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

Inačica 0.9

Zagreb, studeni 2020.

Sadržaj

Uvod Markovljeve modele i njihovu primjenu

Vrlo jednostavan statistički model za modeliranje dinamičkih stohastičkih procesa su Markovljevi lanci i skriveni Markovljevi modeli (engl. Hidden Markov Model, HMM). Nalaze primjenu u brojnim područjima inženjerstva, bioinženjerstva i bioinformatike, ali i u modeliranju ekonomskih i društvenih procesa. HMM-ovi se mogu primijeniti u mnogim poljima gdje je cilj identificirati slijed podataka koji nije odmah osmotriv (ali ostali podaci koji ovise o tom nepoznatom slijedu jesu). Primjene uključuju: računalne financije, kinetičku analizu jednostruke molekule, kriptoanalizu, prepoznavanje i sintezu govora, uključujući sve današnje digitalne osobne asistente temeljene na govornoj komunikaciji, automatsko označavanje dijelova govora, klasifikacija dokumenata u rješenjima za skeniranje, strojno prevođenje jezika, predviđanje gena, prepoznavanje rukopisa, poravnavanje bioloških sekvenci, analizu proizvoljnih vremenskih serija, prepoznavanje aktivnosti, niz primjena u bioinformatici, kao sklapanje proteina, klasifikacija redoslijeda, otkrivanje metamorfnih virusa, otkrivanje motiva DNK, kinetika hibridizacije DNA, otkrivanje stanja kromatina, ili pak u modeliranju i predviđanju prijevoza / prometa.

Njihova posebna privlačnost jest u činjenici da su u osnovi vrlo jednostavan i intuitivan statistički alat, a mogu se postupcima učenja / treniranja prilagoditi bilo kakvom dinamičkom stohastičkom procesu. U osnovi se radi o parametarskom modelu, pa jednom izgrađen (utreniran) model možemo koristiti za zadaće klasifikacije. Nepoznati niz osmatranja proizvoljne duljine za kojeg pretpostavljamo da statistički odgovara jednom od izgrađenih modela možemo testirati (evaluirati) sa svakim od izgrađenih modela i kao klasu odabrati onaj model za kojeg je izvjesnost nepoznatog osmatranja najviša.

Primjena HMM-a za modeliranje respiratornih zaraznih bolesti

Danas je iznimno. aktualno pitanje zaraznih respiratornih bolesti, tj. praćenje njihovog širenja i upravljanje s epidemiološkim pojavama prouzročenim Covid-19 virusom. Čak i u tu svrhu se uspješno koriste Markovljevi modeli, jer su sposobni otkriti skrivene informacije koje nisu odmah vidljive, a koje mogu bitno pomoći u kontroli širenja bolesti. Npr. mogli bismo pratiti i kroz dane bilježiti osmotrive simptome bolesti, kao što su povišena tjelesna temperatura, suhi kašalj, tjelesni umor, bol u mišićima i plućima, bol u grlu, dijareja, konjunktivitis, glavobolja, gubitak okusa i mirisa, osip na koži ili promjena boje prstiju na rukama i nogama, te ozbiljne simptome poput otežanog disanja , bolovi ili pritisak ili u prsima, gubitak govora ili pokreta. Temeljem višestrukih nizova takvih osmotrivih simptoma mogli bi izgraditi tri HMM modela, jedan za Covid-19, jedan za običnu prehladu i zadnji za gripu, pri čemu bi u svrhu treniranja za svaki pohranjeni uzorak (niz) morali imati i ručno (točno) označenu klasu pomoću paralelnog pouzdanog testa (npr. PCR za Covid-19). Gradnjom takvih statističkih modela mogli bi se izbjeći troškovi testiranja, jer bi svaki pacijent samo trebao u aplikaciju zapisati svoje simptome. Evaluacijom izvjesnosti njegovih osmotrenih simptoma sa sva tri modela, za Covid-19, za prehladu i za gripu dobili bi informaciju kojem modelu najbolje odgovara konkretni nepoznati niz simptoma.

Primjena HMM-a u digitalnoj obradi govora i obradi prirodnog jezika

Nama su Markovljevi lanci i modeli interesantni prvenstveno u kontekstu digitalne obrade govora i obrade prirodnog jezika u svim mogućim zadaćama koje su vezane uz ovo iznimno široko područje. Danas su takvi sustavi već i iznimno široko rasprostranjeni, jer ih nalazimo u svim pametnim telefonima, ali čak i u mnoštvu IoT uređaja koji nas okružuju, poput pametnih zvučnika i TV-a, pametnih stalaka za suncobran ili pak inteligentnih zahodski školjki koji svi žele razgovarati s vama. Svaka od vodećih svjetskih IT kompanija nudi svoje specifično rješenje u tu svrhu, poput Google Assistant, Cortana (Microsoft), Amazon Alexa, Siri (Apple), Bixby (Samsung) i cijeli niz drugih manje poznatih proizvoda. Jedna od zadaća ovih sustava jest automatsko prepoznavanje govora (engl. Automatic Speech Recognition, ASR), tj. pretvorba zvučnog govornog zapisa u niz riječi, gdje se od ranih početaka razvoja ASR-a koriste upravo Markovljevi modeli. Markovljevi modeli se koriste i za statističko modeliranje jezika, pri objedinjavanju (grupiranju) riječi u rečenice, u svrhu automatskog označavanja dijelova rečenice kao i interpretacije njenog značenja. Koriste se i u primjenama kao što su automatsko prepoznavanje govornika, ili automatsko prepoznavanje jezika.

U kontekstu automatskog prepoznavanja govora iz zvučnog zapisa, taj model treba opisivati dinamička (vremenski promjenjiva) svojstva vektora značajki govornog signala prilikom izgovora cijele riječi, ili ono što je češće u praksi, prilikom izgovora dijela riječi poput jednog sloga. Dakle u ovom primjeru osmotriva veličina je niz vektora značajki govora koji se izračunavaju približno 100 puta u sekundi. Za slog tipičnog trajanja od pola sekunde, dobivamo približno 50-tak vektora koji opisuju spektralna svojstva govora tijekom izgovora tog sloga. Međutim svaki izgovor istog sloga bit će malo drugačiji, čak i kada je izgovoren od strane istog govornika, a još su veće varijacije u slučaju izgovora različitih govornika (pogotovo ovisno o spolu i dobi govornika). Dodatno, varijacije u trajanju izgovora istog sloga mogu biti vrlo velike, što može biti uvjetovano namjernim varijacijama u brzini izgovora, emocionalnim stanjem govornika ili osobnim sklonostima u govoru. Za sve takve varijabilne nizove vektora, varijabilnih duljina, HMM model mora biti uspješan u predviđanju njihove izvjesnosti osmatranja ako se uistinu radi o istom slogu. Prilikom učenja pojedinih modela koristit ćemo višestruke nizove opservacija iste pojave, npr. vektore značajki određene iz različitih snimki istog sloga izgovorenih od različitih govornika. Temeljem tih nizova pokušat ćemo podesiti parametre HMM modela da maksimiziraju izvjesnost osmatranja svih tih nizova za treniranje, u nadi da kada model evaluiramo s novim nizom koji nikada nije korišten u postupku gradnje modela, da će i dalje izvjesnost osmatranja tog novog niza biti najviša upravo s modelom kojem pripada. To znači da će postupak provjere (evaluacije) uspješnosti klasifikacije trebati provesti uvijek s „nepoznatim“ materijalom koji nije korišten u postupku gradnje modela. Cijela ova priča je gotovo identična kao i kod postupaka gradnje i evaluacije neuronskih mreža, ili danas popularnih dubokih mreža, jer svi ovi statistički modeli pripadaju konceptualno istoj grupi podatkovno izgrađenih modela (engl. data driven models). U današnje komercijalne sustave obrade govora i prirodnog jezika ugrađene su vrhunske znanstvene spoznaje kao i njihove vrlo učinkovite algoritamske implementacije, temeljene na računarstvu u oblaku i istovremeno vrlo jednostavnom krajnjem terminalu koji se lako može integrirati u ugradbeni računalni sustav. Kao primjer, dodavanje Amazon Alexa osobnog asistenta u bilo koji IoT uređaj može se ostvariti ugradnjom namjenskog chipa koji košta svega desetak dolara. Iako se svakom od opisanih problema može posvetiti cijeli predmet (ili predmeti), u okviru kolegija Obradba informacija ilustrirat ćemo najjednostavniji HMM model s diskretnim opservacijama. To znači da su simboli koje model generira, odnosno osmotreni simboli koji su ulaz u model, iz konačnog rječnika simbola. Slučaj HMM modela s kontinuiranim gustoćama kakvi se uistinu koriste u ASR sustavima premašuje opseg ovog izlaganja, ali dijeli zajedničke teorijske osnove, pa će ovo biti izvrstan uvod i u složenije HMM modele.

