UNIVERZITET U NIŠU
PRIRODNO-MATEMATIČKI FAKULTET

Departman za računarske nauke

Velimir M. Ilić

Izračunavanje kros-momenata nad probabilističkim kontekstno-nezavisnim gramatikama i probabilističkim grafičkim modelima

Doktorska disertacija

Niš, 2012.

Zahvalnica

Osećam potrebu da na ovom mestu izrazim zahvalnost onima koji su na posredan ili neposredan način doprineli razvoju i konačnoj formi ove disertacije.

Na prvom mestu želeo bih da se zahvalim supruzi Jeleni koja me je na ovom putu svesrdno pratila i bezuslovno me podržavala.

Takodje, želeo bih da se zahvalim ćerki Dunji koja me je u ovom poduhvatu nekada uspešno, a nekada neuspešno, istrajno ometala, ali me i hranila snagom da nastavim dalje.

Roditeljima dugujem zahvalnost na podršci koju si mi pružali kroz život. Zahvaljujem za veru koju su imali u mene prilikom osnovnih, i još veću za onu koju su imali za vreme mojih doktorskih studija.

Mentoru, Miroslavu Ćiriću, redovnom profesoru Prirodno-matematičkog fakulteta u Nišu, zahvaljujem na nesebičnoj podršci prilikom izrade disertacije, kao i na zasnivanju i unapredjivanju moje vizije pretakanja naučne ideje u naučni rad.

Članu komisije, Miomiru Stankoviću, redovnom profesoru Fakulteta zaštite na radu u Nišu, dugujem veliku zahvalnost za konstnatnu stručnu i moralnu podršku koju mi je pružao od samog početka mog istraživačkog rada, pa do završetka rada na ovoj disertaciji.

Članu komisije, Branimiru Todoroviću, vanrednom profesoru Prirodno-matematičkog fakulteta u Nišu, dugujem zahvalnost za uvodjenje u algoritme nad skrivenim Markovljevim modelima od kojih je moje istraživanje i počelo, da bi došlo do onog što ova disertacija jeste.

Takodje, zahvaljujem ostalim članovima komisije, Zoranu Ognjanoviću, naučnom savetniku Matematičkog instituta u Beogradu i Jeleni Ignjatović, vanrednom profesoru Prirodno-matematičkog fakulteta u Nišu, na korisnim savetima i sugestijama.

Dejanu Mančevu, asistentu Prirodno-matematičkog fakulteta u Nišu, dugujem zahvalnost, najpre za zajednički rad iz oblasti uslovnih slučajnih polja, a zatim i za konstantu razmenu ideja u vezi sa algoritmima obradjenim u ovoj disertaciji.

Konačno, veliku zahvalnost dugujem kolegama iz firme Accordia group na moralnoj i idejnoj podršci koja je bila prisutna kroz sve godine stvaranja ove disertacije.

U Nišu, septembra 2012. godine.

Predgovor

Kros-momenti vektorske slučajne promenljive predstavljaju bazične statističke veličine koje opisuju raspodelu promenljive. Definišu se kao očekivana vrednost proizvoda celobrojnih stepena koordinata vektorske slučajne promenljive. Izračunavanje kros-momenata može postati zahtevno ukoliko je broj mogućih realizacija slučajne promenljive veliki, a praktično neizvodljiv u slučaju kada je beskonačan. Medjutim, za specifičnu strukturu raspodele slučajne promenljive i strukturu slučajne promenljive ovaj problem može biti rešen na efikasan način, što predstavlja temu ove disertacije.

U disertaciji razmatramo tri tipa probabilističkih modela:

- Markovljevi lanci (skriveni Markovljevi modeli i uslovna slučajna polja),
- Probabilistički grafički modeli,
- Probabilističke kontekstno-nezavisne gramatike.

Glavni doprinos ove disertacije nalazi se u novim algoritmima za izračunavanje kros-momenata pomenutih probabilističkih modela. Algoritmi su bazirani na dinamičkom programiranju nad komutativnim poluprstenom. Pri tome se za kros-momente probabilističkih kontekstno-nezavisnih gramatika koriste algoritam za izračunavanje particione funkcije [66] i *inside-outside* algoritma [26], [53], a za probabilističke grafičke modele je korišćen algoritam slanja poruka [1], [15], [49], [68].

Disertacija je organizovana na sledeći način.

U glavi 1 opisana je notacija korišćena u disertaciji, date su teorijske osnove iz algebarskih struktura i teorija verovatnoće, a zatim je na jednostavnom primeru objašnjena ključna ideja pomoću koje je moguće izvršiti efikasno izračunavanje kros-momenata. U disertaciji je korišćena multi-indeks notacija. Umesto standardnog indeksiranja nenegativnim celim brojevima, simboli su indeksirani uredjenim n-torkama nenegativnih celih brojeva nazvanih multi-indeksi. Pokazano je kako se multinomna i generalisana Lajbnicova formula mogu zapisati u multi-indeks notaciji. Zatim, dat je pregled osnovnih algebarskih pojmova, kao što su algebarske strukture i elementarne definicije iz teorije grafova. Takodje, dati su osnovni pojmovi verovatnoće sa posebnim akcentom na statističke veličine kao što su: matematičko očekivanje, entropija i kros-momenti. Na kraju glave na jednostavnom primeru objašnjena je ključna ideja pomoću koje je moguće izvršiti efikasno izračunavanje kros-momenata u slučaju kada funkcije imaju jednostavnu strukturu. Pokazujemo kako se izračunavanje momenata može izvršiti primenom distributivnog zakona u poluprstenu, koji se sastoji od uredjenih n-torki indeksiranih multi-indeksima i operacijama definisanim na odgovarajući način.

U glavi 2 razmatramo izračunavanje očekivanja vektorske slučajne promenljive nad grafičkim modelima kao što su: Bajesovske mreže, Markovljeva slučajna polja i *junction*

stabla, pri čemu je poseban akcenat stavljen na faktor-grafove. Algoritmi iz ove glave mogu se predstaviti procesom slanja poruka kroz graf, pri čemu poruke nose informaciju o matematičkom očekivanju slučajne promenljive. Razvijen je novi algoritam koji funkcioniše kao algoritam slanja poruka nad poluprstenom očekivanja [22], koji se naziva *EMP* algoritam (*entropy message passing*) [38]. Algoritmi iz ove glave su uskoj vezi sa prethodno razvijenim algoritmima za izračunavanje očekivanja nad *junction* stablima [43], [67], [59], [54] i faktor grafovima u [21], [18], [16], [17]. Ove veze su objašnjene i izvršena poredjenja kompleksnosti sa pomenutim algoritmima. Glava 2 bazirana je na radu:

- V. M. Ilić, M. S. Stanković, B. T. Todorović, *Entropy message passing*, IEEE Transactions on Information Theory, 57 (2011), pp. 219–242.

U glavi 3 razmatramo izračunavanje entropije skrivenog Markovljevog modela uz pomoć *forward-backward* algoritma [3], [4], [5], [9], [83], nad entropijskim poluprstenom [13]. Razvijeni algoritam predstavlja numerički stabilnu verziju *EMP* algoritma primenjenog na skrivene Markovljeve modele, i ima znatno manju memorijsku kompleksnost nego algoritam koji su dali *Mann* i *McCallum* [60]. Takodje, pokazano je kako razvijeni algoritam može biti transformisan u algoritam koji su dali *Hernando* i saradnici [31]. Glava 3 bazirana je u na radu:

- V. M. Ilić, M. S. Stanković, B. T. Todorović, *Entropy semiiring forward-backward algorithm for HMM entropy computation*, Transactions on Advanced Research, 8 (2012), pp. 8–15.

U glavi 4 razmatramo izračunavanje gradijenta uslovnih slučajnih polja. Slično kao u glavi 3 izvodimo algoritam koji je baziran na *forward-backward* algoritmu [3], [4], [5], [9], [83], nad log-domen poluprstenom očekivanja. Razvijeni algoritam predstavlja numerički stabilnu verziju *EMP* algoritma primenjenog na uslovna slučajna polja. Razvijeni algoritam ima znatno manju memorijsku kompleksnost i nešto veću vremensku kompleksnost nego algoritam koji su dali *Lafferty* i saradnici [51]. Glava 4 bazirana je na radu:

- V. M. Ilić, D. I. Mančev, B. T. Todorović, M. S. Stanković, *Gradient computation in linear-chain conditional random fields using the entropy message passing algorithm*, Pattern Recognition Letters, 33 (2012), pp. 1776 – 1784.

