The forward algorithm, where $forward[s,t]$ represents $\alpha_t(s)$.

Algoritam „Unaprijed” – matematički izrazi

1. Initialization:

$$\alpha_1(j) \;=\; \pi_j b_j(o_1) \;\; 1 \leq j \leq N$$

2. Recursion:

$$\alpha_t(j) = \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t); \quad 1 \leq j \leq N, 1 < t \leq T$$

3. Termination:

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

Algoritam „Unaprijed” – završni korak

- Završni korak ovog indukcijskog postupka je zbrajanje svih unaprijednih vjerojatnosti u zadnjem koraku $t=T$, preko svih skrivenih stanja modela $i=1,2,\ldots N$, čime se dobiva tražena izvjesnost osmatranja $P(O|\lambda)$ cijelog niza $O = o_1, o_2, \ldots, o_T$, za zadani HMM model $\lambda = (A, B, \pi)$ i to preko svih mogućih sljedova skrivenih stanja.

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

- Veća vrijednost ove izvjesnosti ukazuje na dobro poklapanje modela i dinamičkih statističkih svojstava osmotrenog niza O koja su opisana parametrima ovog HMM modela $\lambda = (A, B, \pi)$.

Dekodiranje stanja modela Viterbi algoritmom

Zadatak dekodiranja stanja modela

- Za bilo koji model, poput HMM-a, koji sadrži skrivene varijable, zadatak određivanja slijeda tih varijabli koje su temeljni izvor nekog slijeda promatranja naziva se **zadatkom dekodiranja**.
- U primjeru sa „sladoledom”, s obzirom na slijed promatranja broja pojedenih sladoleda „3 1 3“ i uz poznat HMM model λ, zadaća dekodera je **pronaći najbolji skriveni vremenski slijed stanja.**

  … ili formalnije zapisano:

- **Dekodiranje**: Temeljem zadanog HMM modela $\lambda = (A, B, \pi)$ za zadani slijed osmotrenih simbola $O = o_1, o_2, ..., o_T$, pronađite najvjerojatniji slijed skrivenih stanja $Q = q_1, q_2, q_3, ... q_T$.

Zadatak dekodiranja stanja modela

- Kako pronaći taj najbolji slijed skrivenih stanja?
- Mogli bismo izvršiti algoritam Unaprijed i izračunati izvjesnost slijeda promatranja s obzirom na svaki mogući pretpostavljeni slijed stanja, pa bismo na kraju odabrali upravo slijed stanja s najvećom izvjesnosti osmatranja.
- Takav pristup je moguć samo za modele s malim brojem stanja s kratkim nizovima osmatranja, jer je ukupan broj mogućih sljedova stanja N^T, a za svaki moramo odrediti izvjesnost osmatranja.
- Umjesto toga, najčešće korišteni algoritam dekodiranja stanja HMM-a je **Viterbijev algoritam**, kojeg ćemo kraće nazivati **Viterbi**, koji nema eksponencijalnu složenost kao izravni pristup.

Zadatak dekodiranja – Viterbi algoritam

- Viterbi je također vrsta dinamičkog programiranja koje koristi rešetke za dinamičko programiranje, a …
- jako podsjeća na drugu varijantu dinamičkog programiranja, na algoritam najkraćeg puta (engl. *minimum edit distance*).
- Ima izuzetno raširenu primjenu u brojnim inženjerskim područjima.
- Viterbi AJ (April **1967**). "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm". *IEEE Transactions on Information Theory*. **13** (2): 260–269. doi:10.1109/TIT.1967.1054010

- Andrea Giacomo Viterbi, rođ. 9.3.1935, Bergamo, Italija

Viterbi algoritam – pomoću rešetke stanja

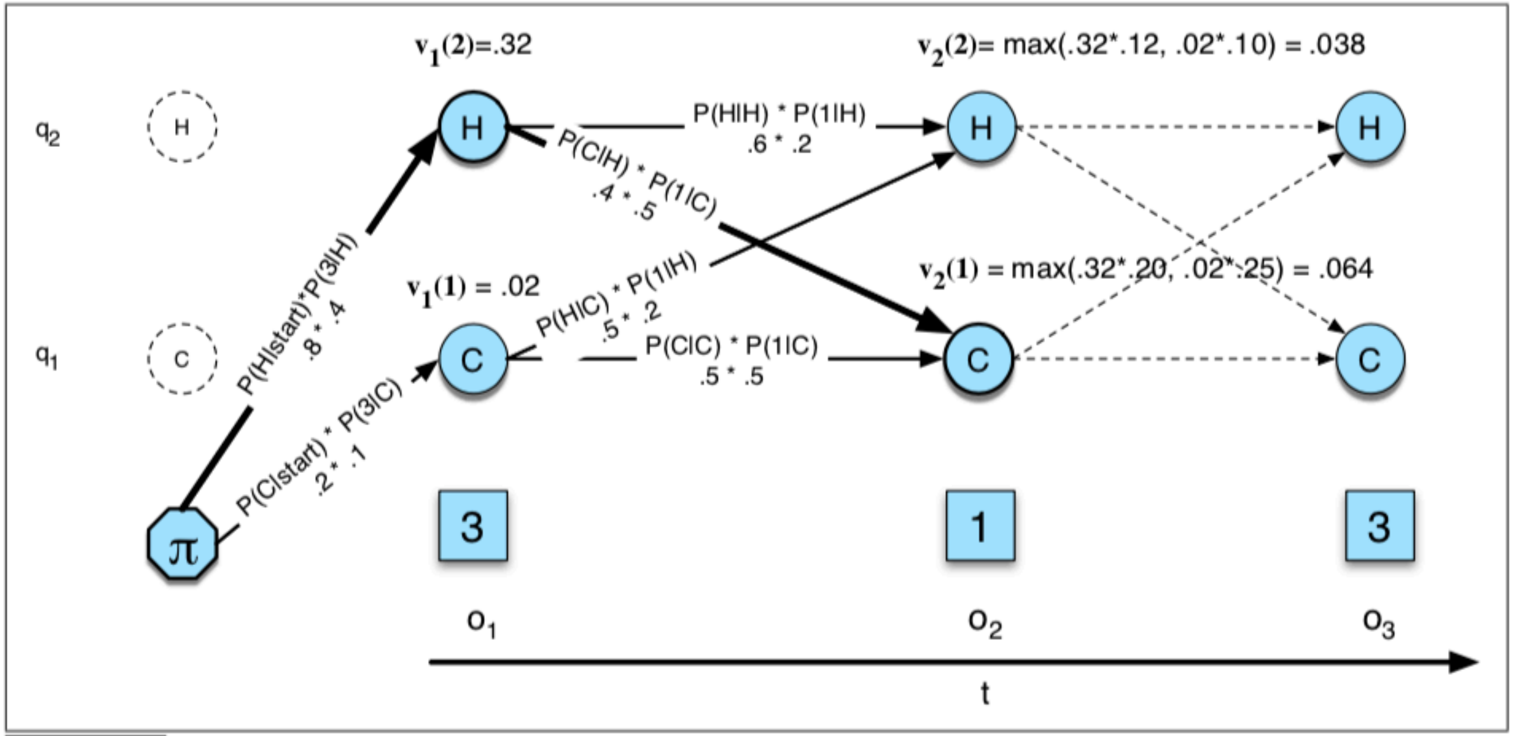


Figure A.8 The Viterbi trellis for computing the best path through the hidden state space for the ice-cream eating events *3 1 3*. Hidden states are in circles, observations in squares. White (unfilled) circles indicate illegal transitions. The figure shows the computation of $v_t(j)$ for two states at two time steps. The computation in each cell follows Eq. A.14: $v_t(j) = \max_{1 \leq i \leq N-1} v_{t-1}(i)\, a_{ij}\, b_j(o_t)$. The resulting probability expressed in each cell is Eq. A.13: $v_t(j) = P(q_0, q_1, \ldots, q_{t-1}, o_1, o_2, \ldots, o_t, q_t = j|\lambda)$.