Izvor korišten za pripremu ovog materijala

Tematici Markovljevih i skrivenih Markovljevim modela posvećene su stotine knjiga i preglednih radova pa je raspoloživa literatura uistinu iznimno opsežna i bogata. S obzirom na očekivani opseg izlaganja u okviru predmeta Obradba informacija, kao najpogodniji izvor 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. On je raspoloživ na izravnoj poveznici:

https://web.stanford.edu/~jurafsky/slp3/A.pdf

Riječ je o vrlo sažetom opisu, koji na svega 17 stranica, na iznimno pregledan i jasan način opisuje Markovljeve modele i sve što bi studenti vaše razine obrazovanja o njima trebali znati. Za stvarne primjene HMM modela u području umjetne inteligencije i podatkovne znanosti, sigurno ćete morati proučiti dodatne materijale, ali ključne informacije su opisane upravo u ovom dodatku i njegovom proširenom prijevodu koji je prezentiran u nastavku.

Ranija izdanja iste knjige raspoloživa su i u tiskanom izdanju, ali i u cjelovitom elektroničkom obliku, poput ove poveznice:
http://www.deepsky.com/~merovech/voynich/voynich_manchu_reference_materials/PDFs/jurafsky_martin.pdf

Od narednog poglavlja, ovaj dokument prati izvornik na engleskom jeziku uz identičnu numeraciju poglavlja. Neka poglavlja su proširena s dodatnim opisima slika. Autorska prava vezana uz korištenja ovih materijala i način njihovog citiranja identična su kao i za izvornik. Sve slike i formule izravno su preuzete i izvornika, pa prije njihovog korištenja u drugim dokumentima detaljno istražite sva prava autora vezana uz reprodukciju. Interesantno je napomenuti da je prilikom izrade ovog prijevoda kao pomoć korišten alat za automatsko prevođenje „Google Translate“ raspoloživ kao web aplikacija na poveznici:

https://translate.google.com

Ovaj alat, koji je također temeljen na „cloud“ pristupu, je iz dana u dan sve uspješniji u automatskom prevođenju, a između ostalog je temeljen na sličnim statističkim modelima poput Markovljevih modela koji su predmet ovog izlaganja. Za probu možete pokušati korištenjem ovog alata prevesti odabrana poglavlja iz izvornog dokumenta na engleskom jeziku i usporediti taj prijevod s tekstom u ovom dokumentu. Moći ćete uočiti da su potrebne dodatne ručne ispravke teksta bile uistinu minorne, pa bi studenti koji ne znaju engleski jezik (iako su takvi u današnje vrijeme iznimno rijetki) mogli samostalno prevoditi slične izvornike znanstvene i stručne tematike na hrvatski jezik.

A.1 Markovljev lanac

HMM se temelji na poopćenju Markovljevog lanca. Markovljev lanac je model koji nam govori nešto o vjerojatnostima nizova slučajnih varijabli, tj. stanjima, od kojih svako može poprimiti vrijednosti iz nekog konačnog skupa. Ovi skupovi mogu biti riječi ili oznake ili simboli koji predstavljaju bilo što, poput meteoroloških prilika. Markovljev lanac je temeljen na vrlo čvrstoj pretpostavci da ako želimo predvidjeti budućnost u nizu, važno je samo poznavati isključivo trenutno stanje. Prethodna stanja nemaju utjecaja na budućnost, osim putem trenutnog stanja. Na primjeru predviđanja vremenskih prilika, ta pretpostavka bi značila di bi u svrhu predviđanja sutrašnjih vremenskih prilika, mogli ispitati današnje vrijeme, ali pri tome ne bi smjeli promatrati jučerašnje vrijeme.

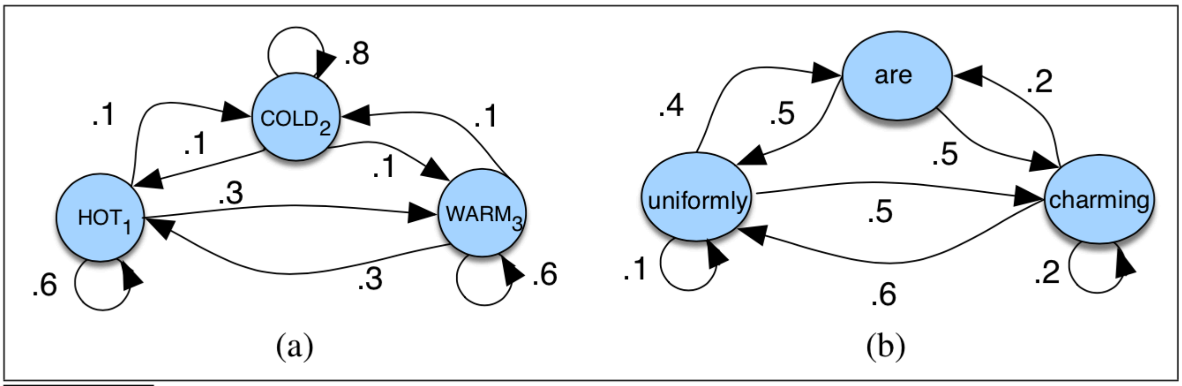

Figure A.1 A Markov chain for weather (a) and one for words (b), showing states and transitions. A start distribution π is required; setting $\pi = [0.1, 0.7, 0.2]$ for (a) would mean a probability 0.7 of starting in state 2 (cold), probability 0.1 of starting in state 1 (hot), etc.

Formalnije, razmotrite niz varijabli stanja $q_1, q_2, \ldots, q_i$. Markovljev model utjelovljuje Markovljevu pretpostavku o vjerojatnostima ovog slijeda: da kada predviđa budućnost, prošlost nije bitna, već samo sadašnjost.

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

Slika A.1a prikazuje Markovljev lanac za dodjeljivanje vjerojatnosti nizu klimatoloških događaja, za koje se rječnik sastoji od HOT, COLD i WARM (VRUĆE, HLADNO i TOPLO). Stanja su na grafu predstavljena kao čvorovi, a prijelazi sa svojim vjerojatnostima kao bridovi. Prijelazi su vjerojatnosti: vrijednosti lukova koji napuštaju određeno stanje moraju se zbrojiti u jedinicu. Slika A.1b prikazuje Markovljev lanac za dodjeljivanje vjerojatnosti nizu riječi w1 ... wn. Ovaj Markovljev lanac bi trebao biti poznat jer zapravo predstavlja bigramski model jezika, koji sa svakim rubom koji izražava uvjetnu vjerojatnost p (wi | wj)! S obzirom na dva modela na slici A.1, možemo dodijeliti vjerojatnost bilo kojem slijedu simbola iz našeg rječnika. Formalno, Markovljev lanac je određen sljedećim komponentama:

$Q = q_1 q_2 \ldots q_N$	a set of N **states**
$A = a_{11} a_{12} \ldots a_{n1} \ldots a_{nn}$	a **transition probability matrix** A, each a_{ij} representing the probability of moving from state i to state j, s.t. $\sum_{j=1}^{n} a_{ij} = 1 \quad \forall i$
$\pi = \pi_1, \pi_2, \ldots, \pi_N$	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, $\sum_{i=1}^{n} \pi_i = 1$

Dakle, Q = q1 q2 ... qN je skup od N mogućih stanja. A = a11 a12 ... an1 ... ann je matrica vjerojatnosti prijelaza A, gdje svaki njen član aij predstavlja vjerojatnost prelaska iz stanja i u stanje j, pri čemu je suma članova aij u svakom retku (tj. za stupce j=1 do N) jednaka jedinici, jer sustav mora ili ostati u istom stanju (i -> i), ili mora prijeći u bilo koje novo stanje (i -> j).

Početna raspodjela vjerojatnosti po stanjima opisana je vektorom π = π1, π2, ..., πN. Vrijednost πi je vjerojatnost da će Markovljev lanac započeti u stanju i. Neka stanja j mogu imati πj = 0, što znači da ne mogu biti početna stanja sustava. Također, suma svih članova ovog vektora mora biti jednaka jedinici (t.j. sustav početno mora biti u nekom od stanja).