U glavi 5 uopšteni su algoritmi iz glava 2, 3 i 4 na generalni slučaj kros-momenata proizvoljnog reda. Razvijena su dva generalna algoritma: 1) algoritam slanja poruka nad poluprstenom polinoma (*polynomial semiring message passing, PSMP*) i 2) algoritam slanja binomnih poruka (*binomial semiring message passing, BSMP*). U ovim algoritmima, poruke predstavljaju uredjene n-torke, i one prenose informaciju o svim kros-momentima, zaključno sa kros-momentom najvišeg reda koji je od interesa. *BSMP* i *PSMP* se poklapaju kada se koriste za izračunavanje kros-momenata reda (1) i $(1,1)$ i mogu se shvatiti kao generalizacija *EMP* algoritma u slučaju reda (1), i kao generalizacija algoritma koji su razvili *Kulesza* i *Taskar* [50] u slučaju reda $(1,1)$. Takodje, *PSMP* predstavlja generalizaciju algoritma za izračunavanje skalarnih momenata koji su razvili *Cowell* i saradnici [15], dok je *BSMP* generalizacija algoritma koji su razvili *Heim* i saradnici [30] za modele sa strukturom lanca. Glava 5 bazirana je u na radu:

- V. M. Ilić, M. S. Stanković, B. T. Todorović, *Computation of cross-moments using message passing over factor-graphs*, vol. 6, American Institute of MAthematical Sciences.

U glavi 6 razmatrani su kros momenti probabilističkih kontekstno-nezavisnih gramatika. Na polju probabilističkih kontekstno-nezavisnih gramatika problem izračunavanja kros-momenata je u velikoj meri već obradjivan, ali za specijalne slučajeve kros-momenata reda ne većeg od dva i uglavnom za skalarne promenljive. Razmatraju se dva slučaja: 1) skup elementarnih dogadjaja je skup svih izvodjenja gramatike (naziv kros-moment se obično odnosi na ovaj slučaj) i 2) skup elementarnih dogadjaja je skup svih izvodjenja za datu reč generisanu gramatikom (u ovom slučaju govorimo o uslovnim kros-momentima). Kros-momente skalarnih promenljivih prvog reda (ili, kraće, momente prvog reda), kao što su očekivana dužina izvodjenja i očekivana dužina izvedene reči razmatrao je *Wetherell* [87]. Izračunavanje entropije probabilističke kontekstno-nezavisne gramatike razmatrali su *Nederhof* i *Satta* [65]. Postupak za izračunavanje momenta dužine izvedene reči dala je Hutchins [33], koja je izvela formule za momente prvog i drugog reda. Algoritam za izračunavanje uslovne entropije dala je *Hwa* [34]. Algoritam za izračunavanje uslovnih kros-momenata vektorske promenljive reda dva dali su *Li* i *Eisner* [56]. Glava 6 je bazirana na radu:

- V. M. Ilić, M. D. Ćirić, M. S. Stanković, *Cross-moments computation for stochastic context-free grammars*, CoRR, abs/1108.0353 (2011).

Sadržaj

Glava 1

Osnovni pojmovi i ideje

U ovoj glavi opisana je notacija korišćena u disertaciji, date su teorijske osnove iz algebarskih struktura i teorija verovatnoće, a zatim je na jednostavnom primeru objašnjena ključna ideja pomoću koje je moguće izvršiti efikasno izračunavanje kros-momenata. U disertaciji je korišćena multi-indeks notacija, koja je opisana u poglavlju 1.1. Umesto standardnog indeksiranja nenegativnim celim brojevima, simboli su indeksirani uredjenim n-torkama nenegativnih celih brojeva nazvanih multi-indeksi. Pokazano je kako se multinomna i generalisana Lajbnicova formula mogu zapisati u multi-indeks notaciji. U poglavlju 1.2, dat je pregled osnovnih algebarskih pojmova kao što su algebarske strukture i elementarne definicije iz teorije grafova. U poglavlju 1.3 dati su osnovni pojmovi verovatnoće sa posebnim akcentom na statističke veličine kao što su: matematičko očekivanje, entropija i kros-momenti. U poglavlju 1.4 je na jednostavnom primeru objašnjena ključna ideja pomoću koje je moguće izvršiti efikasno izračunavanje kros-momenata za funkcije sa jednostavnom strukturom. Pokazujemo kako se izračunavanje momenata može izvršiti primenom distributivnog zakona u poluprstenu koji se sastoji od uredjenih n-torki indeksiranih multi-indeksima i operacijama definisanim na odgovarajući način.

1.1 Osnovna notacija

1.1.1 Skupovi i sekvence

Skup prirodnih brojeva označavamo sa $\mathbb{N}$, a prošireni skup prirodnih brojeva sa $\mathbb{N}_0$. Skup realnih brojeva označavamo sa $\mathbb{R}$, a prošireni skup realnih brojeva, $\mathbb{R}\cup\{-\infty,+\infty\}$ označavamo sa $\mathbb{R}^*$. Broj elemenata konačnog skupa V označavamo sa $|V|$. Ako se skup sastoji od jednog elementa $\{n\}$, nećemo pisati zagrade ukoliko to ne dovodi do zabune. Tako ćemo, na primer, razliku izmedju skupa M i skupa n označavati sa $M \setminus n$. Konačan podskup skupa prirodnih brojeva $V = \{i_1, \cdots, i_{|V|}\} \subseteq \mathbb{N}_0$ nazivamo *indeksni skup*, a njegove elemente *indeksi*. Skupove indeksa ćemo označavati velikim slovima abecede (M, K, itd.), a indekse malim (n, k, itd.).

Pod *sekvencom* dužine n podrazumevamo uredjenu n-torku. Neka je $(X_{i_1}, \ldots, X_{i_{|V|}})$ proizvoljna sekvenca simbola indeksiranih sa $i_1 \cdots i_{|V|}$. Ukoliko je $i_1 < i_2 < \cdots < i_{|V|}$ za sekvencu $(X_{i_1}, \ldots, X_{i_{|V|}})$ koristićemo zapis X_V. Skup indeksa $\{l, l+1, \ldots, r\}$ će skraćeno biti označen sa $l : r$, a skup $0 : l-1 \cup r+1 : T$ sa $-l : r$. Shodno tome, sekvenca simbola $(X_l, X_{l+1}, \ldots X_r)$ se obeležava sa $X_{l:r}$, a sekvenca $(X_0, X_1, \ldots, X_{l-1}, X_{r+1}, \ldots, X_T)$ sa $X_{-l:r}$.

Simbol $\sum_{x\in\mathbb{X}}$ označava sumiranje po skupu $\mathbb{X}$. Ukoliko nema opasnosti od konfuzije izostavljaćemo simbol $\mathbb{X}$ i pisati $\sum_x$.

1.1.2 Multi-indeks notacija

Multi-indeks se definiše kao uredjena n-torka nenegativnih celih brojeva $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$. Definišemo *dimenziju* multi-indeksa, $\mathrm{d}(\boldsymbol{\alpha}) = d$; *faktorijel* multi-indeksa, $\boldsymbol{\alpha}! = \alpha_1! \cdots \alpha_d!$ i nula multi-indeks, $\mathbf{0} = (0, \ldots, 0)$.

Za $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_d) \in \mathbb{N}_0^d$, sa $\boldsymbol{\beta} < \boldsymbol{\alpha}$ označavamo da je $\beta_i < \alpha_i$ za svako $i = 1, \ldots, d$. Sa $\boldsymbol{\beta} \leq \boldsymbol{\alpha}$ označavamo da je $\beta_i \leq \alpha_i$ za svako $i = 1, \ldots, d$. Za multi-indeks $\boldsymbol{\alpha}$ kažemo da je nenegativan ako je $\mathbf{0} \leq \boldsymbol{\alpha}$. Zbir i razlika multi-indeksa $\boldsymbol{\alpha}$ i $\boldsymbol{\beta}$ definišu se kao $\boldsymbol{\alpha} \pm \boldsymbol{\beta} = (\alpha_1 \pm \beta_1, \ldots, \alpha_d \pm \beta_d)$. Za dva multi-indeksa $\boldsymbol{\alpha}$ i $\boldsymbol{\beta}$, *binomni koeficijenti* se definišu kao

$$\binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}} = \frac{\boldsymbol{\alpha}!}{\boldsymbol{\beta}!(\boldsymbol{\alpha} - \boldsymbol{\beta})!}. \tag{1.1}$$