Viterbi algoritam – pomoću rešetke stanja

- Ideja je obraditi slijed osmatranja slijeva udesno, popunjavajući rešetku, u koju umjesto unaprijednih vjerojatnosti sada upisujemo **Viterbi vjerojatnosti** $v_t(j)$.
- Svaki čvor ove rešetke, $v_t(j)$, predstavlja vjerojatnost da je HMM završio u stanju *j* nakon što je prethodno osmotrio prvih *t* izlaznih simbola i pri čemu je prošao **kroz najvjerojatniji slijed stanja** $q_1, ..., q_{t-1}$, s obzirom na zadani automat λ.
- Vrijednost svakog čvora $v_t(j)$ izračunava se rekurzivnim postupkom, zadržavanjem samo najvjerojatnijeg puta koji nas je kroz rešetku mogao dovesti do ovog novog čvora.

Viterbi algoritam – Viterbi vjerojatnost

- Formalno, svaki čvor izražava ovu **Viterbijevu vjerojatnost**:

$$v_t(j) = \max_{q_1,\ldots,q_{t-1}} P(q_1 \ldots q_{t-1}, o_1, o_2 \ldots o_t, q_t = j|\lambda) \qquad \text{(A.13)}$$

- … prepoznajte iz gornjeg izraza da najvjerojatniji put određujemo nalazeći maksimum vjerojatnosti preko svih mogućih prethodnih sljedova stanja sve do trenutka *t*-1.
- S obzirom na to da smo u prošlom koraku već izračunali vjerojatnosti da se nalazimo u bilo kojem mogućem stanju u trenutku *t*-1, izračunavamo Viterbijevu vjerojatnost u koraku *t* uzimajući **samo najvjerojatnije produljenje putova** koji vode do trenutnog čvora.
- To je osnova visoke učinkovitosti ovog rekurzivnog postupka!

Viterbi algoritam – Viterbi vjerojatnost

- Za pretpostavljeno stanje q_j u trenutku t, vrijednost Viterbi vjerojatnosti $v_t(j)$ u koraku t izračunava se kao:

$$v_t(j) = \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t) \qquad \text{(A.14)}$$

- Tri faktora u svakom umnošku koji se **maksimiziraju** u jednadžbi A.14, a koji se koriste za produljenje prethodnih putova u svrhu izračuna Viterbijeve vjerojatnost u trenutku t su:
- $v_{t-1}(i)$ … **prethodna Viterbijeva vjerojatnost** iz prethodnog vremenskog koraka t-1 u svim stanjima modela,
- a_{ij} ……… **vjerojatnost prijelaza** iz prethodnog stanja q_i u trenutno stanje q_j,
- $b_j(o_t)$ …. **izvjesnost osmatranja** izlaznog simbola o_t u trenutačnom stanju j.

Viterbi algoritam– formalni opis algoritma

function VITERBI(*observations* of len T,*state-graph* of len N) **returns** *best-path*, *path-prob*

create a path probability matrix *viterbi[N,T]*

for each state s **from** 1 **to** N **do** ; initialization step

$\quad viterbi[s,1] \leftarrow \pi_s * b_s(o_1)$

$\quad backpointer[\mathrm{s},1] \leftarrow 0$

for each time step t **from** 2 **to** T **do** ; recursion step

  for each state s **from** 1 **to** N **do**

$$viterbi[\mathrm{s,t}] \leftarrow \max_{s'=1}^{N} viterbi[s', t-1] * a_{s',s} * b_s(o_t)$$

$$backpointer[\mathrm{s,t}] \leftarrow \operatorname{argmax}_{s'=1}^{N} viterbi[s', t-1] * a_{s',s} * b_s(o_t)$$

$bestpathprob \leftarrow \max_{s=1}^{N} viterbi[s, T]$; termination step

$bestpathpointer \leftarrow \operatorname{argmax}_{s=1}^{N} viterbi[s, T]$; termination step

bestpath ← the path starting at state *bestpathpointer*, that follows backpointer[] to states back in time

return *bestpath*, *bestpathprob*

Viterbi algoritam– formalni opis algoritma

- Prethodna slika prikazuje pseudokôd za Viterbijev algoritam.
- Imajte na umu da je Viterbijev algoritam identičan algoritmu „Unaprijed", osim što uzima samo **maksimum vjerojatnosti preko prethodnih puteva** u rešetci, dok algoritam „Unaprijed" uzima zbroj vjerojatnosti preko svih puteva.
- Viterbijev algoritam ima jednu dodatnu komponentu koju algoritam „Unaprijed" nema: ima tzv. **povratne pokazivače**.
- Dok algoritam „Unaprijed" treba izračunati samo ukupnu izvjesnost promatranja zadanog modela preko svih puteva, Viterbijev algoritam mora izračunati vjerojatnost uzduž najboljeg puta, ali istovremeno i **najvjerojatniji slijed stanja** (najbolji put kroz rešetku).

Viterbi algoritam– povratni pokazivači

- Najbolju sekvencu stanja nalazimo prateći put skrivenih stanja koja su vodila do svakog stanja, kao što je ilustrirano na slici A.10, a zatim na kraju kad smo došli do zadnjeg osmotrenog simbola o_T, i time završili u najvjerojatnijem završnom stanju q_T, vraćamo se unazad do prvog stanja, ali prateći samo najbolji put do početka, tzv. **Viterbijev trag rešetke**.
- Zato, za svaki čvor rešetke pored Viterbijeve vjerojatnosti $v_t(j)$ moramo bilježiti i **povratni pokazivač** koji povezuje taj čvor s pripadajućim stanjem u prethodnom koraku **iz kojeg smo došli s najvećom vjerojatnosti**.
- Takav najbolji identificirani povratni put je prikazan podebljanim strelicama u rešetci na slici A.10 uz pretpostavku skraćenog niza (T=2).