Prije nego što nastavimo, za vježbu upotrijebite odabrane uzorke vjerojatnosti sa slike A.1a (uz početne vjerojatnosti stanja π = [.1, .7., 2]) da biste izračunali vjerojatnost svake od sljedećih osmotrenih sekvenci:

- (A.2) vruće vruće vruće vruće
- (A.3) hladno vruće hladno vruće

Što vam razlika u tim vjerojatnostima govori o modelu vremenskih prilika koje su kodirane mrežom na slici A.1a?

A.2 Skriveni Markovljev model

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.
Konkretno sjetimo se gradiva koje smo učili u osnovnoj i srednjoj školi o analizi rečenice. Tada smo naučili da se riječi dijele po vrstama riječi na: imenice, glagole, pridjeve, zamjenice, brojeve, priloge, prijedloge, veznike i uzvike. Također smo naučili da se riječi dijele i s obzirom na zadatak kojeg pojedine riječi nose u rečenici, što je definiranom službom riječi. Prema službi riječi u rečenici dijelimo ih na predikat, subjekt, objekt, priložnu oznaku, atribut i apoziciju.
Međutim, prilikom čitanja u tekstu obično ne uočavamo točne oznake dijela govora, tj. oznake vrsta riječi nisu navedene u tekstu. Umjesto toga, vidimo samo riječi i moramo zaključiti oznake iz niza riječi. Stoga oznake nazivamo skrivenim jer nisu izravno osmotrive.

Skriveni Markovljev model (HMM) omogućuje nam da istovremeno razgovaramo o opaženim događajima (poput niza riječi koje vidimo na ulazu) kao i o skrivenim događajima (poput oznaka dijela govora) za koje smatramo da su uzročni čimbenici u našem vjerojatnosnom modelu. Općenito HMM je određen sljedećim komponentama:

$Q = q_1 q_2 \ldots q_N$	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, $\sum_{i=1}^{n} \pi_i = 1$

Niz stanja Q, prijelazna matrica A i vektor inicijalne razdiobe vjerojatnosti stanja π jednaki su kao i za raniji opis Markovljevog lanca. Međutim sada za skriveni Markovljev model imamo i niz osmotrenih izlaznih simbola O = o1 o2 ... oT kojeg možemo nazvati slijedom opažanja duljine T, pri čemu je svako opažanje izvučeno iz konačnog rječnika simbola V = v1, v2, ..., vV.

Dodatni parametar HMM modela je i matrica vjerojatnosti pojedinog opažanja B = bi (ot) koja opisuje slijed vjerojatnosti promatranja, koje se još nazivaju i emisijskim vjerojatnostima. Svaki element bi(ot) ove matrice iskazuje vjerojatnost generiranja osmotrenog simbola ot u i-tom stanju modela. Važno je napomenuti da su broj stanja modela N, duljina osmotrenog niza T i veličina rječnika izlaznih simbola V općenito različiti brojevi, što će se vidjeti iz primjera.

Skriveni Markovljev model prvog reda podrazumijeva dvije pojednostavljujuće pretpostavke. Prvo, kao i kod Markova lanca prvog reda, vjerojatnost određenog stanja ovisi isključivo o prethodnom stanju:

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

Drugo, vjerojatnost izlaznog promatranja oi ovisi samo o stanju qi u kojem je proizvedeno to promatranje, a ne o bilo kojem drugom stanju ili bilo kojem drugom promatranju:

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

Kako bi ilustrirali primjere ovih modela poslužit ćemo se zadatkom kojeg je predložio 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 koji navodi koliko je sladoleda Jason svakog dana jeo tog ljeta. Cilj nam je koristiti ta zabilježena opažanja za procjenu temperature svaki dan. Pojednostavit ćemo maksimalno ovaj vremenski zadatak pretpostavljajući da postoje samo dvije vrste dana: hladni (C(old)) i vrući (H(ot)). Dakle, Eisnerov zadatak je sljedeći:
S obzirom na slijed zabilježenih promatranja O (svaki cijeli broj u nizu predstavlja broj pojedenih sladoleda određenog 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.
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.

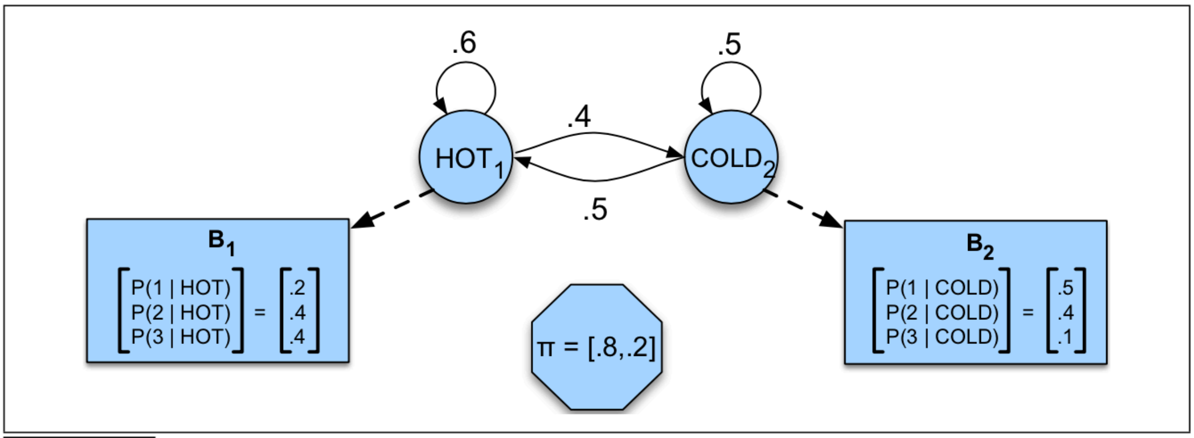

Figure A.2 A hidden Markov model for relating numbers of ice creams eaten by Jason (the observations) to the weather (H or C, the hidden variables).

U vrlo utjecajnom preglednom članku Rabiner-a 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 je ideju da skriveni Markovljevi modeli trebaju biti karakterizirani s tri temeljna problema:

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

Problem 1 (Likelihood):	Given an HMM $\lambda = (A, B)$ and an observation sequence O, determine the likelihood $P(O\|\lambda)$.
Problem 2 (Decoding):	Given an observation sequence O and an HMM $\lambda = (A, B)$, discover the best hidden state sequence Q.
Problem 3 (Learning):	Given an observation sequence O and the set of states in the HMM, learn the HMM parameters A and B.

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

U sljedeća dva poglavlja uvodimo algoritme „Unaprijed" i „Unaprijed-unazad" za rješavanje problema 1 i 3, a dajemo više informacija o načinu rješavanja problema 2 korištenjem Viterbijevog algoritma.

A.3. Izračun izvjesnosti – algoritam „Unaprijed“

Naš prvi problem je izračunati izvjesnost određenog slijeda promatranja za zadani model. Na primjer, s obzirom na HMM na slici A.2 koji statistički opisuje jedenje sladoleda, odredimo kolika je vjerojatnost osmatranja slijeda „3 1 3“? Formalnije:

Izračun izvjesnosti: S obzirom na zadani HMM λ = (A, B) i niz promatranja O, odredite izvjesnost P (O | λ).

Za Markovljev lanac, gdje su vanjska promatranja 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 skriveni slijed stanja!

Krenimo od malo jednostavnije situacije. Pretpostavimo da smo nekako ipak saznali kakvo je bilo vrijeme u ta tri dana i željeli bismo samo predvidjeti koliko će Jason pojesti sladoleda. Ovo je koristan detalj mnogih zadataka HMM-a. Za pretpostavljeni slijed skrivenog stanja (npr. vruće vruće hladno) možemo lako izračunati izlaznu izvjesnost niza „3 1 3“.
Da vidimo kako ćemo to učiniti. Prvo, podsjetimo se da za skrivene Markovljeve modele svako skriveno stanje daje samo jedno promatranje. Dakle, slijed skrivenih stanja i slijed promatranja imaju identičnu duljinu. S obzirom na takvo preslikavanje jedan-na-jedan i na Markovljeve pretpostavke izražene jednadžbom A.4, za određeni slijed skrivenog stanja Q = q0, q1, q2, ..., qT i slijed izlaznih promatranja O = o1, o2, ..., oT, izvjesnost osmotrenog slijeda se nalazi kao:

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

Izračun unaprijedne vjerojatnosti za naše promatranje 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.