Za multi-indekse iste dimenzije $\boldsymbol{\beta}_n$, $n = 1, \ldots, T$ i $\boldsymbol{\alpha}$ važi $\boldsymbol{\beta}_1 + \cdots + \boldsymbol{\beta}_T = \boldsymbol{\alpha}$, definišemo *multinomne koeficiente* sa

$$\binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T} = \frac{\boldsymbol{\alpha}!}{\boldsymbol{\beta}_1! \cdots \boldsymbol{\beta}_T!}. \tag{1.2}$$

Primetimo da važi

$$\binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}} = \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}, \boldsymbol{\alpha} - \boldsymbol{\beta}}. \tag{1.3}$$

Za vektor $\boldsymbol{z} = (z_1, \ldots, z_d) \in \mathbb{R}^d$ i multi-indeks $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_d) \in \mathbb{N}_0^d$ *stepen vektora* definisan je kao

$$\boldsymbol{z}^{\boldsymbol{\beta}} = z_1^{\beta_1} \cdots z_d^{\beta_d}. \tag{1.4}$$

1.1.3 Multinomna i Lajbnicova formula

Multinomna teorema [73] može da se izrazi pomoću multi-indeksa na sledeći način

$$\Big(\sum_{i=1}^{T} \boldsymbol{z}_i\Big)^{\boldsymbol{\alpha}} = \sum_{\boldsymbol{\beta}_1 + \cdots + \boldsymbol{\beta}_T = \boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T} \prod_{i=1}^{T} \boldsymbol{z}_i^{\boldsymbol{\beta}_i}, \tag{1.5}$$

za vektor $\boldsymbol{z} = (z_1, \ldots, z_d) \in \mathbb{R}^d$ i multi-indeks $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$.

Neka je $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d)$, i neka $C_{\boldsymbol{\alpha}}$ označava skup svih funkcija $u : \mathbb{R}^d \to \mathbb{R}$ koje imaju $\boldsymbol{\alpha}$-ti parcijalni izvod u nuli. Za svako $u \in C_{\boldsymbol{\alpha}}$ definišemo preslikavanje $\mathcal{D}^{(\boldsymbol{\alpha})} : C_{\boldsymbol{\alpha}} \to \mathbb{R}$ sa

$$\mathcal{D}^{(\boldsymbol{\alpha})}\{u\} = \frac{\partial^{|\boldsymbol{\alpha}|} u(\boldsymbol{t})}{\partial^{\alpha_1} t_1 \ldots \partial^{\alpha_d} t_d}\Big|_{\boldsymbol{t}=\mathbf{0}}. \tag{1.6}$$

Primetimo da je $\mathcal{D}^{(\mathbf{0})}\{u(\boldsymbol{t})\} = u(\mathbf{0})$. Prema *generalisanoj Lajbnicovoj formuli* [75] važi sledeća jednakost

$$\mathcal{D}^{(\boldsymbol{\alpha})}\{FG\} = \sum_{\mathbf{0} \leq \boldsymbol{\beta} \leq \boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}} \mathcal{D}^{(\boldsymbol{\beta})}\{F\} \cdot \mathcal{D}^{(\boldsymbol{\alpha}-\boldsymbol{\beta})}\{G\}, \tag{1.7}$$

za sve $F, G \in C_{\boldsymbol{\alpha}}$. Izvodi proizvoda više od dve funkcije mogu se izračunati pomoću [82]

$$\mathcal{D}^{(\boldsymbol{\alpha})}\Big\{\prod_{i=1}^{T} F_i\Big\} = \sum_{\boldsymbol{\beta}_1+\cdots+\boldsymbol{\beta}_T=\boldsymbol{\nu}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_T} \prod_{i=1}^{T} \mathcal{D}^{(\boldsymbol{\beta}_i)}\big\{F_i(\boldsymbol{x})\big\}, \tag{1.8}$$

za svako $F_i \in C_{\boldsymbol{\alpha}}$; $i = 1, \ldots, T$.

1.1.4 Indeksiranje uredjenih n-torki

U ovoj disertaciji bavimo se poluprstenima definisanim nad skupovima uredjenih n-torki, koje su indeksirane multi-indeksima. Skup svih multi-indeksa manjih ili jednakih $\boldsymbol{\nu}$ označava se sa $\mathcal{A}_{\boldsymbol{\nu}}$,

$$\mathcal{A}_{\boldsymbol{\nu}} = \big\{\boldsymbol{\alpha} \in \mathbb{N}_0^{\mathrm{d}(\boldsymbol{\nu})} \mid \boldsymbol{\alpha} \le \boldsymbol{\nu}\big\}, \tag{1.9}$$

a sa $|\mathcal{A}_{\boldsymbol{\nu}}|$ se označava njegova kardinalnost.

Za $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d)$ i $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_d)$ definišemo relaciju leksikografskog poretka $\prec$, tako da je $\boldsymbol{\alpha} \prec \boldsymbol{\beta}$ ako je

$$\alpha_1 = \beta_1, \ldots, \alpha_n = \beta_n \text{ i } \alpha_{n+1} < \beta_{n+1}. \tag{1.10}$$

Neka je $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_d) \in \mathbb{N}_0^d$ multi-indeks i neka su $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_{|\mathcal{A}_{\boldsymbol{\nu}}|}$ multi-indeksi iz $\mathcal{A}_{\boldsymbol{\nu}}$ takvi da $\mathbf{0} = \boldsymbol{\alpha}_1 \prec \boldsymbol{\alpha}_2 \prec \cdots \prec \boldsymbol{\alpha}_{|\mathcal{A}_{\boldsymbol{\nu}}|} = \boldsymbol{\nu}$. Neka je $\boldsymbol{z} = (z_1, \ldots, z_{|\mathcal{A}_{\boldsymbol{\nu}}|}) \in \mathbb{R}^{|\mathcal{A}_{\boldsymbol{\nu}}|}$ i neka je $z : \mathcal{A}_{\boldsymbol{\nu}} \to \mathbb{R}$ funkcija koja svakom $\boldsymbol{\alpha}_i$ iz $\mathcal{A}_{\boldsymbol{\nu}}$ dodeljuje realni broj $z^{(\boldsymbol{\alpha}_i)}$, tako da je $z^{(\boldsymbol{\alpha}_i)} = z_i$. Koristimo sledeću notaciju za vektor $\boldsymbol{z}$

$$\boldsymbol{z} = \big(\, z^{(\boldsymbol{\alpha})} \,\big)_{\boldsymbol{\alpha}\in\mathcal{A}_{\boldsymbol{\nu}}} = (z^{(\boldsymbol{\alpha}_1)}, \ldots, z^{(\boldsymbol{\alpha}_{|\mathcal{A}_{\boldsymbol{\nu}}|})}).$$

1.2 Algebarske strukture

1.2.1 Monoidi i poluprsteni

Monoid je uredjena trojka $(\mathbb{K}, \oplus, 0)$, gde je $\oplus$ asocijativna binarna operacija na skupu $\mathbb{K}$, a 0 je neutralni element za $\oplus$, tj. $a \otimes 0 = 0 \oplus a = a$, za svako $a \in \mathbb{K}$. Monoid je komutativan ako je operacija $\oplus$ komutativna.

Poluprsten je petorka $(\mathbb{K}, \oplus, \otimes, 0, 1)$ takva da

1. $(\mathbb{K}, \oplus, 0)$ je komutativni monoid, a 0 je neutralni element za $\oplus$,
2. $(\mathbb{K}, \otimes, 1)$ je monoid, a 1 je neutralni element za $\otimes$,
3. $\otimes$ distribuira nad $\oplus$, tj. $(a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c)$ and $c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b)$, za svako svako a, b, c iz $\mathbb{K}$,
4. za svako a iz $\mathbb{K}$, važi $a \otimes 0 = 0 \otimes a = 0$.

Poluprsten je komutativan, ako je operacija $\otimes$ komutativna. Operacije $\oplus$ i $\otimes$ se redom zovu sabiranje i množenje u $\mathbb{K}$. Za topologiju τ definišemo *topološki poluprsten* kao par $\big(\mathbb{K}, \tau\big)$.

Definicija 1 *Sum-product poluprsten je petorka* $\big(\mathbb{R}, +\cdot, 1, 0\big)$*, gde je* $\mathbb{R}$ *skup realnih brojeva, operacije* $+$ *i* $\cdot$ *su definisane na standardni način.*

1.2.2 Grafovi

Graf je uredjeni par $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, gde je $\mathcal{V}$ konačan skup *čvorova*, a $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ skup *grana*. Za čvorove α i β kažemo da su *susedi*, ukoliko je $(\alpha, \beta) \in \mathcal{E}$. Skup svih suseda čvora n obeležava se sa $ne(n)$. Kardinalnost skupa $ne(n)$ naziva se *stepen čvora* n i označava se sa $d(n) = |ne(n)|$. Za granu $(\alpha, \beta) \in \mathcal{E}$ kažemo da je *usmerena* ukoliko $(\beta, \alpha) \notin \mathcal{E}$. U ovom slučaju, α je *roditelj* od β, a β je *potomak* od α. Graf se naziva *usmeren* ukoliko su sve njegove grane usmerene, a *neusmeren* ukoliko su sve njegove grane neusmerene.