Viterbi algoritam – povratni pokazivači & opt. put

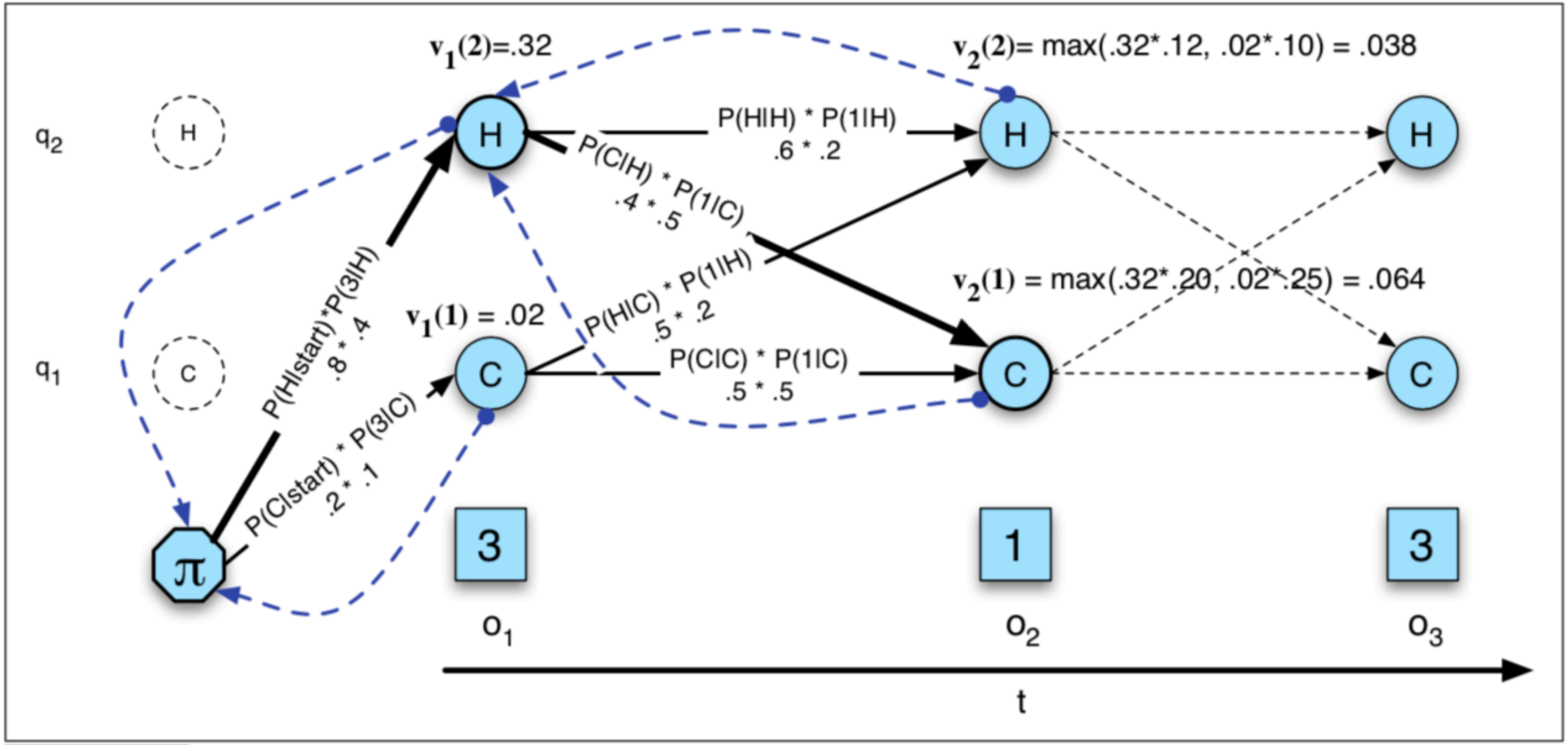


Figure A.10 The Viterbi backtrace. As we extend each path to a new state account for the next observation, we keep a backpointer (shown with broken lines) to the best path that led us to this state.

Viterbi algoritam – pomoću rešetke stanja

- Kako riječima tumačimo prikazane vrijednosti u rešetci stanja u A10?
- HMM mreža kreće iz početnog stanja π (**start**) u kojem se sustav nalazi prije pojave prvog izlaznog simbola, te prelazi u stanje **H** (q_1=2) s vjerojatnosti 0.8 ili u stanje **C** (q_1=1) s vjerojatnosti 0.2.
- U prvom vremenskom trenutku t=1 osmotren je simbol o_1=3, pa će vjerojatnost da se mreža zatekla u stanju **H** biti $\pi_2 * b_2(o_1=3)$ = 0.8*0.4=0.32, a vjerojatnost da se po isteku tok događaja zatekla u stanju **C** biti jednaka $\pi_1 * b_1(o_1=3)$ = 0.2*0.1=0.02.
- Te su Viterbi vjerojatnosti na slici označene kao $v_1(2)$=0.32, odnosno $v_1(1)$=0.02, pri čemu subskript 1 označava prvi vremenski trenutak t=1.
- Ovaj dio opisa identičan je algoritmu „Unaprijed“ .

Viterbi algoritam – pomoću rešetke stanja

- Sada ove dvije vrijednosti koristimo za izračun $v_2(2)$ i $v_2(1)$ u drugom vremenskom koraku temeljem matrice prijelaza *A,* te poznavanjem vjerojatnosti osmatranja drugog izlaznog simbola o_2=1 u svakom od dva moguća stanja (**H** i **C**) koje su definirane u matrici izlaznih vjerojatnosti *B*.
- Konkretno, u stanju **H** u koraku t=2 možemo završiti preko dva moguća puta: ravnog puta s vjerojatnošću $v_1(2) * a_{22} * b_2(o_2=1) = 0.32 * 0.6 * 0.2$, ili preko dijagonalnog puta iz stanja **C** s vjerojatnosti $v_1(1) * a_{12} * b_2(o_2=1) = 0.02 * 0.5 * 0.2$.
- Umjesto zbrajanja ove dvije vrijednosti kao što smo radili kod algoritma „Unaprijed“ **uzimamo samo veću** od ove dvije vjerojatnosti i dobivamo $v_2(2)$=0.038, te biramo samo horizontalnu granu.

Viterbi algoritam – pomoću rešetke stanja

- Analogno maksimiziranjem vjerojatnosti za dva moguća puta koji završavaju u stanju **C** u koraku *t*=2 dobivamo vjerojatnost $v_2(1)=0.064$ uz korištenje dijagonalnog puta.
- Znači, da je uz zadani model λ, vjerojatnost opažanja prva dva simbola „3 1“ jednaka ili $v_2(2)=0.038$ ili $v_2(1)=0.064$, ovisno o tome u kojem smo stanju završili u trenutku *t*=2.
- Da je to ujedno bio i zadnji simbol osmotrenog slijeda *O*, odabrali bismo samo put najveće vjerojatnosti koji završava u stanju **C**, jer $v_2(1)>v_2(2)$, a kojem je prema povratnom pokazivaču prethodilo stanje **H** u koraku *t*=1, što daje najbolji put za prva dva osmotrena simbola ($q_1=2$ & $q_2=1$), a označen je debelim strelicama na A10.
- Isti postupak ponavljamo i za treći osmotreni simbol $o_3=3$.

Viterbi algoritam – parcijalni put stanja?

- Nađeno parcijalno rješenje za najbolji slijed skrivenih stanja do koraka *t*=2 temeljem prva dva osmotrena simbola **ne mora nužno biti i najvjerojatniji slijed za cijeli osmotreni niz**!
- Može se dogoditi da čvor s najvećom Viterbi vjerojatnosti u koraku *t*=3 kao prethodno stanje prozove stanje **H**, pa bi tada prva dva stanja u skladu s povratnim pokazivačima bila **H H**, a ne **H L**, kao što prozivaju samo prva dva osmotrena simbola.
- Stoga je nužno ovaj rekurzivni algoritam **izvršiti sve do zadnjeg osmotrenog simbola** i tek se tada iz završnog stanja najviše Viterbi vjerojatnosti vratiti do prvog trenutka korištenjem povratnih pokazivača koje smo izgradili u rekurzivnom postupku.
- Korištenje parcijalnog optimalnog puta može biti „**kratkovidno**”!

Viterbi algoritam– svi matematički izrazi

1. **Initialization:**

$$
\begin{aligned}
v_1(j) &= \pi_j b_j(o_1) \qquad 1 \le j \le N \\
bt_1(j) &= 0 \qquad\qquad\quad\; 1 \le j \le N
\end{aligned}
$$

2. **Recursion**

$$
\begin{aligned}
v_t(j) &= \max_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t); \quad 1 \le j \le N, 1 < t \le T \\
bt_t(j) &= \operatorname*{argmax}_{i=1}^{N} v_{t-1}(i)\, a_{ij}\, b_j(o_t); \quad 1 \le j \le N, 1 < t \le T
\end{aligned}
$$

3. **Termination:**

$$
\begin{aligned}
\text{The best score:} \quad P* &= \max_{i=1}^{N} v_T(i) \\
\text{The start of backtrace:} \quad q_T* &= \operatorname*{argmax}_{i=1}^{N} v_T(i)
\end{aligned}
$$

Treniranje modela algoritmom „Unaprijed – unazad”

Treniranje HMM modela

- Od svih do sada prikazanih zadataka vezanih uz Markovljeve modele, ovaj je najsloženiji.
- **Treniranje**: S obzirom na slijed promatranja *O* i skup mogućih stanja HMM-a, naučite HMM parametre *A* i *B* korištenjem odabranog kriterija.
- Ulaz u algoritam:
  - neoznačeni slijed promatranja *O,* i
  - isključivo rječnik potencijalnih skrivenih stanja *Q*, ali ne i sam niz stanja.
- … za primjer zadatka „sladoleda” ulaz bi bio slijed promatranja, tj. broj pojedenih sladoleda O = {1, 3, 2, 3, ...,} uz skup skrivenih stanja **H** i **C**.