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

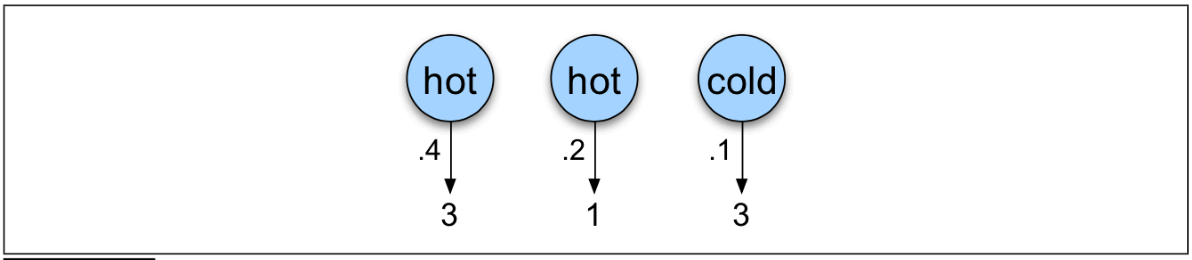

Figure A.3 The computation of the observation likelihood for the ice-cream events *3 1 3* given the hidden state sequence *hot hot cold*.

Ali naravno, zapravo mi ne znamo kakav je bio skriveni slijed stanja (promjene vremena). Stoga morat ćemo izračunati vjerojatnost sladolednih događaja „3 1 3“ zbrajanjem svih mogućih vremenskih sljedova, ponderiranih njihovom vjerojatnošću. Prvo, izračunajmo zajedničku vjerojatnost da se nalazimo u određenom vremenskom slijedu Q i da pri tome generiramo određeni izlazni slijed O događaja sladoleda. Općenito, to možemo zapisati ovim izrazom:

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

Izračun zajedničke vjerojatnosti našeg promatranja sladoleda „3 1 3“ i jednog mogućeg slijeda skrivenih stanja „vruće vruće hladno“ prikazan je u jednadžbi A.9. Slika A.4 prikazuje grafički izračun te zajedničke vjerojatnosti.

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

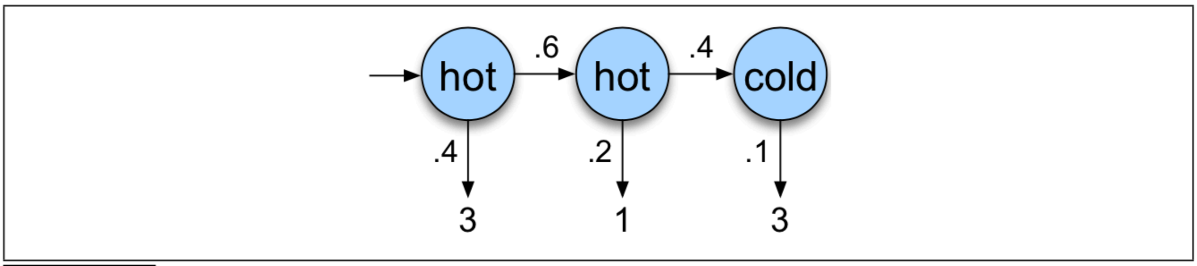

Figure A.4 The computation of the joint probability of the ice-cream events *3 1 3* and the hidden state sequence *hot hot cold*.

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“, odnosno prema izrazu:

$$P(3\ 1\ 3) = P(3\ 1\ 3, \text{cold cold cold}) + P(3\ 1\ 3, \text{cold cold hot}) + P(3\ 1\ 3, \text{hot hot cold}) + \ldots$$

Za HMM s N skrivenih stanja i sa slijedom opažanja 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 promatranja računanjem zasebne vjerojatnosti promatranja za svaku skrivenu sekvencu stanja, a zatim ih zbrojiti.

Umjesto korištenja takvog algoritma čija je složenost ekstremno eksponencijalna, koristimo učinkoviti $O(N^2T)$ algoritam koji se naziva **algoritam „Unaprijed“** (engl. forward algorithm). Ovaj algoritam je vrsta **dinamičkog programiranja**, odnosno algoritam koji koristi tablicu za pohranu međuvrijednosti 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“**.

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 o1=3 i o2=1, a analogno se proširuje i za zadnji osmotreni simbol o3=3.

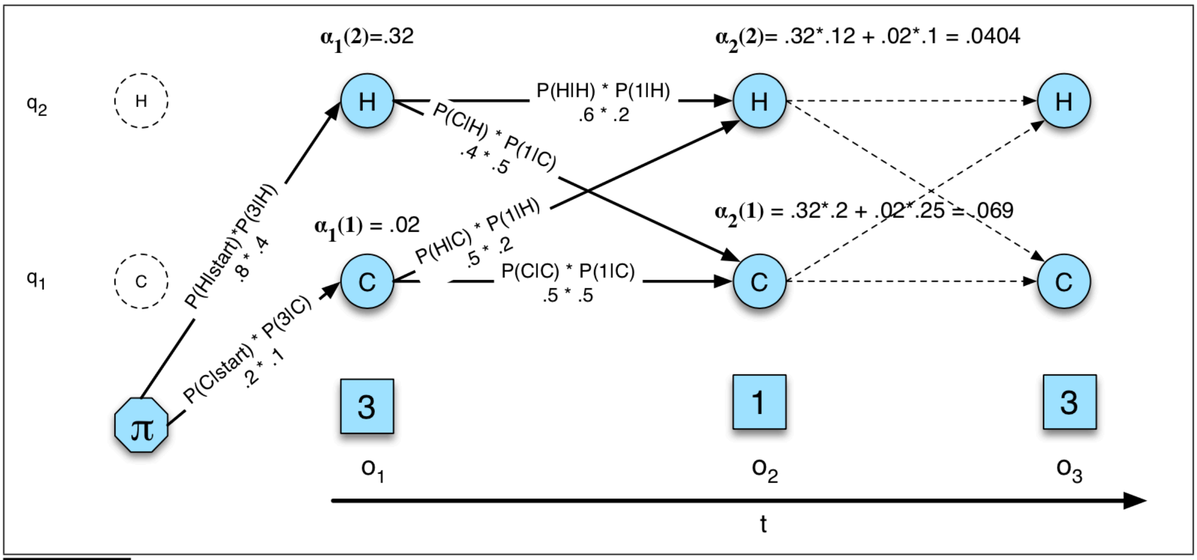

Figure A.5 The forward trellis for computing the total observation likelihood for the ice-cream events *3 1 3*. Hidden states are in circles, observations in squares. The figure shows the computation of $\alpha_t(j)$ for two states at two time steps. The computation in each cell follows Eq. A.12: $\alpha_t(j)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

Svaki stupac u ovoj rešetci odgovara pojedinom osmotrenom simbolu, dok svaki redak označava jedno od mogućih stanja u kojem je HMM 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. 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. Međutim u prvom vremenskom trenutku t=1 osmotren je simbol o1=3, pa će vjerojatnost da se mreža zatekla u stanju H biti π2* b2(o1=3) = 0.8*0.4=0.32, a vjerojatnost da se po isteku tok događaja zatekla u stanju C biti jednaka π1* b1(o1=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 stanje H je indeksirano kao drugo stanje, a stanje C kao prvo stanje, pa je taj indeks stanja označen kao argument unutar zagrada.

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 o2=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 $\alpha_1(2)$ * a22 * b2(o2=1) = 0.32 * 0.6 * 0.2, ili preko dijagonalnog puta iz stanja C s vjerojatnosti $\alpha_1(1)$ * a12 * b2(o2=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 o3=3, kako bi dobili izvjesnost osmatranja cijelog niza za zadani model.

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, a s obzirom na zadane parametre automata λ. 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:

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

Ovdje q_t=j označava „da je t-to stanje u nizu stanja upravo stanje j". Izračunavamo ovu vjerojatnost $\alpha_t(j)$ 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 sume koje koristimo za izračun unaprijedne vjerojatnosti u trenutku t temeljem produljenja svih prethodnih puteva su:

$\alpha_{t-1}(i)$	the **previous forward path probability** from the previous time step
a_{ij}	the **transition probability** from previous state q_i to current state q_j
$b_j(o_t)$	the **state observation likelihood** of the observation symbol o_t given the current state j

Dakle, to su:

- α_{t-1}(i) **prethodna vjerojatnost puta unaprijed** iz prethodnog vremenskog koraka
- aij **vjerojatnost prijelaza** iz prethodnog stanja qi u trenutno stanje qj
- bj (ot) **izvjesnost osmatranja** izlaznog simbola ot u stanju j

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". Izračunavamo ovu vrijednost proširujući α-vjerojatnosti iz vremenskog koraka 1, kroz dva moguća puta , gdje se svako proširenje sastoji od tri navedena faktora: $\alpha_1(1)$ × P (H | C) × P (1 | H) i $\alpha_1(2)$ × P (H | H) × P (1 | H).