Put je niz čvorova, takav da za svaka dva uzastopna čvora u nizu α i β važi $(\alpha, \beta) \in \mathcal{E}$. Put se naziva *prost* ukoliko u njemu nema ponovljenih čvorova, u suprotnom naziva se *ciklus*. Graf bez ciklusa naziva se *stablo*. Stablo je *usmereno* ako su sve njegove grane usmerene, a *neusmereno* ukoliko su sve njegove grane neusmerene. Čvorovi stabla koji imaju samo jednog suseda nazivaju se *listovi*. Stablo $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ je *podstablo* stabla $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, ako je $\mathcal{V} \subseteq \mathcal{V}$ i $\mathcal{E}' \subseteq \mathcal{E}$. *Klika* je skup čvorova od kojih su svaka dva susedi.

1.3 Osnovi teorije verovatnoće

1.3.1 Diskretni prostor verovatnoća

Neka je Ω prebrojivi skup *elementarnih dogadjaja*. Komplementarni dogadjaj od A označavamo kao A^C. Presek dogadjaja $A \cap B$ označavamo kao AB ili A, B. Klasa $\mathcal{F}$ dogadjaja čini σ-polje ako

1. $\Omega \in \mathcal{F}$;
2. ako $A \in \mathcal{F}$ tada $A^C \in \mathcal{F}$;
3. ako $A_n \in \mathcal{F}, n = 1, 2, \ldots$ tada $\bigcup_{n=1}^{\infty} A_n \in \mathcal{F}$.

Verovatnoća P definisana nad sigma-poljem dogadaja $\mathcal{F}$ je funkcija iz Ω u $\mathbb{R}$, koja ima sledeće osobine:

1. nenegativnost: za svako $A \in \mathcal{F}$ važi $P(A) \geq 0$;
2. normiranost: $P(\Omega) = 1$;
3. σ-aditivnost: ako su $A_n \in \mathcal{F}, n = 1, 2, \ldots$, uzajamno disjunktni dogadjaji (tj. $A_i \cap A_j$ za $i \neq j$) tada je
$$P\Big(\sum_{n=1}^{\infty} A_n\Big) = \sum_{n=1}^{\infty} P(A_n). \tag{1.11}$$

Prostor verovatnoća je uredjena trojka $(\Omega, \mathcal{F}, P)$. Lako se pokazuje da je partitivni skup prebrojivog skupa σ-polje i u daljem tekstu za $\mathcal{F}$ podrazumevamo partitivni skup od Ω (u zapisu 2^{Ω}).

Neka su $A, B \in \Omega$ i $P(A) > 0$. *Uslovna verovatnoća* dogadjaja B u odnosu na dogadjaj A je funkcija $P : \Omega^2 \to \mathbb{R}$, koja se definiše kao

$$P(B|A) = \frac{P(A \cap B)}{P(A)}. \tag{1.12}$$

Na osnovu definicije, lako se pokazuje da uslovna verovatnoća zadovoljava aksiome verovatnoće i važi pravilo lanca

$$P(A_1,\dots,A_n) = P(A_1)\cdot P(A_2|A_1)\cdots P(A_n|A_1,A_2,\dots,A_{n-1}), \tag{1.13}$$

ukoliko je $P(A_1,A_2,\dots,A_i) > 0$ za svako $i = 1,\dots,n$.

1.3.2 Diskretne slučajne promenljive

Diskretna jednodimenziona (skalarna) slučajna promenljiva X definisana nad $(\Omega, 2^\Omega, P)$ je funkcija koja slika Ω u $\mathbb{X} \subset \mathbb{R}$, tj. $X : \Omega \to \mathbb{X}$. U daljem tekstu slučajne promenljive označavaćemo velikim slovima abecede X, Y, Z i O, sa indeksima ili bez njih, a vrednosti koje one mogu da uzmu malim x, y, z, o. Takodje, za inverznu sliku od $x \in \mathbb{X}$, $X^{-1}(x)$, upotrebljavaćemo oznaku $X = x$.

Lako se pokazuje da slučajna promenljiva definisana nad $(\Omega, 2^\Omega, p)$ definiše prostor verovatnoće $(\mathbb{X}, 2^\mathbb{X}, p_X)$ na sledeći način

$$\forall x \in \mathbb{X}, \quad p_X(x) = P(X = x). \tag{1.14}$$

Funkcija p_X naziva se *raspodela verovatnoća* slučajne promenljive X. Prema tome, kada se posmatra samo slučajna promenljiva X, apstraktni polazni prostor verovatnoća $(\Omega, 2^\Omega, p)$ se može zameniti konkretnim prostorom vrednosti slučajne promenljive X, $(\mathbb{X}, 2^\mathbb{X}, p_X)$. Nadalje podrazumevamo da jednodimenzione slučajne promenljive uzimaju vrednosti iz $\mathbb{X}$.

Ukoliko su $X_1,\dots,X_n$ jednodimenzione slučajne promenljive, $\boldsymbol{X} = (X_1,\dots,X_n)$ se naziva *višedimenziona (vektorska) slučajna promenljiva*. *Združena raspodela verovatnoća* za $\boldsymbol{X}$ definiše se kao

$$p_{X_{1:n}}(x_{1:n}) = P(X_1 = x_1 \cap \cdots \cap X_n = x_n). \tag{1.15}$$

Primetimo da se u slučaju $n = 1$ višedimenziona slučajna promenljiva redukuje na jednodimenzionu slučajnu promenljivu.

Neka je (X, Y) slučajna promenljiva diskretnog tipa sa raspodelom $p_{X,Y}$. *Marginalna raspodela* za X je definisana sa

$$p_X(x) = P(X = x) = P\Big(\bigcup_y (X = x \cap Y = y)\Big) = \sum_y p_{X,Y}(x,y). \tag{1.16}$$

Marginalna raspodela za Y se definiše analogno. Takodje, u slučaju više dimenzionih promenljivih $X_{1:n}$ i $Y_{1:m}$, marginalna raspodela raspodele $p_{X_{1:n},Y_{1:m}}$ za promenljivu $Y_{1:m}$ definiše se kao

$$p_{Y_{1:m}}(y_{1:m}) = \sum_{x_{1:n}} p_{X_{1:n},Y_{1:m}}(x_{1:n}, y_{1:m}). \tag{1.17}$$

Uslovna raspodela za X pri uslovu $Y = y$ ($p_Y(y) > 0$) dobija se neposredno iz definicije uslovne verovatnoće

$$p_X(x|Y = y) = P(X = x|Y = y) = \frac{P(X = x \cap Y = y)}{P(Y = y)} = \frac{p_{X,Y}(x,y)}{p_Y(y)}. \tag{1.18}$$

Uslovna raspodelu za Y pri uslovu $X = x$ ($p_X(x) > 0$) definiše se analogno. Takodje, u slučaju višedimenzionih promenljivih $(X_{1:n}, Y_{1:m})$ imamo

$$p_{X_{1:n}}(x_{1:n}|Y_{1:m} = y_{1:m}) = \frac{p_{X_{1:n},Y_{1:m}}(x_{1:n}, y_{1:m})}{p_{Y_{1:m}}(y_{1:m})}. \tag{1.19}$$

Na osnovu definicije, uslovna raspodela za $X_{1:n}$ pri uslovu $Y_{1:m} = y_{1:m}$ predstavlja raspodelu slučajne promenljive $X_{1:n}$. Takodje, direktno iz definicije sledi *pravilo lanca* za proizvoljnu višedimenzionu slučajnu promenljivu

$$p_{X_{1:n}}(x_{1:n}) = \prod_{i=1}^{n} p_{X_i}(x_i | X_{1:i-1} = x_{1:i-1}). \tag{1.20}$$

U daljem tekstu ćemo izostavljati slučajne promenljive iz zapisa onda kada nema opasnosti od konfuzije. Tako ćemo umesto $p_X(x)$, $p_{X_{1:n}}(x_{1:n})$ i $p_{X_{1:n}}(x_{1:n}|Y_{1:n} = y_{1:n})$, pisati redom $p(x)$, $p(x_{1:n})$ i $p(x_{1:n}|y_{1:n})$.