Baum – Welch algoritam („Unaprijed-unazad”)

- Standardni algoritam za treniranje HMM modela je algoritam „Unaprijed-unazad“ koji je poznat pod nazivom Baum-Welch algoritam (Baum, 1972) prema prezimenima autora

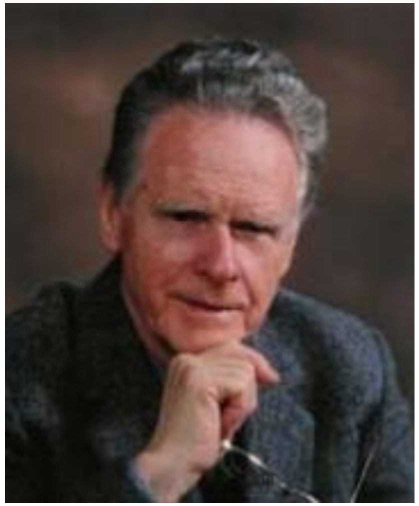
- Baum, L. E. (1972). An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. In Shisha, O. (Ed.), *Inequalities III: Proceedings of the 3rd Symposium on Inequalities*, 1–8. Academic Press.

- Leonard Esau **Baum** (23.8.1931 –14.8.2017.) American mathematician

- Lloyd Richard **Welch** (28.9.1927. -) American information theorist

Baum – Welch algoritam („Unaprijed-unazad”)

- Predstavlja poseban slučaj EM algoritma maksimizacije očekivanja (*engl. Expectation Maximization, EM*) (Dempster i sur., 1977).
- Omogućiti će treniranje matrice prijelaznih vjerojatnosti *A* i matrice vjerojatnosti osmatranja izlaznih simbola *B* HMM-a.
- EM je iterativni algoritam, koji prvo izračunava početnu procjenu vjerojatnosti, zatim koristeći te procjene izračunava bolje procjene, i tako dalje, iterativno poboljšavajući vjerojatnosti koje nauči.



- Arthur Pentland Dempster (born 1929), Professor Emeritus in the Harvard University Department of Statistics

Treniranje HMM modela

- Da bi objasnili kako ovaj algoritam radi, krenimo od jednog puno jednostavnijeg slučaja učenja potpuno osmotrivog Markovljevog modela, tj.,
  - pretpostavimo da na neki način poznamo i temperaturu (stanje) i broj pojedenih sladoleda (opservaciju) za svaki dan za tri niza osmatranja.

3	3	2		1	1	2		1	2	3
hot	hot	cold		cold	cold	cold		cold	hot	hot

- Poznavanjem ovog „poravnanja” stanja i opservacija, jednostavno možemo izračunati parametre HMM-a estimacijom najveće izvjesnosti iz skupa ulaznih podataka za trening, jednostavnim prebrojavanjem.

Treniranje HMM modela – vektor π

- Odredimo prvo vektor π početnih vjerojatnosti stanja

```
3    3   2          1    1    2          1    2   3
hot  hot cold       cold cold cold       cold hot hot
```

- Vidimo da se u prvom koraku t=1 pojavljuje stanje **hot** u jednom od tri slučaja, a **cold** u preostala dva od tri slučaja, pa razdiobu vjerojatnosti početnog stanja možemo prilagoditi raspoloživim slučajnim ulaznim uzorcima koje koristimo za treniranje:

$$\pi_h = 1/3 \text{ i } \pi_c = 2/3$$

Treniranje HMM modela – matrica A

- Izračunajmo matricu *A* iz svih mogućih prijelaza stanja za ova tri slijeda osmatranja tako da jednostavno prebrojimo sve promjene stanja iz prvog u drugi korak i iz drugog u treći korak …

3	3	2	1	1	2	1	2	3
hot	hot	cold	cold	cold	cold	cold	hot	hot

- Model je dva puta prešao iz stanja hot u hot i jednom iz stanja hot u stanje cold, pa formiramo prve dvije vjerojatnosti prijelaza:

$$p(\text{hot}|\text{hot}) = 2/3 \quad p(\text{cold}|\text{hot}) = 1/3$$

- Analogno, model je dva puta prešao iz stanja cold u cold i jednom iz stanja cold u stanje hot, pa nalazimo i druge dvije vjerojatnosti prijelaza:

$$p(\text{cold}|\text{cold}) = 2/3 \quad p(\text{hot}|\text{cold}) = 1/3$$

Treniranje HMM modela – matrica B

- Na sličan način, prebrojavanjem nalazimo i emisijske vjerojatnosti matrice *B* HMM modela:

3	3	2	1	1	2	1	2	3
hot	hot	cold	cold	cold	cold	cold	hot	hot

$p(1|\text{hot}) = 0/4 = 0$ $\qquad$ $p(1|\text{cold}) = 3/5 = .6$

$p(2|\text{hot}) = 1/4 = .25$ $\qquad$ $p(2|\text{cold} = 2/5 = .4$

$p(3|\text{hot}) = 3/4 = .75$ $\qquad$ $p(3|\text{cold}) = 0$

Treniranje HMM modela

- Za stvarni HMM ne možemo izračunati te vjerojatnosti prebrojavanjem izravno iz ulaznih nizova osmatranja jer nažalost ne znamo koji je put stanja prošao model za zadani ulaz.

- Na primjer, pretpostavimo da nismo poznavali temperaturu samo 2. dana i da ste je morali na neki način pogoditi, ali ste (magično) ipak imali gornje vjerojatnosti i temperature svih ostalih dana.
- Mogli biste primijeniti Bayesovu aritmetiku temeljenu na poznatim vjerojatnostima kako bi dobili procjene vjerojatne temperature tog dana koji nedostaje, a zatim pomoću njih dobiti i očekivane brojeve pojava visokih i niskih temperatura za taj 2. dan.

Treniranje HMM modela – inicijalizacija BW alg.

- Međutim stvarni je problem još bitno teži jer ne znamo statistiku niti za jedno od skrivenih stanja!!
- Baum-Welchov algoritam to rješava *iterativnom procjenom statistike*.
- Započet ćemo s **proizvoljnom (slučajnom) procjenom vjerojatnosti prijelaza i opservacija**, a zatim ćemo koristiti te procijenjene vjerojatnosti za izvođenje sve boljih vjerojatnosti.
- To ćemo učiniti izračunavanjem izvjesnosti osmatranja *O* algoritmom „Unaprijed” i zatim raspodjelom te ukupne mase vjerojatnosti između svih mogućih putova koji su pridonijeli toj ukupnoj unaprijednoj vjerojatnosti modela s trenutačnim parametrima *A* i *B*.
- Da bismo razumjeli algoritam, moramo uvesti još jednu korisnu vjerojatnost koja je povezana s unaprijednom vjerojatnosti, a to je **unazadna vjerojatnost** (*engl. backward probability*).

Unazadna vjerojatnost HMM modela – algoritam „Unazad”

Unazadna vjerojatnost HMM-a

- Potpuno analogno s definicijom unaprijedne vjerojatnosti, **unazadna vjerojatnost β** bit će jednaka vjerojatnost osmatranja zadanog niza od vremenskog trenutka *t*+1 do kraja niza, ali uz pretpostavku da smo bili u stanju *i* u trenutku *t* (i to s obzirom na zadani automat *λ*):

$$\beta_t(i) = P(o_{t+1}, o_{t+2} \ldots o_T | q_t = i, \lambda) \tag{A.15}$$

- Slično kao i vjerojatnost unaprijed, i ova se vjerojatnost unazad može izračunati iterativnim postupkom koji kreće od zadnjeg osmotrenog simbola u trenutku *t*=*T*, te se zatim vraća korak po korak unazad sve do željenog trenutka *t* (koji može biti i *t*=1, ako želimo odrediti ukupnu izvjesnost cijelog osmatranja O za zadani model).