Slika A.6 prikazuje općenitu vizualizaciju ovog indukcijskog koraka za izračunavanje vrijednosti u jednom novom čvoru rešetke.

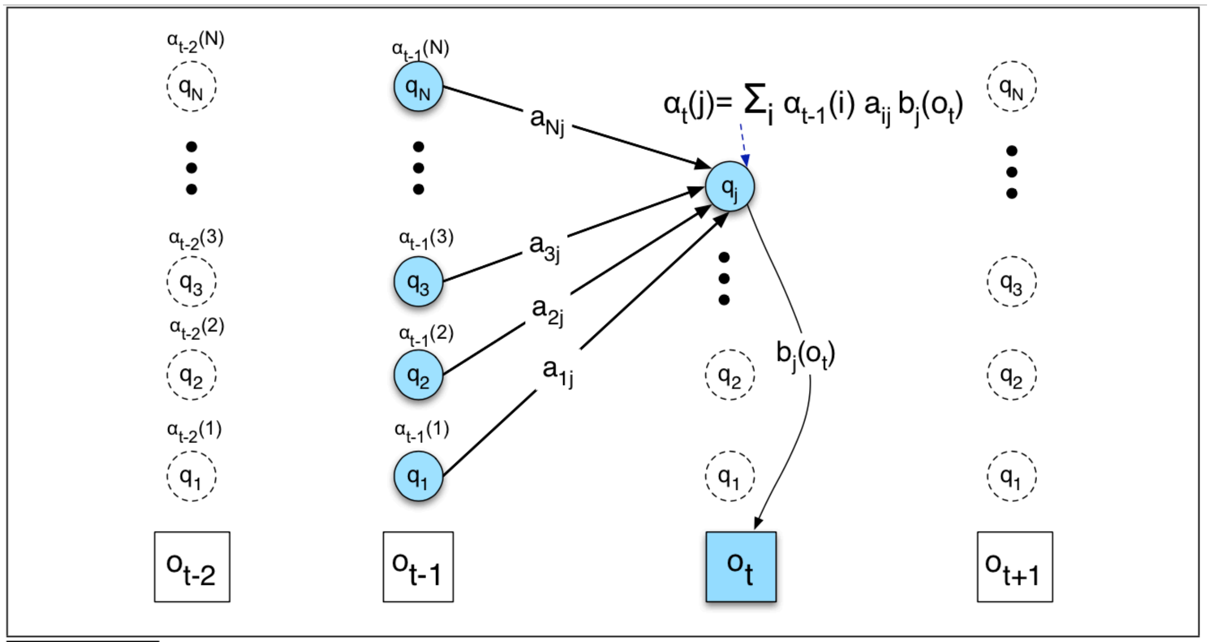

Figure A.6 Visualizing the computation of a single element $\alpha_t(i)$ in the trellis by summing all the previous values α_{t-1}, weighted by their transition probabilities a, and multiplying by the observation probability $b_i(o_t)$. For many applications of HMMs, many of the transition probabilities 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 $\alpha_t(i)$.

Konačno, dajemo i formalnu definiciju algoritma „Unaprijed“ opisanog pseudokodom na slici A.7.

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

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

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

return *forwardprob*

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

Ovu rekurzivni postupak možemo opisati i sljedećim formalnim matematičkim izrazima koji opisuju inicijalizaciju, iteraciju rekurzivnog postupka i konačno završetak kojim nalazimo traženu izvjesnost osmotrenog niza O uz zadani model:

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:

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

A.4 Dekodiranje: Viterbijev algoritam

Za bilo koji model, poput HMM-a, koji sadrži skrivene varijable, zadatak određivanja koji je slijed varijabli 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, zadaća **dekodera** je pronaći najbolji skriveni vremenski slijed (H H H). Formalnije ovaj postupak možemo definirati,

> **Dekodiranje**: Temeljem zadanog HMM modela λ = (A, B) za zadani slijed osmotrenih simbola O = o1, o2, ..., oT, pronađite najvjerojatniji slijed skrivenih stanja Q = q1, q2, q3, ... qT.

Postupak pronalaženja najboljeg slijeda bi mogli opisati ovako: za svaki mogući slijed skrivenih stanja (HHH, HHC, HCH, itd.) mogli bismo izvršiti algoritam Unaprijed i izračunati izvjesnost slijeda promatranja s obzirom na pretpostavljeni slijed stanja. Tako bismo mogli odabrati skriveni slijed stanja s najvećom izvjesnosti promatranja. Iz diskusije prethodnog poglavlja trebalo bi biti jasno da to ne možemo učiniti jer postoji eksponencijalno velik broj mogućih sljedova stanja.
Umjesto toga, najčešće korišteni algoritam dekodiranja stanja HMM-a je **Viterbijev algoritam**, kojeg ćemo kraće nazivati **Viterbi**. Poput algoritma Unaprijed, Viterbi je vrsta **dinamičkog programiranja** koje koristi rešetke za dinamičko programiranje. Viterbi također jako podsjeća na drugu varijantu dinamičkog programiranja, na algoritam najkraćeg puta (engl. minimum edit distance).

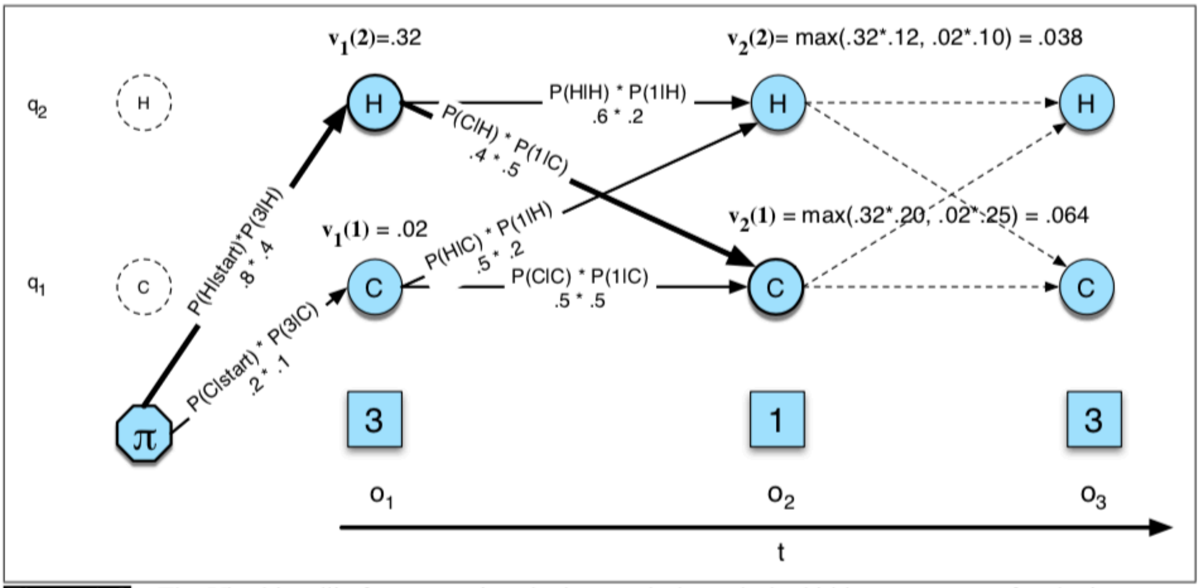

Figure A.8 The Viterbi trellis for computing the best path through the hidden state space for the ice-cream eating events *3 1 3*. Hidden states are in circles, observations in squares. White (unfilled) circles indicate illegal transitions. The figure shows the computation of $v_t(j)$ for two states 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: $v_t(j) = P(q_0, q_1, \ldots, q_{t-1}, o_1, o_2, \ldots, o_t, q_t = j|\lambda)$.