1.3.3 Matematičko očekivanje

Neka je X diskretna slučajna promenljiva sa raspodelom p_X. *Matematičko očekivanje* $\mathbb{E}[X]$ slučajne promenljive X definiše se sa

$$\mathbb{E}[X] = \sum_x p_X(x) \cdot x, \tag{1.21}$$

pri čemu se, u slučaju prebrojivog skupa vrednosti, koje može da uzme slučajna promenljiva, zahteva $\mathbb{E}[X] < \infty$, što je ispunjeno ako red apsolutno konvergira, odnosno ako je $\sum_x p_X(x) \cdot |x| < \infty$. Funkcija $g : \mathbb{X} \to \mathbb{R}$ definiše diskretnu slučajnu promenljivu $Y = g(X)$, sa raspodelom $p_Y(y) = \sum_{x:f(x)=y} p_X(x)$. Ako $\mathbb{E}[Y]$ postoji, imamo

$$\mathbb{E}[Y] = \sum_y p_Y(y) \cdot y = \sum_y \Big(\sum_{x:f(x)=y} p_X(x) \cdot y \Big) = \sum_y \sum_{x:f(x)=y} p_X(x) \cdot g(x) \Big) = \sum_x p_X(x) \cdot g(x), \tag{1.22}$$

pri čemu je promena redosleda sumiranja dozvoljena, jer red $\sum_y p_Y(y) \cdot y$ apsolutno konvergira. Analogno, neka je $X_{1:n}$ višedimenziona slučajna promenljiva, i neka funkcija $\boldsymbol{g} = [g_1, \ldots, g_m] : \mathbb{X}^n \to \mathbb{R}^m$ definiše m-dimenzionu slučajnu promenljivu $Y_{1:m} = \boldsymbol{g}(X_{1:n}) = [g_1(X_{1:n}), \ldots, g_m(X_{1:n})]$. Matematičko očekivanje slučajne promenljive $Y_{1:m}$ definiše se kao

$$\mathbb{E}[Y_{1:m}] = \mathbb{E}[\boldsymbol{g}(X_{1:n})] = \sum_{x_{1:n}} p_{X_{1:n}}(x_{1:n}) \cdot \boldsymbol{g}(x_{1:n}), \tag{1.23}$$

i nazivaćemo ga matematičko očekivanje funkcije $\boldsymbol{g}$ u odnosu na $p_{X_{1:T}}$. Ukoliko ne postoji opasnost od zabune, pored zapisa $\mathbb{E}[\boldsymbol{g}(X_{1:n})]$, koristićemo i skraćeni zapis $\mathbb{E}[\boldsymbol{g}]$.

1.3.4 Entropija

Koncept Šenonove entropije ima centralnu ulogu u teoriji informacija [14], [77]. *Entropija* slučajne promenljive X sa raspodelom p definiše se kao

$$H(X) = -\sum_x p_X(x) \cdot \ln p_X(x), \tag{1.24}$$

pri čemu je logaritam sa osnovom 2, a entropija se meri u bitovima. Entropija se naziva i mera neizvesnosti, s obzirom na to da predstavlja dobru meru za neodredjenost slučajne promenljive. Primetimo da entropija može biti predstavljena kao očekivana vrednost slučajne promenljive $1/\ln p_X$ nazvane *količina informacija*

$$H(X) = \mathbb{E}[1/\ln p_X(x)]. \tag{1.25}$$

Združena entropija $H(X,Y)$ para slučajnih promenljivih sa združenom raspodelom $p_{X,Y}$ definiše se sa

$$H(X,Y) = \sum_x \sum_y p_{X,Y}(x,y) \cdot \ln p_{X,Y}(x,y), \tag{1.26}$$

a *uslovna entropija* $H(Y|X)$ sa

$$H(Y|X) = \sum_x p_X(x) \cdot H(Y|X=x), \tag{1.27}$$

gde je

$$H(Y|X=x) = \sum_y p_Y(y|X=x) \cdot \ln p_Y(y|X=x). \tag{1.28}$$

Korišćenjem pravila lanca za uslovne verovatnoće dokazuje se pravilo lanca za združenu entropiju

$$H(X,Y) = H(X) + H(Y|X). \tag{1.29}$$

Analogno, u slučaju višedimenzione slučajne promenljve $X_{1:n}$ sa raspodelom $p_{X_{1:n}}$ imamo

$$H(X_{1:n}) = -\sum_{x_{1:n}} p_{X_{1:n}}(x_{1:n}) \cdot \ln p_{X_{1:n}}(x_{1:n}). \tag{1.30}$$

Za proizvoljne višedimenzione promenljive definiše se uslovna entropija $H(X_{1:n}|Y_{1:m})$ pomoću

$$H(X_{1:n}|Y_{1:m}) = \sum_{x_{1:n}} p_{X_{1:n}}(x_{1:n}) \cdot H(Y_{1:m}|X_{1:n}=x_{1:n}), \tag{1.31}$$

gde je

$$H(Y_{1:m}|X_{1:n}=x_{1:n}) = \sum_{y_{1:n}} p_{Y_{1:n}}(y_{1:m}|X_{1:n}=x_{1:n}) \cdot \ln p_{Y_{1:n}}(y_{1:m}|X_{1:n}=x_{1:n}). \tag{1.32}$$

Primenom pravila lanca na združenu raspodelu $p_{X_{1:n}}$ dobija se pravilo lanca za višedimenzioni slučaj

$$H(X_{1:n}) = \sum_{i=1}^{n} H(X_i|X_{1:i-1}). \tag{1.33}$$

1.3.5 Kros-momenti slučajne promenljive

Neka je $\boldsymbol{Y} = Y_{1:d}$ višedimenziona slučajna promenljiva sa raspodelom $p_{\boldsymbol{Y}}$. *Kros-moment* reda $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d)$ slučajne promenljive $\boldsymbol{Y}$, u oznaci $\mu_{\boldsymbol{Y}}^{(\boldsymbol{\alpha})}$, definiše se kao

$$\mu_{p,\boldsymbol{Y}}^{(\boldsymbol{\alpha})} = \sum_{\boldsymbol{y}} p_{\boldsymbol{Y}}(\boldsymbol{y}) \cdot \boldsymbol{y}^{\boldsymbol{\alpha}}. \tag{1.34}$$

Dalje, neka je $\boldsymbol{g} = [g_1, \ldots, g_d] : \mathbb{X}^T \to \mathbb{R}^d$ funkcija slučajne promenljive $X_{1:T}$. Tada ona definiše d-dimenzionu slučajnu promenljivu $\boldsymbol{Y} = Y_{1:d} = \boldsymbol{g}(X_{1:T}) = [g_1(X_{1:T}), \ldots, g_d(X_{1:T})]$. Može se pokazati (videti [41]) da je kros-moment reda $\boldsymbol{\alpha}$ slučajne promenljive $\boldsymbol{Y}$, u oznaci $\mu_{p,\,\boldsymbol{g}}^{(\boldsymbol{\alpha})}$, jednak

$$\mu_{p,\,\boldsymbol{g}}^{(\boldsymbol{\alpha})} = \sum_{x_{1:T}} p(x_{1:T}) \cdot \boldsymbol{g}(x_{1:T})^{\boldsymbol{\alpha}}. \tag{1.35}$$

Funkcija generatrise momenta (*moment generating function*, *MGF*) funkcije $\boldsymbol{g}$, u odnosu na f, je realna funkcija $M_{p,\boldsymbol{g}} : \mathbb{R}^d \to \mathbb{R}$ definisana sa [57]

$$M_{p,\boldsymbol{g}}(\boldsymbol{t}) = \sum_{x_{1:T}} p(x_{1:T}) \cdot \boldsymbol{e}^{\,\boldsymbol{g}(x_{1:T})\cdot \boldsymbol{t}}, \tag{1.36}$$

za svako $\boldsymbol{t} \in \mathbb{R}^d$. Ukoliko funkcionalni red u izrazu (1.36) uniformno konvergira u nekoj okolini nule, kros-moment reda $\boldsymbol{\alpha}$ može se izračunati kao parcijalni izvod *MGF* [29]

$$\mu^{(\boldsymbol{\alpha})}_{p,\,\boldsymbol{g}} = \mathcal{D}^{(\boldsymbol{\alpha})} \left\{ M_{p,\boldsymbol{g}}(\boldsymbol{t}) \right\}_{\boldsymbol{t}=0}. \tag{1.37}$$

Ekvivalentno, *MGF* se može izraziti preko Tejlorovog reda

$$M_{p,\boldsymbol{g}}(\boldsymbol{t}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \frac{\mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{g}}}{\boldsymbol{\alpha}!} \cdot \boldsymbol{t}^{(\boldsymbol{\alpha})}. \tag{1.38}$$

Kros-momente jednodimenzionih slučajnih promenljivih nazivamo *momenti*. U specijalnom slučaju, moment prvog reda skalarne funkcije $\boldsymbol{g} : \mathbb{X}^T \to \mathbb{R}^d$ slučajne promenljive $X_{1:T}$ svodi se na matematičko očekivanje od $\boldsymbol{g}(X_{1:T})$.