Algoritam „Unazad” – matematički izrazi

1. **Initialization:**

$$\beta_T(i) \;=\; 1, \quad 1 \le i \le N$$

2. **Recursion**

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(o_{t+1})\, \beta_{t+1}(j), \quad 1 \le i \le N, 1 \le t < T$$

3. **Termination:**

$$P(O|\lambda) = \sum_{j=1}^{N} \pi_j\, b_j(o_1)\, \beta_1(j)$$

Algoritam „Unazad” – indukcija t+1 -> t

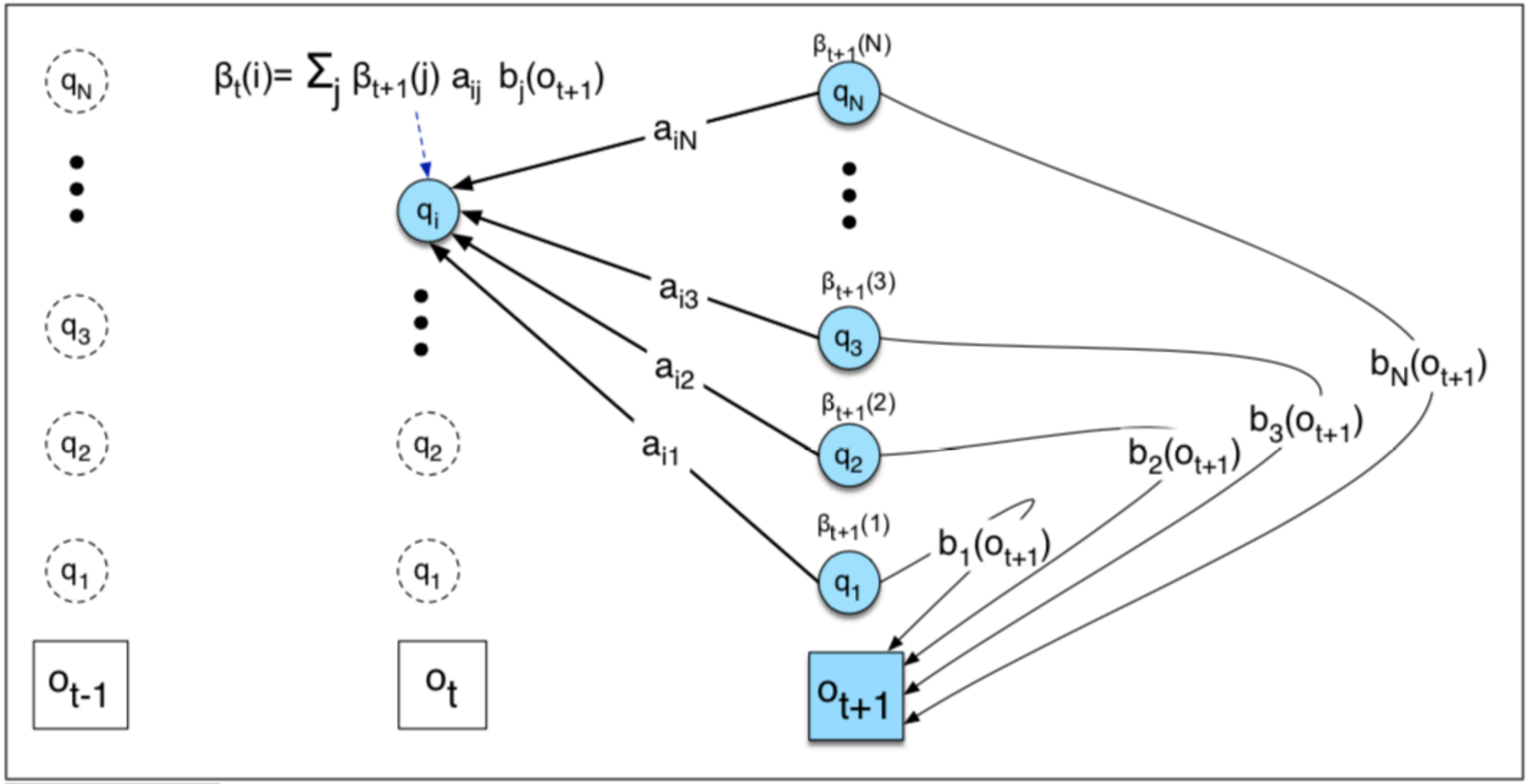


Figure A.11 The computation of $\beta_t(i)$ by summing all the successive values $\beta_{t+1}(j)$ weighted by their transition probabilities a_{ij} and their observation probabilities $b_j(o_{t+1})$. Start and end states not shown.

Algoritam „Unaprijed-unazad” za treniranje HMM modela

Treniranje HMM modela

- Sada smo spremni istražiti kako vjerojatnosti unaprijed i unazad mogu pomoći u izračunavanju vjerojatnosti prijelaza a_{ij} i vjerojatnosti osmatranja izlaznog simbola o_t u stanju i, $b_i(o_t)$ iz niza promatranja, iako je stvarni put kroz niz stanja modela nepoznat (skriven).
- Počnimo s prikazom kako procijeniti vjerojatnost prijelaza a_{ij} varijantom jednostavne estimacije maksimalne izvjesnosti:

$$\hat{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} \tag{A.16}$$

- … dakle kao kvocijent očekivanog broja prijelaza iz stanja *i* u stanje *j* i ukupnog očekivanog broja prijelaza iz stanja *i* u bilo koje stanje.

Treniranje HMM modela – matrica A

- Kako izračunati brojnik ovog izraza?
- Pretpostavimo da smo imali neku procjenu vjerojatnosti da se zadani prijelaz *i* → *j* dogodio u određenom trenutku u vremenu *t* u slijedu promatranja.
- Kad bismo znali tu vjerojatnost za svaki trenutak *t*, mogli bismo jednostavno zbrojiti prijelaze preko svih vremenskih trenutaka kako bismo procijenili ukupan broj prijelaza *i* → *j*.
- Formalnije, … definirajmo vjerojatnost ξ_t (*ksi*) kao vjerojatnost da budemo u stanju *i* u trenutku *t*, a stanju *j* u trenutku *t*+1, s obzirom na slijed osmotrenih simbola *O* i naravno s obzirom na aktualni model λ:

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j | O, \lambda) \qquad \text{(A.17)}$$

Treniranje HMM modela – matrica A

- Da bismo odredili ξ_t prvo izračunavamo jednu pomoćnu vjerojatnost koja je slična ξ_t, ali se razlikuje po tome što uključuje i vjerojatnost opažanja *O*.

$$\text{not-quite-}\xi_t(i,j) = P(q_t = i, q_{t+1} = j, O|\lambda) \qquad \text{(A.18)}$$

- Uočite različito iskazanu uvjetnu vjerojatnost u izrazima A.17 i A.18!
- Slika A.12 prikazuje razne vjerojatnosti koje sudjeluju u izračunavanju ove pomoćne veličine koju ćemo nazvati „ne baš-ξ_t“, odnosno u izvornom obliku na engleskom jeziku, kao „not-quite-ξ_t“.