Slika A.8 ilustrira primjenu Viterbijeve rešetke za izračunavanje najboljeg niza skrivenih stanja za primjer sladoleda za odabrani slijed osmatranja „3 1 3". Ideja je obraditi slijed osmatranja slijeva udesno, popunjavajući rešetku. Svaki čvor 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 q1, ..., qt − 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 čvora. Formalno, svaki čvor izražava ovu vjerojatnost:

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

Prepoznajte iz gornjeg izraza da najvjerojatniji put određujemo nalazeći maksimum vjerojatnosti preko svih mogućih prethodnih sljedova stanja do trenutka t-1. Kao i preostali algoritmi dinamičkog programiranja, Viterbi rekurzivno izračunava svaki čvor rešetke. 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. Za pretpostavljeno stanje qj u trenutku t, vrijednost $v_t(j)$ izračunava se kao:

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

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:

$v_{t-1}(i)$	the **previous Viterbi path probability** from the previous time step
a_{ij}	the **transition probability** from previous state q_i to current state q_j
$b_j(o_t)$	the **state observation likelihood** of the observation symbol o_t given the current state j

Dakle, to su:

- $v_{t-1}(i)$ **prethodna Viterbijeva vjerojatnost** iz prethodnog vremenskog koraka
- aij **vjerojatnost prijelaza** iz prethodnog stanja qi u trenutno stanje qj
- bj (ot) **izvjesnost osmatranja** izlaznog simbola ot u trenutačnom stanju j

```
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
      viterbi[s,1] ← π_s * b_s(o_1)
      backpointer[s,1] ← 0
for each time step t from 2 to T do                   ; recursion step
   for each state s from 1 to N do
      viterbi[s,t] ← max_{s'=1}^{N} viterbi[s',t−1] * a_{s',s} * b_s(o_t)
      backpointer[s,t] ← argmax_{s'=1}^{N} viterbi[s',t−1] * a_{s',s} * b_s(o_t)
bestpathprob ← max_{s=1}^{N} viterbi[s,T]             ; termination step
bestpathpointer ← 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
```

Figure A.9 Viterbi algorithm for finding optimal sequence of hidden states. Given an observation sequence and an HMM $\lambda = (A, B)$, the algorithm returns the state path through the HMM that assigns maximum likelihood to the observation sequence.

Slika A.9 prikazuje pseudokod 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. Također imajte na umu da Viterbijev algoritam ima jednu dodatnu komponentu koju algoritam Unaprijed nema: ima tzv. povratne pokazivače. Razlog je taj što dok algoritam Unaprijed treba izračunati samo ukupnu izvjesnost promatranja zadanog modela, Viterbijev algoritam mora izračunati vjerojatnost uzduž najboljeg puta, ali istovremeno i najvjerojatniji slijed stanja. Ovu 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 ot, i time završili u najvjerojatnijem završnom stanju qt, vraćamo se unazad do prvog stanja, ali prateći samo najbolji put do početka (tzv. Viterbijev trag rešetke). Taj najbolji put je prikazan podebljanim strelicama u rešetci na slici A.10.

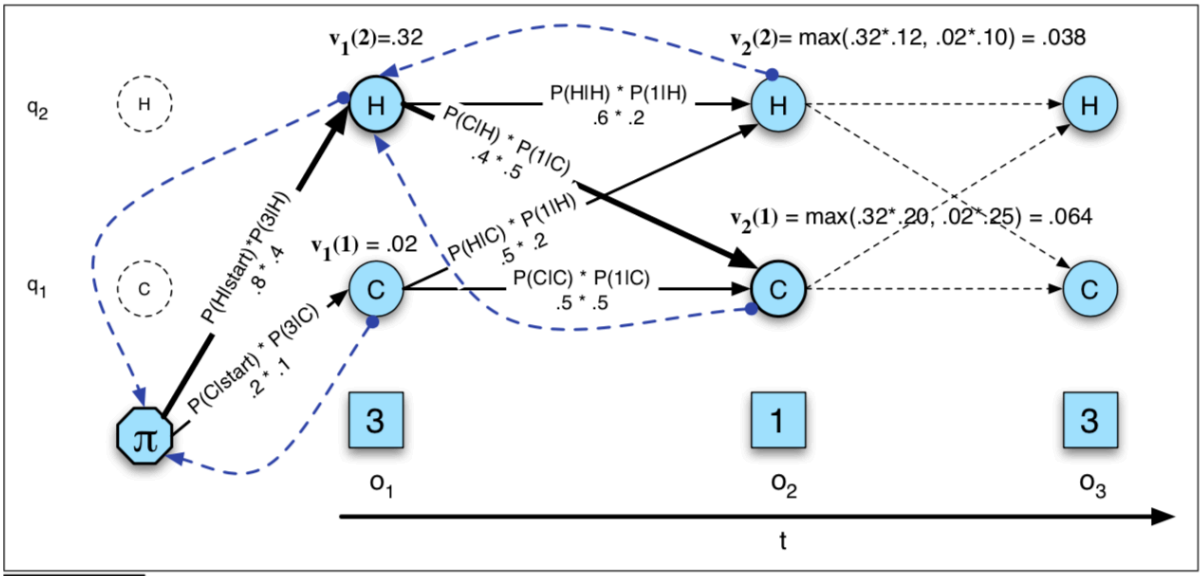

Figure A.10 The Viterbi backtrace. As we extend each path to a new state account for the next observation, we keep a backpointer (shown with broken lines) to the best path that led us to this state.

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. Međutim u prvom vremenskom trenutku t=1 osmotren je simbol o1=3, pa će vjerojatnost da se mreža zatekla u stanju H biti π2* b2(o1=3) = 0.8*0.4=0.32, a vjerojatnost da se po isteku tok događaja zatekla u stanju C biti jednaka π1* b1(o1=3) = 0.2*0.1=0.02. Te su vjerojatnosti na slici označene kao $v_1(2)$=0.32, odnosno $v_1(1)$=0.02. Ovaj dio opisa identičan je algoritmu „Unaprijed“.
Sada ove dvije vrijednosti koristimo za izračun $v_2(2)$ i $v_2(1)$ u drugom vremenskom koraku temeljem matrice prijelaza A (koja opisuje vjerojatnosti svih mogućih unaprijednih puteva u mreži), te poznavanjem vjerojatnosti osmatranja drugog izlaznog simbola o2=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 $v_1(2)$ * a22 * b2(o2=1) = 0.32 * 0.6 * 0.2, ili preko dijagonalnog puta iz stanja C s vjerojatnosti $v_1(1)$ * a12 * b2(o2=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, jer biramo samo horizontalnu granu. 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 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, odabrali bismo samo put najveće vjerojatnosti koji završava u stanju C, kojem je prema povratnom pokazivaču prethodilo stanje H u koraku t=1. Isti postupak sada ponavljamo i za treći osmotreni simbol o3=3, kako bi dobili Viterbijevu vjerojatnost i najvjerojatniji slijed stanja cijelog osmotrenog niza O za zadani model.
Važno je napomenuti da ovo 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, što prozivaju prva dva osmotrena simbola . Stoga je nužno ovaj rekurzivni algoritam izvršiti sve do zadnjeg osmotrenog simbola i tek tada se iz završnog stanja najviše Viterbi vjerojatnosti vratiti do prvog trenutka korištenjem povratnih pokazivača koje smo izgradili u rekurzivnom postupku.

Ovaj Viterbijev rekurzivni postupak možemo opisati i formalnim matematičkim izrazima koji opisuju inicijalizaciju, iteraciju rekurzivnog postupka i konačno završetak kojim nalazimo traženu Viterbijevu vjerojatnost osmotrenog niza O, ali i najvjerojatniji niz skrivenih stanja O uz zadani model:

1. **Initialization:**

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

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:**

$$\text{The best score:} \quad P* = \max_{i=1}^{N} v_T(i)$$

$$\text{The start of backtrace:} \quad q_T* = \operatorname*{argmax}_{i=1}^{N} v_T(i)$$

U koraku rekurzije računamo Viterbijeve vjerojatnosti svakog čvora rešetke $v_t(j)$, ali i pripadajuće povratne pokazivače $bt_t(j)$ koji indeksiraju najvjerojatnije stanje u koraku t-1 koje prethodi stanju j u koraku t za zadani model i zadane izlazne opservacije.

A.5 Treniranje HMM-a: Algoritam „Unaprijed-unazad“

Konačno posvećujemo pažnju i trećem problemu vezanom uz HMM modele: učenju parametara HMM-a, odnosno A i B matrica, što se još naziva treniranjem modela. Formalno,

> **Treniranje**: S obzirom na slijed promatranja O i skup mogućih stanja HMM-a, naučite HMM parametre A i B korištenjem odabranog kriterija.