1.4 Izračunavanje kros-momenata za funkcije jednostavne strukture

U ovom poglavlju na primeru funkcija jednostavne strukture pokazujemo kako se na efikasan način mogu izračunati kros-momenti primenom distributivnog zakona. Pristup izložen u ovom poglavlju biće korišćen u narednim glavama.

1.4.1 Kros-momenati funkcije jednostavne strukture

Direktno izračunavanje kros-momenata

$$M_{p,\boldsymbol{g}}(\boldsymbol{t}) = \sum_{x_{1:T}} p(x_{1:T}) \cdot \boldsymbol{e}^{\,\boldsymbol{g}(x_{1:T})\cdot \boldsymbol{t}}, \tag{1.39}$$

enumeracijom svih vrednosti iz $\mathbb{X}^T$ zahteva $\mathcal{O}(|\mathbb{X}|^T)$ operacija, zapisano u *Landau* $\mathcal{O}$-notaciji [12]. Ovakav postupak može biti izuzetno zahtevan ukoliko je $|\mathbb{X}|$ veliko. Medjutim, za specifičnu strukturu raspodele p i funkcije $\boldsymbol{g}$, ovaj problem može biti rešen primenom distributivnog zakona u odgovarajućem poluprstenu. Na primer, neka za raspodelu p i funkciju $\boldsymbol{g}$ važi

$$p(x_{1:T}) = \prod_{t=1}^{T} \phi_t(x_t), \quad \boldsymbol{g}(x_{1:T}) = \sum_{t=1}^{T} \boldsymbol{g}_t(x_t), \tag{1.40}$$

gde su $\phi_t : \mathbb{X} \to \mathbb{R}$ i $\boldsymbol{g}_t : \mathbb{X} \to \mathbb{R}^d$ za $t = 1, \ldots, T$. Tada odgovarajući kros-momenti imaju oblik

$$\mu^{(\boldsymbol{\alpha})}_{p,\,\boldsymbol{g}} = \sum_{x_{1:T}} \prod_{t=1}^{T} \phi_t(x_t) \cdot \Big(\sum_{t=1}^{T} \boldsymbol{g}_t(x_t) \Big)^{\boldsymbol{\alpha}}. \tag{1.41}$$

U ovom poglavlju, na primeru funkcija jednostavne strukture (1.40) pokazujemo kako se distributivni zakon u poluprstenu polinoma može iskoristiti za izračunavanje kros-momenata (1.41). U narednim glavama disertacije razmatraćemo složenije funkcije i druge poluprstenove.

1.4.2 Generalisani distributivni zakon

Neka promenljiva $x_{1:T}$ uzima vrednosti iz skupa $\mathbb{X}^T$, i neka se funkcija $w : \mathbb{X}^T \to \mathbb{K}$ faktoriše u komutativnom poluprstenu $(\mathbb{K}, \oplus, \otimes, 0, 1)$ kao

$$w(x_{1:T}) = \bigotimes_{t=1}^{T} w_t(x_t), \tag{1.42}$$

gde su $w_t : \mathbb{X} \to \mathbb{K}$. Direktno izračunavanje sume $\bigoplus_{x_{1:T}} w(x_{1:T})$ enumerisanjem svih mogućih vrednosti za $x_{1:T}$ (*brute-force* izračunavanje), zahteva $\mathcal{O}(|\mathbb{X}|^T)$ operacija. S druge strane, posle primene distirbutivnog zakona

$$\bigoplus_{x_{1:T}} w(x_{1:T}) = \bigoplus_{x_{1:T}} \bigotimes_{t=1}^{T} w_t(x_t) = \bigotimes_{t=1}^{T} \bigoplus_{x_t} w_t(x_t), \tag{1.43}$$

broj operacija se redukuje na $\mathcal{O}(|\mathbb{X}| \cdot T)$. Na ovaj način dobili smo "brzi algoritam" za izračunavanje sume (1.42) koji nazivamo *generalisani distributivni zakon*. U odeljku 1.4.4 pokazujemo kako se primenom ovog algoritma mogu izračunati kros-momenti za funkcije sa jednostavnom strukturom.

1.4.3 Poluprsten polinoma

Poluprsten polinoma reda $\boldsymbol{\nu}$ je petorka $(\mathbb{R}^{|\mathcal{A}_\nu|}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$, gde su operacije $\oplus$ i $\otimes$ definisane sa

$$u \oplus v = \left(u^{(\boldsymbol{\alpha})} + v^{(\boldsymbol{\alpha})}\right)_{\boldsymbol{\alpha} \in A_\nu},$$
$$u \otimes v = \left(\sum_{\boldsymbol{\beta}+\boldsymbol{\gamma}=\boldsymbol{\alpha}} u^{(\boldsymbol{\beta})} \cdot v^{(\boldsymbol{\gamma})}\right)_{\boldsymbol{\alpha} \in A_\nu},$$

za svako $u, v \in \mathbb{R}^{|\mathcal{A}_\nu|}$. Neutralni elementi za $\oplus$ i $\otimes$ su dati sa

$$\mathbf{0} = (\underbrace{0, 0, \ldots, 0}_{|\mathcal{A}_\nu| \text{ times}}), \tag{1.44}$$

$$\mathbf{1} = (1, \underbrace{0, \ldots, 0}_{|\mathcal{A}_\nu|-1 \text{ times}}). \tag{1.45}$$

Sledeća lema se jednostavno dokazuje indukcijom.

Lema 1.4.1 *Neka je* $w_t \in \mathbb{R}^{|\mathcal{A}_\nu|}$; $t = 1, \ldots, T$. *Tada važi sledeća jednakost*

$$\left(\bigoplus_{t=1}^{T} w_t\right)^{(\boldsymbol{\alpha})} = \sum_{t=1}^{T} w_t^{(\boldsymbol{\alpha})}, \tag{1.46}$$

$$\left(\bigotimes_{t=1}^{T} w_t\right)^{(\boldsymbol{\alpha})} = \sum_{\boldsymbol{\beta}_1+\cdots+\boldsymbol{\beta}_T=\boldsymbol{\alpha}} \prod_{t=1}^{T} w_t^{(\boldsymbol{\beta}_t)}. \tag{1.47}$$

1.4.4 Izračunavanje kros-momenata u poluprstenu polinoma

Neka je $(\mathbb{K}, \oplus, \otimes, 0, 1)$ poluprsten polinoma, i neka je

$$w_t = \Big(\frac{\phi_t(x_t) \cdot \boldsymbol{g}_t(x_t)^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \Big)_{\boldsymbol{\alpha} \in A_v}, \tag{1.48}$$

za $t = 1, \ldots, T$. Na osnovu leme 1.4.1, važi

$$\begin{aligned}
\Big(\bigotimes_{t=1}^{T} w_t(x_t) \Big)^{(\boldsymbol{\alpha})} &= \sum_{\boldsymbol{\beta}_1 + \cdots + \boldsymbol{\beta}_T = \boldsymbol{\alpha}} \prod_{t=1}^{T} w_t^{(\boldsymbol{\beta}_t)}(x_t) = \sum_{\boldsymbol{\beta}_1 + \cdots + \boldsymbol{\beta}_T = \boldsymbol{\alpha}} \prod_{t=1}^{T} \frac{\phi_t(x_t) \cdot \boldsymbol{g}_t(x_t)^{\boldsymbol{\beta}_t}}{\boldsymbol{\beta}_t!} = \\
&= \frac{1}{\boldsymbol{\alpha}!} \cdot \prod_{t=1}^{T} \phi_t(x_t) \cdot \sum_{\boldsymbol{\beta}_1 + \cdots + \boldsymbol{\beta}_T = \boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T} \prod_{t=1}^{T} \boldsymbol{g}_m(x_t)^{\boldsymbol{\beta}_t} = \\
&= \frac{1}{\boldsymbol{\alpha}!} \cdot \prod_{t=1}^{T} \phi_t(x_t) \cdot \Big(\sum_{t=1}^{T} \boldsymbol{g}_t(x_t) \Big)^{\boldsymbol{\alpha}},
\end{aligned} \tag{1.49}$$

a posle sumiranja u poluprstenu polinoma dobijamo

$$\bigoplus_{x_{1:T}} \bigotimes_{t=1}^{T} w_t(x_t) = \Big(\frac{\mu_{p,\,g}^{(\boldsymbol{\alpha})}}{\boldsymbol{\alpha}!} \Big)_{\boldsymbol{\alpha} \in A_v}. \tag{1.50}$$

Na ovaj način dobili smo algoritam za izračunavanje svih kros-momenata, zaključno sa redom $\boldsymbol{\alpha}$, u slučaju funkcija sa jednostavnom strukturom (1.40), primenom generalisanog distributivnog zakona u poluprstenu polinoma. Ova ideja će u narednim glavama biti primenjivana za funkcije koje imaju komplikovaniju strukturu. Najpre razmatramo funkcije čija se struktura opisuje faktor-grafovima, a umesto generalisanog distributivnog zakona koristimo algoritam slanja poruka nad faktor-grafom [1], [15], [49], [68]. Nakon toga razmatramo funkcije koje su opisane probabilističkim kontekstno-nezavisnim gramatikama, uz pomoć *inside* algoritma [26], [53], i algoritma za izračunavanje particione konstante [66].