Treniranje HMM modela – nalaženje „ne-baš-ξ_t"

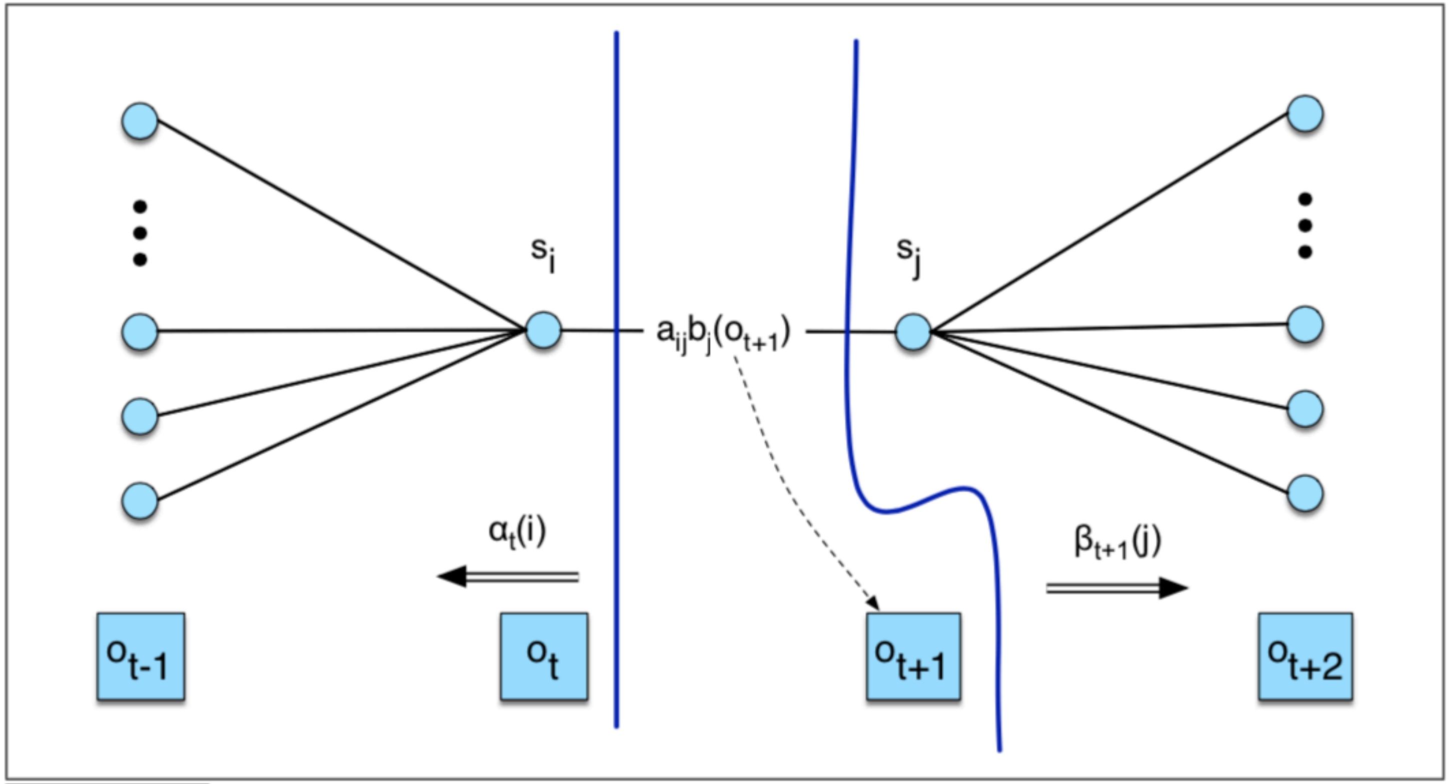


Figure A.12 Computation of the joint probability of being in state i at time t and state j at time $t+1$. The figure shows the various probabilities that need to be combined to produce $P(q_t = i, q_{t+1} = j, O|\lambda)$: the α and β probabilities, the transition probability a_{ij} and the observation probability $b_j(o_{t+1})$. After Rabiner (1989) which is ©1989 IEEE.

Treniranje HMM modela – nalaženje „ne-baš-ξ_t”

- To su:
  - vjerojatnost prijelaza za dotični luk, a_{ij},
  - unaprijedna vjerojatnost α prije luka (izračunata algoritmom „Unaprijed”),
  - unazadna vjerojatnost β nakon luka (izračunata algoritmom „Unazad”) i
  - vjerojatnost izlaznog simbola neposredno nakon luka (u trenutku t+1).
- Ova se četiri faktora množe kako bi se dobila vrijednost ove pomoćne veličine, kako slijedi:

$$\text{not-quite-}\xi_t(i,j) = \alpha_t(i)\, a_{ij}\, b_j(o_{t+1})\, \beta_{t+1}(j) \qquad \text{(A.19)}$$

Treniranje HMM modela – nalaženje ξ_t

- Kako bi u konačnici odredili ξ_t iz ove pomoćne veličine, koristimo dobro poznato pravilo iz teorije vjerojatnosti koje glasi:

$$P(X|Y,Z) = \frac{P(X,Y|Z)}{P(Y|Z)} \tag{A.20}$$

- Dakle zapravo samo treba podijeliti ovu pomoću veličinu s ukupnom izvjesnosti osmotrenog niza O za zadani model λ, $P(O|\lambda)$, koja se na temelju unaprijedne i unazadne vjerojatnosti može naći kao:

$$P(O|\lambda) = \sum_{j=1}^{N} \alpha_t(j)\beta_t(j) \tag{A.21}$$

$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij} b_j(o_{t+1})\beta_{t+1}(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \tag{A.22}$$

Treniranje HMM modela – nalaženje novih a_{ij}

- Očekivani broj prijelaza iz stanja *i* u stanje *j* tada je jednostavno zbroj ξ_t za svaki *t*.
- Za našu procjenu vjerojatnosti prijelaza a_{ij} u jednadžbi A.16, trebamo samo još jednu dodatnu stvar: to je ukupan očekivani broj prijelaza iz stanja *i* u nazivniku tog izraza.
- Njega možemo dobiti zbrajanjem svih prijelaza iz stanja *i* u bilo koje stanje modela, što daje konačnu formulu za procjenu a_{ij}:

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{k=1}^{N} \xi_t(i,k)} \qquad \text{(A.23)}$$

- Prepoznaj da smo za izračun novih vrijednosti $\hat{a}_{ij}$ koristili aktualne vrijednosti istih parametara a_{ij} (dižemo sami sebe za pertle)!

Treniranje HMM modela – nalaženje $b_j(v_k)$

- Sjetimo se da su nam pored članova matrice prijelaza A potrebne i emisijske vjerojatnosti modela u matrici B.
- To je vjerojatnost osmatranja danog simbola v_k iz rječnika izlaznih simbola V, s obzirom na stanje j: $b_j(v_k)$. To ćemo učiniti pokušavajući izračunati ovaj kvocijent:

$$\hat{b}_j(v_k) = \frac{\text{expected number of times in state } j \text{ and observing symbol } v_k}{\text{expected number of times in state } j} \qquad \text{(A.24)}$$

- dakle kao omjer očekivanog broja osmatranja simbola v_k u stanju j u odnosu na ukupni očekivani broj trenutaka u kojima se HMM model nalazio u stanju j. Za to ćemo trebati znati vjerojatnost da budemo u stanju j u trenutku t, a tu pomoćnu vjerojatnost ćemo nazvati $\gamma_t(j)$.