Ulaz u takav algoritam učenja bio bi neoznačeni slijed promatranja O i isključivo rječnik potencijalnih skrivenih stanja Q, ali ne i niz stanja. Dakle, za primjer zadatka sladoleda započeli bismo slijedom promatranja O = {1, 3, 2 , ...,} uz skup skrivenih stanja H i C.
Standardni algoritam za treniranje HMM modela je algoritam „Unaprijed-unazad“ koji je poznat pod nazivom Baum-Welch algoritam (Baum, 1972) prema prezimenima autora. On predstavlja poseban slučaj EM algoritma maksimizacije očekivanja (engl. expectation maximization, EM) (Dempster i sur., 1977). Ovaj algoritam će nam omogućiti treniranje matrice prijelaznih vjerojatnosti A i matrice vjerojatnosti osmatranja izlaznih simbola B HMM-a. Matrica B se još naziva i matricom emisijskih vjerojatnosti modela, jer to odgovara činjenici da HMM model emitira niz izlaznih simbola, koji se u idealnom slučaju poklapaju sa zadanim opservacijama O. 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.
Započnimo s razmatranjem puno jednostavnijeg slučaja učenja potpuno osmotrivog Markovljevog modela, tj. pretpostavimo da na neki način poznamo i temperaturu i broj pojedenih sladoleda za svaki dan. Odnosno, zamislimo da vidimo sljedeći skup ulaznih promatranja i magično poznajemo poravnane nizove skrivenih stanja:

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

To bi omogućilo jednostavno izračunavanje parametara HMM estimacijom najveće izvjesnosti iz ulaznih podataka za trening. Prvo, možemo izračunati vektor početne vjerojatnosti π prebrojavanjem početnih otkrivenih stanja za tri navedena slučaja. 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 ovom slučajnom ulaznom uzorku:

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

Dalje možemo izravno izračunati A matricu iz svih mogućih prijelaza stanja za ova tri slijeda osmatranja izlaznih simbola modela tako da jednostavno prebrojavamo sve promjene stanja iz prvog u drugi korak i iz drugog u treći korak. Model je dva puta prešao iz stanja hot u hot i jednom iz stanja hot u stanje cold, pa formiramo prve dvije vjerojatnosti prijelaza:

$$p(hot|hot) = 2/3 \qquad p(cold|hot) = 1/3$$

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:

$$p(cold|cold) = 2/3 \qquad p(hot|cold) = 1/3$$

Na sličan način, prebrojavanjem nalazimo i emisijske vjerojatnosti B modela:

P(1|hot) = 0/4 = 0	p(1|cold) = 3/5 = .6
P(2|hot) = 1/4 = .25	p(2|cold = 2/5 = .4
P(3|hot) = 3/4 = .75	p(3|cold) = 0

Za stvarni HMM ne možemo izračunati te vjerojatnosti prebrojavanjem izravno iz ulaznih nizova osmatranja jer ne znamo koji je put stanja prošao model za zadani ulaz. Na primjer, pretpostavimo da nismo poznavali temperaturu 2. dana i da ste je morali na neki način pogoditi, ali ste (magično) ipak imali gornje vjerojatnosti i temperature svih ostalih dana. Možete primijeniti Bayesovu aritmetiku temeljenu na poznatim vjerojatnostima kako biste dobili procjene vjerojatne temperature tog dana koji nedostaje, a pomoću njih dobiti i očekivane brojeve pojava visokih i niskih temperatura za 2. dan.

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 procjenom vjerojatnosti prijelaza i opservacija, a zatim ćemo koristiti te procijenjene vjerojatnosti za izvođenje sve boljih vjerojatnosti. A to ćemo učiniti izračunavanjem vjerojatnosti unaprijed za promatranje O i zatim raspodjelom te ukupne mase vjerojatnosti između svih mogućih putova koji su pridonijeli toj vjerojatnosti unaprijed.

Da bismo razumjeli algoritam, moramo uvesti još jednu korisnu vjerojatnost koja je povezana s vjerojatnosti unaprijed, nazvanu vjerojatnost unazad (engl. backward probability). Potpuno analogno s definicijom vjerojatnosti unaprijed, vjerojatnost unazad β će biti 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, 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 do željenog trenutka t (koji može biti i t=1, ako želimo odrediti ukupnu izvjesnost cijelog osmatranja O za zadani model) :

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:**

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

Slika A.11 prikazuje jedan korak ovog rekurzivnog izračuna vjerojatnosti unazad.

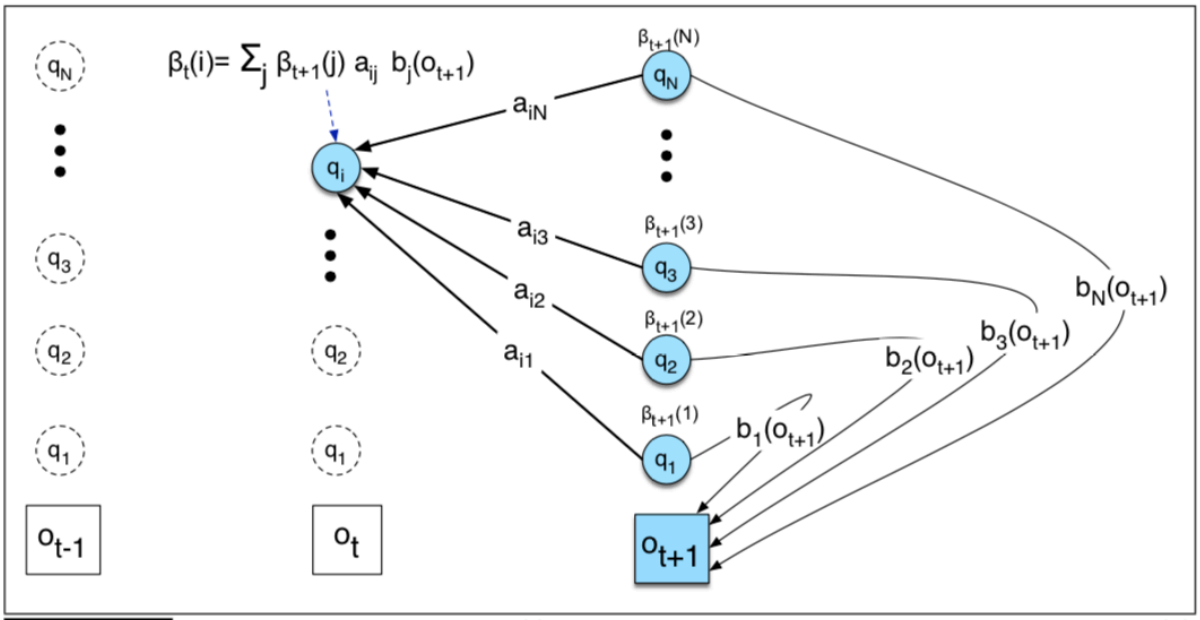

Figure A.11 The computation of $\beta_t(i)$ by summing all the successive values $\beta_{t+1}(j)$ weighted by their transition probabilities a_{ij} and their observation probabilities $b_j(o_{t+1})$. Start and end states not shown.

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 model 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. Kako izračunavamo brojnik? Evo intuitivnog objašnjenja. 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 ξt kao vjerojatnost da budemo u stanju i u trenutku t, a stanju j u trenutku t + 1, s obzirom na slijed osmotrenih simbola i naravno s obzirom na model:

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

Da bismo izračunali ξt, prvo izračunavamo jednu pomoćnu vjerojatnost koja je slična ξt, ali se razlikuje po tome što uključuje vjerojatnost opažanja. Uočite različito iskazanu uvjetnu vjerojatnost u izrazima A.17 i A.18:

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

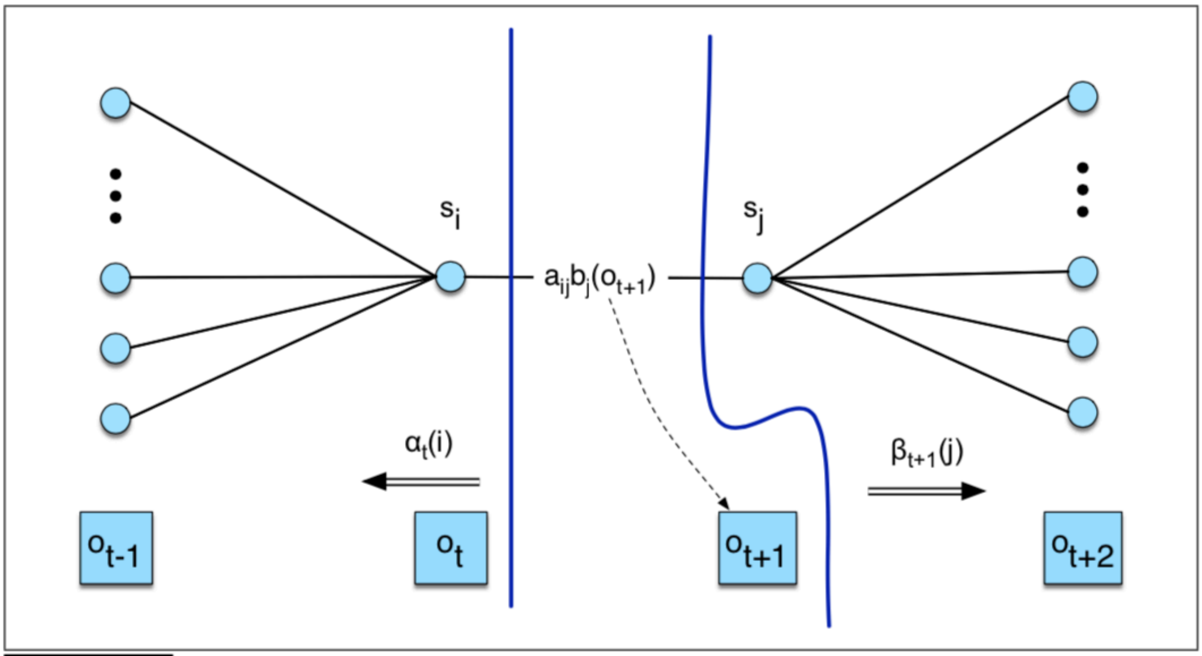

Figure A.12 Computation of the joint probability of being in state i at time t and state j at time $t+1$. The figure shows the various probabilities that need to be combined to produce $P(q_t = i, q_{t+1} = j, O|\lambda)$: the α and β probabilities, the transition probability a_{ij} and the observation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