Glava 2

Izračunavanje matematičkog očekivanja vektorske slučajne promenljive

Grafički modeli [15], su tip probabilističkih mreža koji vuku korene iz različitih naučnih disciplina kao što su: veštačka inteligencija [69], statistika [55] i obrada signala [88]. Grafički modeli koriste graf za reprezentaciju strukture združene raspodele višedimenzione slučajne promenljive, pri čemu se pod strukturom podrazumeva njeno dekompoziciono svojstvo. Odgovarajući graf može biti usmeren, kao što je slučaj kod Bajesovskih mreža, ili neusmeren, kao što je slučaj kod Markovljevih slučajnih polja, *junction* stabala i faktor-grafova. Pomenuti tipovi grafičkih modela opisani su u poglavlju 2.1.

Algoritam slanja poruka (*message passing algorithm, MP*) je najpoznatiji algoritam nad grafičkim modelima. *MP* algoritam na efikasan način izračunava marginalne vrednosti funkcije koja uzima vrednosti iz komutativnog poluprstena i može biti opisan kao postupak slanja poruka kros graf pridružen funkciji. Prvu verziju algoritma koja je operisala nad Bajesovskim mrežama dao je *Pearl* [68] i poznata je kao *belief propagation* algoritam. Nešto kasnije , *Lauritzen* i *Spiegelhalter* daju verziju algoritma koji funkcioniše nad *junction* stablima [55], [54], [43]. Vezu izmedju *belief propagation* algoritma i algoritama koji se koriste za dekodovanje zašitinih kodova [88] dao je *McEliece* sa saradnicima u poznatom radu [61].

U poglavlju 2.2 razmatramo *MP* algoritam u slučaju funkcija koje su opisane faktor-grafom, ali je princip funkcionisanja algoritma isti za sve tipove grafičkih modela [1], [15], [49], [68]. U slučaju faktor-grafa bez ciklusa, *MP* algoritam daje tačan rezultat, ali u odredjenim poluprstenima i za odredjene strukture faktor-grafa, *MP* algoritam može biti primenjen i za grafove sa ciklusima, pri čemu se dobija aproksimativno rešenje [64], [86], [89]. U ovoj glavi razmatramo faktor-grafove bez ciklusa.

Glavna tema ove glave je primena algoritma slanja poruka za izračunavanje očekivanja vektorske slučajne promenljive kojoj odgovara graf bez ciklusa. U principu, moguća su dva načina za izračunavanje kros-momenata. Prvi, u kome se uz pomoć *MP* algoritma izračunavaju marginalne vrednosti, a zatim u posebnom prolazu, uz pomoć izračunatih marginalnih vrednosti, izračunavaju se kros-momenti. Ovakav pristup razmatran je u radovima [16], [17], [18], [21], za faktor-grafove, i u radu [67], za *junction* stabla. Iako naizgled najjednostavniji, ovaj metod zahteva pamćenje svih marginalnih vrednosti, zbog čega algoritam ima memorijsku kompleksnost proporcionalnu broju marginalnih vrednosti, odnosno veličini grafa nad kojim se izvršava algoritam. Drugu mogućnost, predstavlja izračunavanje očekivanja u jedinstvenom prolazu primenom *MP* algoritma koji operiše nad

poluprstenom očekivanja (*EMP* algoritam, poglavlje 2.3). *EMP* algoritam ima znatno manji utrošak memorije u odnosu na prvi pristup, uz istu vremensku kompleksnost kao u prvom slučaju. Ovakav pristup korišćen je za izračunavanje momenata nad *junction* stablima u [59], a u poglavlju 2.3 pokazaćemo kako se mogu izračunati kros-momenti nad faktor-grafovima.

2.1 Grafički modeli

Kao što smo pomenuli, grafički modeli koriste graf za reprezentaciju strukture združene raspodele višedimenzione slučajne promenljive, pri čemu se pod strukturom podrazumeva njeno dekompoziciono svojstvo. U zavisnosti od toga da li je graf usmeren ili neusmeren, govorimo o usmerenom, odnosno neusmerenom grafičkom modelu. U ovom poglavlju opisujemo jedan usmeren grafički model, Bajesovske mreže, i tri neusmerena grafička modela: Markovljeva polja, *junction* stabla i faktor-grafove.

2.1.1 Bajesovske mreže

Bajesovske mreže [15], [69] su grafički modeli bazirani na usmerenom grafu. U kombinaciji sa *belief propagation* algoritmom koji je razvio *Pearl* [69], postale su bitan alat u ekspertskim sistemima. Neka je $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ usmereni graf sa skupom čvorova $\mathcal{V} = \{1, 2, \ldots, T\}$ i skupom grana $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Svaki čvor $v \in \mathcal{V}$ odgovara slučajnoj promenljivoj X_v. Neka je skup roditelja čvora v

$$\mathcal{P}(v) = \left\{u \in \mathcal{V} | (u, v) \in \mathcal{E}\right\}, \tag{2.1}$$

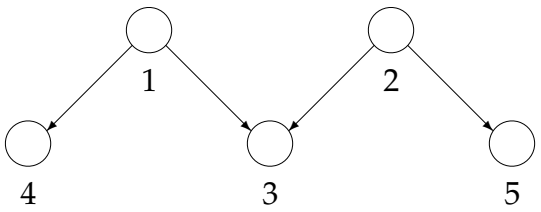
i neka je moguće predstaviti raspodelu $p_{X_{1:T}}$ kao proizvod

$$p_{X_{1:T}}(x_1, x_2, \ldots, x_T) = \prod_{i=1}^{T} p(x_i | x_{\mathcal{P}(i)}), \tag{2.2}$$

pri čemu za čvorove bez roditelja usvajamo $p(x_v | x_\varnothing) = p(x_v)$. Tada se par $(\mathcal{G}, p_{X_{1:T}})$ naziva Bajesovska mreža. Primer Bajesovske mreže koja odgovara raspodeli

$$p(x_1, x_2, x_3, x_4, x_5) = p(x_1)p(x_2)p(x_3|x_1, x_2)p(x_4|x_1)p(x_5|x_2), \tag{2.3}$$

data je na slici 2.1.



Slika 2.1: Bajesovska mreža koja odgovara raspodeli
$p(x_1)p(x_2)p(x_3|x_1, x_2)p(x_4|x_1)p(x_5|x_2)$

2.1.2 Markovljeva slučajna polja

Markovljeva slučajna polja (Markov random fields, MRF) su široko primenjivani probabilistički grafički modeli [40], [46], [69]. Definišu se na sledeći način. Neka je $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ neusmereni graf sa skupom čvorova $\mathcal{V} = \{1, 2, \ldots, N\}$ i skupom grana $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Neka je $p_{X_{1:T}}$ raspodela vektorske slučajne promenljive $X_{1:T}$. Za uredjeni par $(\mathcal{G}, p_{X_{1:T}})$ kažemo da je *MRF*, ukoliko je za svaki čvor $v \in \mathcal{V}$ zadovoljeno lokalno Markovljevo svojstvo

$$p(X_v | X_{\mathcal{V} \setminus v}) = p(X_v | X_{\mathcal{N}(v)}), \tag{2.4}$$

gde je $\mathcal{N}(v)$ skup suseda čvora v. Drugim rečima, $(\mathcal{G}, p_{X_{1:T}})$ je *MRF*, ako je svaka slučajna promenljiva sa indeksom v nezavisna od slučajnih promenljivih sa indeksima koji nisu susedi od v, pod uslovom da su $X_{\mathcal{N}(v)}$ date.