Treniranje HMM modela – nalaženje $\gamma_t(j)$

- Dakle, **pomoćnu vjerojatnost** $\gamma_t(j)$ definiramo kao:

$$\gamma_t(j) = P(q_t = j | O, \lambda) \tag{A.25}$$

- Analogno kao kod izračuna ξ_t, i ovdje ćemo drugačije definirati uvjetnu vjerojatnost, što ćemo kompenzirati dijeljenjem ove modificirane uvjetne vjerojatnosti s izvjesnosti cijelog niza osmotrenih simbola za zadani model, prema izrazu:

$$\gamma_t(j) = \frac{P(q_t = j, O | \lambda)}{P(O | \lambda)} \tag{A.26}$$

Treniranje HMM modela – nalaženje $\gamma_t(j)$

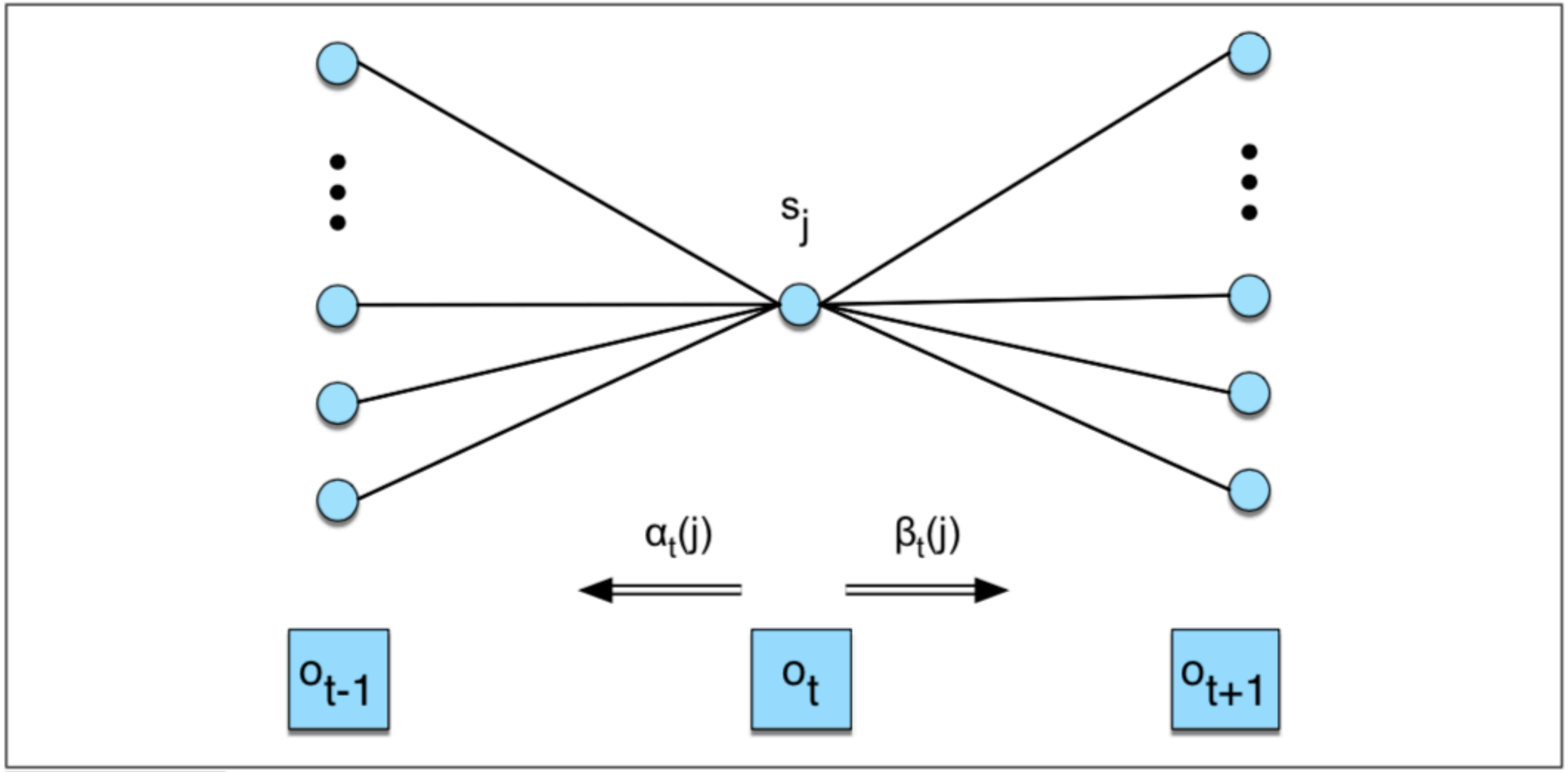


Figure A.13 The computation of $\gamma_t(j)$, the probability of being in state j at time t. Note that γ is really a degenerate case of ξ and hence this figure is like a version of Fig. A.12 with state i collapsed with state j. After Rabiner (1989) which is ©1989 IEEE.

Treniranje HMM modela – nalaženje $\gamma_t(j)$

- Izračun ove pomoćne veličine $\gamma_t(j)$ ilustriran je na slici A.13, pri čemu je brojnik izraza A.26 jednak umnošku vjerojatnosti unaprijed do trenutka *t* i vjerojatnosti unazad od kraja niza do istog trenutka *t*.

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{P(O|\lambda)} \tag{A.27}$$

- Time smo pripremili sve potrebne veličine za izračunavanje emisijskih vjerojatnosti u matrici *B*.
- Za brojnik izraza A.24 zbrajamo $\gamma_t(j)$ za sve vremenske korake *t* u kojima je osmotreni simbol o_t jednak simbolu v_k čiju emisijsku vjerojatnost tražimo. Za nazivnik zbrajamo $\gamma_t(j)$ u svim vremenskim koracima *t*, neovisno od osmotrenog simbola.

Treniranje HMM modela – nalaženje novih $b_j(v_k)$

- Konačni rezultat je udio: … koliko puta kada smo bili u stanju j smo osmotrili baš simbol v_k (oznaka iza sume u brojniku se čita “Zbroj preko svih t za koje je osmotreni simbol u trenutku t bio simbol v_k”):

$$\hat{b}_j(v_k) = \frac{\sum_{t=1\ s.t. O_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \qquad \text{(A.28)}$$

Treniranje HMM modela – iterativna reestimacija

- Jednadžba A.23 (za nove vjerojatnosti prijelaza) i jednadžba A.28 (za nove emisijske vjerojatnosti) omogućavaju ponovnu procjenu vjerojatnosti prijelaza *A* i izlaznih osmatranja *B* iz zadanog niza (jednog ili više) osmotrenih simbola *O*, …
- …. ali pod pretpostavkom da već imamo prethodnu procjenu *A* i *B*, jer smo za izračun ovih novih estimacija morali koristiti pomoćne vjerojatnosti $\xi_t(i,j)$ i $\gamma_t(j)$ koje su bile određene temeljem postojećeg modela $\lambda = (A, B)$.
- Takove ponovne procjene (reestimacije) čine samu srž ovog iterativnog algoritma „Unaprijed-unazad“, pa zato kažemo da se „dižemo za pertle”.

Algoritam „Unaprijed-unazad” – formalni opis

function FORWARD-BACKWARD(*observations* of len *T*, *output vocabulary V*, *hidden state set Q*) **returns** *HMM=(A,B)*

initialize *A* and *B*
iterate until convergence
E-step

$$\gamma_t(j) = \frac{\alpha_t(j)\beta_t(j)}{\alpha_T(q_F)} \quad \forall\, t \text{ and } j$$

$$\xi_t(i,j) = \frac{\alpha_t(i)\, a_{ij} b_j(o_{t+1})\beta_{t+1}(j)}{\alpha_T(q_F)} \quad \forall\, t,\ i, \text{ and } j$$

M-step

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \sum_{k=1}^{N} \xi_t(i,k)}$$

$$\hat{b}_j(v_k) = \frac{\sum_{t=1 s.t.\ O_t=v_k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

return *A*, *B*

Figure A.14 The forward-backward algorithm.

Algoritam „Unaprijed-unazad”

- Algoritam „Unaprijed-unazad“ (prikazan slikom A.14) započinje s nekom početnom procjenom HMM parametara $\lambda = (A, B)$.
- Zatim, iterativno ponavljamo dva koraka. Kao i kod drugih primjena EM algoritma (algoritma maksimizacije očekivanja), opisani algoritam „Unaprijed-unazad“ koji također pripada istoj porodici algoritama ima dva ključna koraka:
- **korak očekivanja** ili tzv. *E-korak* i **korak maksimizacije** ili tzv. *M-korak*.
- U *E-koraku* izračunavamo očekivani broj zauzetosti stanja $\gamma_t(j)$ i očekivani broj prijelaza stanja ξ_t iz „fiksiranih“ vjerojatnosti *A* i *B* određenih u prošlom koraku. U *M-koraku* koristimo $\gamma_t(j)$ i ξ_t za ponovno izračunavanje novih vjerojatnosti *A* i *B*, koje zatim „fiksiramo“ i koristimo u narednom *E-koraku*.