Slika A.12 prikazuje razne vjerojatnosti koje sudjeluju u izračunavanju ove pomoćne veličine koju ćemo nazvati „ne baš-ξt“, odnosno u izvornom obliku na engleskom jeziku, kao „not-quite-ξt“. To su vjerojatnost prijelaza za dotični luk, α vjerojatnost prije luka (izračunata algoritmom Unaprijed), β vjerojatnost nakon luka (izračunata algoritmom Unazad) i konačno 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:

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

Kako bi u konačnici odredili ξt iz ove pomoćne veličine koristimo dobro poznato pravilo iz teorije vjerojatnosti i jednostavno je dijelimo s ukupnom vjerojatnosti osmotrenog niza O za zadani model λ, P(O|λ), jer ovo primijenjeno pravilo glasi:

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

Vjerojatnost osmatranja s obzirom na model P(O|λ), jednostavno je vjerojatnost cijelog osmotrenog niza simbola izračunata algoritmom Unaprijed (ili alternativno, vjerojatnost istog niza izračunata algoritmom Unazad):

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

Stoga, konačni izraz za ξt dobivamo u obliku:

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

Očekivani broj prijelaza iz stanja i u stanje j tada je jednostavno zbroj ξt za svaki t. Za našu procjenu vjerojatnosti prijelaza aij u jednadžbi A.16, trebamo samo još jednu dodatnu stvar: to je ukupan očekivani broj prijelaza iz stanja i u nazivniku izraza. Njega možemo dobiti zbrajanjem svih prijelaza iz stanja i u bilo koje stanje modela. Evo konačne formule za procjenu a_{ij}:

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

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

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

dakle kao omjer očekivanog broja osmatranja simbola vk 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 γt(j):

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

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

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

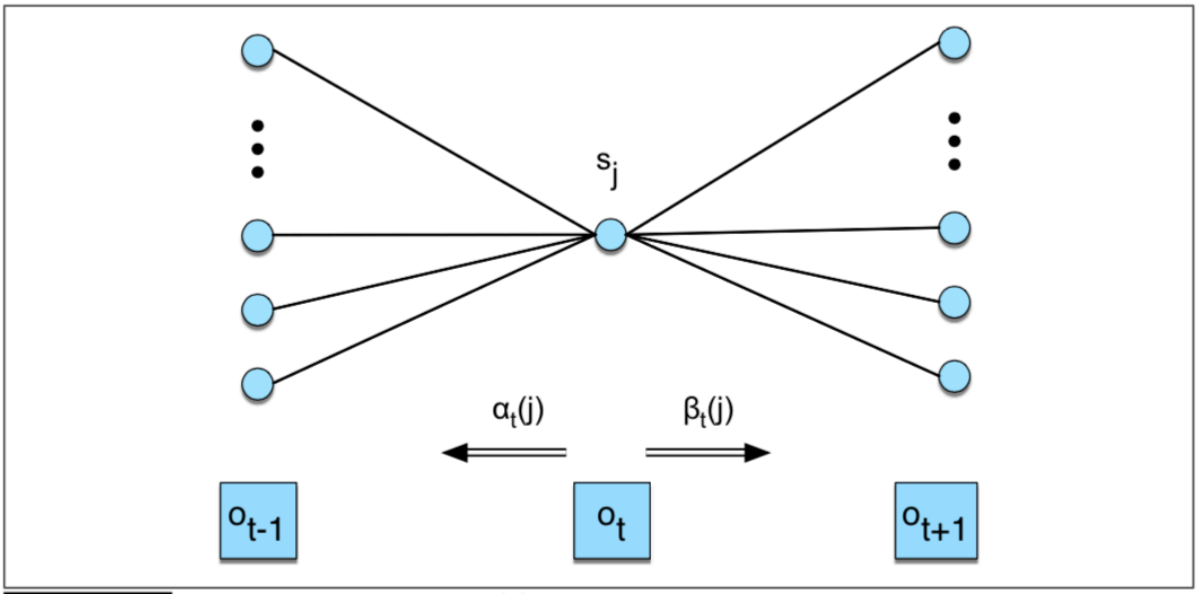

Figure A.13 The computation of $\gamma_t(j)$, the probability of being in state j at time t. Note that γ is really a degenerate case of ξ and hence this figure is like a version of Fig. A.12 with state i collapsed with state j. After Rabiner (1989) which is ©1989 IEEE.

Izračun ove pomoćne veličine γt(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 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 B. Za brojnik zbrajamo γ_t (j) za sve vremenske korake t u kojima je osmotreni simbol ot jednak simbolu vk koji nas zanima. Za nazivnik zbrajamo γ_t (j) u svim vremenskim koracima t, neovisno od osmotrenog simbola. Rezultat je udio koliko puta kada smo bili u stanju j smo osmotrili simbol vk (oznaka iza sume u brojniku se čita "Zbroj preko svih t za koje je osmotreni simbol u trenutku t bio simbol vk"):

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

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.

Jednadžba A.23 (za vjerojatnosti prijelaza) i jednadžba A.28 (za emisijske vjerojatnosti) omogućavaju ponovnu procjenu vjerojatnosti prijelaza A i promatranja B iz zadanog niza osmotrenih simbola O, ali pod pretpostavkom da već imamo prethodnu procjenu A i B, jer smo za izračun ovih novih estimacija morali koristiti postojeći model λ = (A, B). Te ponovne procjene čine srž iterativnog algoritma „Unaprijed-unazad". Algoritam „Unaprijed-unazad" (prikazan slikom A.14) započinje nekom početnom procjenom HMM parametara λ = (A, B). Zatim iterativno ponavljamo dva koraka. Kao i kod drugih slučajeva primjene EM algoritma (algoritma maksimizacija očekivanja), opisani algoritam „Unaprijed-unazad" koji

također pripada porodici EM algoritama isto ima ta dva ključna koraka: **korak očekivanja** ili tzv. **E-korak** i **korak maksimizacije** ili tzv. **M-korak**.
U E-koraku izračunavamo očekivani broj zauzetosti stanja γ i očekivani broj prijelaza stanja ξ iz „fiksiranih" vjerojatnosti A i B određenih u prošlom koraku. U M koraku koristimo γ i ξ za ponovno izračunavanje novih vjerojatnosti A i B, koje „fiksiramo" i koristimo u narednom koraku.

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 vjerojatnosti emisije (B) i prijelaza A (koje ne smiju biti jednake nula) treniraju iz niza sekvenci promatranja O.

A.6 Sažetak

U ovom dokumentu smo predstavili skrivene Markovljeve modele za vjerojatnosnu klasifikaciju opservacija proizvoljnih duljina iz diskretnog rječnika.

- Skriveni Markovljevi modeli (HMM) način su 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. Viterbijev algoritam se obično koristi za dekodiranje.
- 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

Markovljeve lance prvi je upotrijebio Andrey Andreyevich Markov (1913), Андрéй Андрéевич Мáрков, (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". Kruskal i drugi daju barem sljedeće neovisno otkrivene inačice 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

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 obični izraz - dinamičko programiranje. Forney, mlađi (1973.) napisao je rani istraživački rad koji istražuje podrijetlo Viterbijevog algoritma u kontekstu teorije informacija i komunikacija.

Prezentacija ideje u ovom dokumentu 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 ('Example of a statistical investigation of the text of "Eugene Onegin" illustrating the dependence between samples in 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. Akade´miai Kiado´.

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.