Algoritam „Unaprijed-unazad”

- Iako se u principu algoritmom „Unaprijed-unazad“ može provesti potpuno nenadzirano učenje parametara *A* i *B*, u praksi su početni uvjeti vrlo važni.
- Iz tog razloga algoritmu se često prosljeđuju dodatne informacije.
- Na primjer, za automatsko prepoznavanje govora temeljeno na HMM-u, struktura HMM-a se često postavlja ručno, a samo se emisijske vjerojatnosti *B* i vjerojatnosti prijelaza *A* (koje ne smiju biti jednake nula) treniraju iz niza sekvenci osmatranja *O*.

Sažetak prezentacije

Skriveni Markovljevi modeli

- U ovoj prezentaciji smo predstavili **skrivene Markovljeve model**e za **vjerojatnosnu klasifikaciju opservacija** proizvoljnih duljina, ali isključivo diskretnog rječnika izlaznih simbola, tzv. diskretne HMM-ove.
- Skriveni Markovljevi modeli (HMM) način su formalnog povezivanja niza **osmotrenih simbola** s nizom **skrivenih klasa** ili **skrivenih stanja** koja objašnjavaju ova promatranja.
- Proces otkrivanja slijeda skrivenih stanja, s obzirom na slijed osmatranja, poznat je kao **dekodiranje ili zaključivanje**. Predstavljen je **Viterbijev** algoritam koji se obično koristi u tu svrhu.
- Parametri HMM-a su matrica vjerojatnosti prijelaza *A* i matrica emisijskih vjerojatnosti *B*. Oboje se mogu trenirati s algoritmom **Baum-Welch** ili algoritmom „**Unaprijed-unazad**“.

Bibliografske i povijesne napomene

Skriveni Markovljevi modeli- povijest

- Markovljeve lance prvi je upotrijebio Andrey Andreyevich Markov (1913), *Андре́й Андре́евич Ма́рков*, (prijevod Markov 2006), da bi predvidio hoće li nadolazeće slovo u Puškinovom *Evgenjinu Onjeginu* biti samoglasnik ili suglasnik.
- Skrivene Markovljeve modele razvili su Baum i kolege iz Instituta za obrambene analize u Princetonu (Baum i Petrie 1966, Baum i Eagon 1967).

- Viterbijev algoritam prvi je primijenio na obradu govora i jezika u kontekstu prepoznavanja govora Vintsyuk (1968), ali algoritam ima ono što Kruskal (1983) naziva "izvanrednom poviješću višestrukog neovisnog otkrivanja i objavljivanja".

Skriveni Markovljevi modeli - povijest

- Kruskal i drugi daju barem sljedeće neovisno otkrivene inačice istog Viterbi algoritma objavljene u četiri zasebna polja:

Citation	Field
Viterbi (1967)	information theory
Vintsyuk (1968)	speech processing
Needleman and Wunsch (1970)	molecular biology
Sakoe and Chiba (1971)	speech processing
Sankoff (1972)	molecular biology
Reichert et al. (1973)	molecular biology
Wagner and Fischer (1974)	computer science

… što pokazuje da su svi došli na istu ideju u gotovo istom trenutku povijesti.

Skriveni Markovljevi modeli - povijest

- Upotreba izraza *Viterbi* danas je standardna za primjenu *dinamičkog programiranja* na bilo koju vrstu vjerojatnosnog problema maksimizacije u obradi govora i jezika.
- Za ne-vjerojatnosne probleme (kao što je *algoritam minimalne udaljenosti uređivanja*), često se koristi uobičajeni naziv - *dinamičko programiranje*.
- Forney, mlađi (1973.) napisao je rani istraživački rad koji istražuje podrijetlo Viterbijevog algoritma u kontekstu teorije informacija i komunikacija.

Skriveni Markovljevi modeli - povijest

- Prezentacija osnovne ideje u ovom prikazu da bi skriveni Markovljevi modeli trebali biti karakterizirani s tri temeljna problema preuzeta je prema utjecajnom preglednom radu Rabiner-a (1989.), a koji se i sam temeljio na tutorijalima Jacka Fergusona iz IDA-e 1960-ih.
- Jelinek (1997) i Rabiner i Juang (1993) daju vrlo cjelovite i detaljne opise algoritma „Unaprijed-unazad“ primijenjenog na problem prepoznavanja govora.
- Jelinek (1997) također ukazuje odnos algoritma „Unaprijed-unazad“ i općenitih EM algoritama maksimizacije vjerojatnosti.

Literatura

- **Baum, L. E.** (1972). An inequality and associated maximization technique in statistical estimation for probabilis- tic functions of Markov processes. In Shisha, O. (Ed.), *Inequalities III: Proceedings of the 3rd Symposium on In- equalities*, 1–8. Academic Press.
- **Baum, L. E. and Eagon, J. A.** (1967). An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, *73*(3), 360– 363.
- **Baum, L. E. and Petrie, T.** (1966). Statistical inference for probabilistic functions of finite-state Markov chains. *Annals of Mathematical Statistics*, *37*(6), 1554–1563.
- **Dempster, A. P., Laird, N. M., and Rubin, D. B.** (1977). Maximum likelihood from incomplete data via the *EM* algorithm. *Journal of the Royal Statistical Society*, *39*(1), 1–21.
- **Eisner, J.** (2002). An interactive spreadsheet for teaching the forward-backward algorithm. In *Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teach- ing NLP and CL*, 10–18.
- **Forney, Jr., G. D.** (1973). The Viterbi algorithm. *Proceedings of the IEEE*, *61*(3), 268–278.

- **Jelinek, F.** (1997). *Statistical Methods for Speech Recognition*. MIT Press.
- **Kruskal, J. B.** (1983). An overview of sequence comparison. In Sankoff, D. and Kruskal, J. B. (Eds.), *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, 1–44. Addison-Wesley.
- **Markov, A. A.** (1913). Essai d'une recherche statistique sur le texte du roman "Eugene Onegin" illustrant la liaison des epreuve en chain. *Izvistia Imperatorskoi Akademii Nauk (Bulletin de l'Acade´mie Impe´riale des Sciences de St.-Pe´tersbourg)*, *7*, 153–162.
- **Markov, A. A.** (2006). Classical text in translation: A. A. Markov, an example of statistical investigation of the text Eugene Onegin concerning the connection of samples in chains. *Science in Context*, *19*(4), 591–600. Translated by David Link.
- **Needleman, S. B. and Wunsch, C. D.** (1970). A general method applicable to the search for similarities in the amino-acid sequence of two proteins. *Journal of Molecular Biology*, *48*, 443–453.
- **Rabiner, L. R.** (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286

- **Rabiner, L. R. and Juang, B. H.** (1993). *Fundamentals of Speech Recognition*. Prentice Hall.
- **Reichert, T. A., Cohen, D. N., and Wong, A. K. C.** (1973). An application of information theory to genetic mutations and the matching of polypeptide sequences. *Journal of Theoretical Biology*, *42*, 245–261.
- **Sakoe, H. and Chiba, S.** (1971). A dynamic programming approach to continuous speech recognition. In *Proceedings of the Seventh International Congress on Acoustics*, Vol. 3, 65–69. Akadémiai Kiadó.
- **Sankoff, D.** (1972). Matching sequences under deletion- insertion constraints. *Proceedings of the National Academy of Sciences*, *69*, 4–6.
- **Vintsyuk, T. K.** (1968). Speech discrimination by dynamic programming. *Cybernetics*, *4*(1), 52–57. Russian Kibernetika 4(1):81-88. 1968.
- **Viterbi, A. J.** (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, *IT-13*(2), 260–269.
- **Wagner, R. A. and Fischer, M. J.** (1974). The string-to-string correction problem. *Journal of the Association for Computing Machinery*, *21*, 168–173.