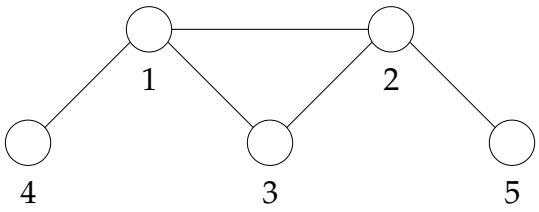
Pod relativno slabom pretpostavkom (kao što je pozitivnost raspodele verovatnoće), združena raspodela *MRF*-a može se izraziti kao proizvod faktora skupa *Gibsovih potencijala*, definisan na skupu klika u $\mathcal{G}$, označenog sa $\mathcal{C}$,

$$p_{X_{1:T}}(x_{1:T}) = Z^{-1} \prod_{C \in \mathcal{C}} f_C(X_C), \tag{2.5}$$

gde je Z normalizaciona konstanta i svako $C \in \mathcal{C}$ je klika. Na primer, u slučaju raspodele (2.3), imamo sledeću faktorizaciju

$$p(x_1, x_2, x_3, x_4, x_5) = f_{123}(x_1, x_2, x_3) f_{14}(x_1, x_4) f_{25}(x_2, x_5), \tag{2.6}$$

i odgovarajući graf na slici 2.2.



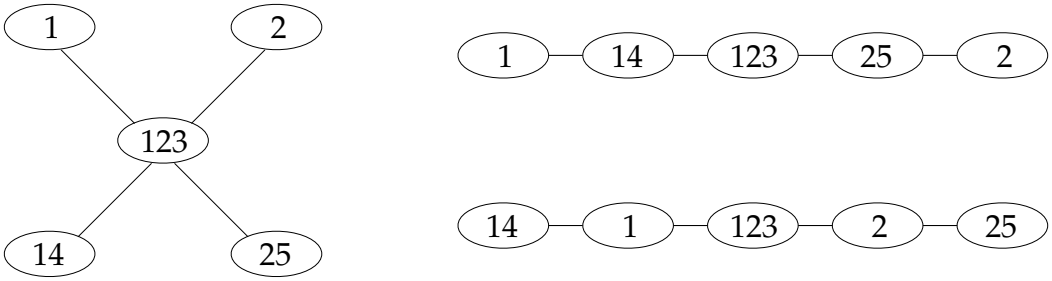
Slika 2.2: Markovljevo slučajno polje koje odgovara raspodeli $p(x_1)p(x_2)p(x_3|x_1, x_2)p(x_4|x_1)p(x_5|x_2)$

2.1.3 *Junction* stabla

Neka je $\mathcal{M}$ kolekcija podskupova skupa $\{0, \ldots, T\}$ i neka je $\mathcal{G} = (\mathcal{M}, \mathcal{E})$ neusmereno stablo sa skupom čvorova $\mathcal{M}$ i skupom grana $\mathcal{E} \subseteq \mathcal{M} \times \mathcal{M}$. Stablo $\mathcal{G}$ naziva se *junction stablo* ako se svaki presek $A \cap B$ proizvoljnog para A, B skupova iz $\mathcal{M}$ sadrži u svim čvorovima na jedinstvenom putu izmedju A i B. Ekvivalentno, za svako $k \in \{0, \ldots, T\}$, skup svih podskupova u $\mathcal{M}$, koji sadrže k, indukuje povezano podstablo od $\mathcal{G}$.

Da bismo predstavili izraze (2.2) i (2.5) *junction* stablom, predstavimo ih najpre u generalnijem obliku. Neka promenljiva $x_{1:T}$ uzima vrednosti iz skupa $\mathbb{X}^T$, i neka se funkcija $f : \mathbb{X}^T \to \mathbb{K}$ faktoriše u komutativnom poluprstenu $(\mathbb{K}, \oplus, \otimes, 0, 1)$ kao

$$f(x_{1:T}) = \bigotimes_{M \in \mathcal{M}} f_M(x_M), \tag{2.7}$$



Slika 2.3: Moguća junction stabla koja odgovaraju faktorizaciji $f_A(x_1) \otimes f_B(x_2) \otimes f_C(x_1, x_2, x_3) \otimes f_D(x_1, x_4) \otimes f_E(x_2, x_5)$.

gde je $\mathcal{M}$ kolekcija podskupova skupa $\{0, \ldots, T\}$. Na primer, faktorizacija (2.3) se u *sum-product* poluprstenu može zapisati kao

$$f(x_1, x_2, x_3, x_4, x_5) = f_A(x_1) \otimes f_B(x_2) \otimes f_C(x_1, x_2, x_3) \otimes f_D(x_1, x_4) \otimes f_E(x_2, x_5), \tag{2.8}$$

gde je $A = \{1\}$, $B = \{2\}$, $C = \{1, 2, 3\}$, $D = \{1, 4\}$ i $E = \{2, 5\}$.

Faktorizacija se predstavlja *junction* stablom, tako što se svakom od skupova M dodeli po jedan čvor. U slučaju da za datu faktorizaciju skup $\mathcal{M}$ ne može biti organizovan u *junction* stablo, moguće je proširiti domene faktora i funkciju prtetstaviti ekvivalentnom faktorizacijom za koju *junction* stablo postoji. S druge strane, moguće je da za odredjenu faktorizaciju postoji više različitih *junction* stabala. Na ovom mestu nećemo ulaziti u detalje u vezi sa izborom najprihvatljivijeg *junction* stabla, kao ni u pitanja konstrukcije stabla, a čitalac se upućuje na [42], [44], [55] i [1].

Primeri *junction* stabala za faktorizaciju (2.8) dati su na slici 2.3.

2.1.4 Faktor-grafovi

Faktor-graf je uredjena trojka $(\mathcal{M}, \mathcal{N}, \mathcal{E})$ skupa *faktor-čvorova* $\mathcal{M}$, skupa *čvorova promenljivih* $\mathcal{N}$ i skupa grana $\mathcal{E}$, takva da:

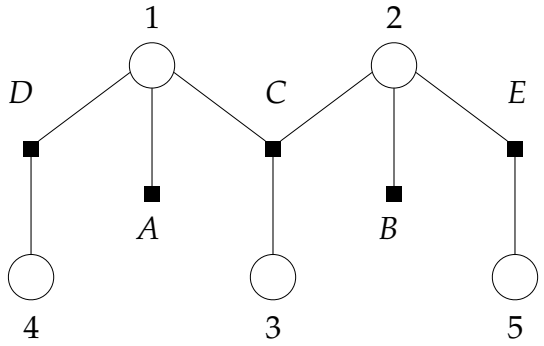
- $\mathcal{M} \cap \mathcal{N} = \varnothing$,
- $(\mathcal{V}, \mathcal{E})$ je neusmereni graf, gde je $\mathcal{V} = \mathcal{M} \cup \mathcal{N}$,
- $(\alpha, \beta) \in \mathcal{E}$ ako i samo ako je $\alpha \in \mathcal{N}, \beta \in \mathcal{M}$ ili $\alpha \in \mathcal{M}, \beta \in \mathcal{N}$.

Prilikom grafičke reprezentacije, faktor-čvorovi se predstavljaju kvadratrima, čvorovi promenljivih krugovima, a grane vezama izmedju čvorova (slika 2.4). Primetimo da za stepen faktor-čvora važi jednakost $d(M) = |x_M|$. Za sume stepena faktor-čvorova i čvorova promenljivih važi sledeća jednakost

$$\sum_{M \in \mathcal{M}} d(M) = \sum_{n \in \mathcal{N}} d(n). \tag{2.9}$$

Poslednja jednakost sledi iz činjenice da obe strane u izrazu (2.9) izračunavaju broj grana u faktor-grafu, s obzirom na to da se grane nalaze isključivo izmedju faktor-čvorova i čvorova promenljivih.

Faktorizacija (2.7) se može vizuelizovati uz pomoć faktor-grafa u kome svakoj promenljivoj x_n odgovara čvor promenljive n, fakoru f_M odgovara faktor-čvor M, a grana izmedju čvora promenljive n i faktor-čvora M, označava da faktor f_M zavisi od promenljive n [6], [49], [58]. Primer faktor-grafa koji odgovara faktorizaciji (2.8) dat je na slici 2.4.



Slika 2.4: Faktor-graf za $f_A(x_1) \otimes f_B(x_2) \otimes f_C(x_1, x_2, x_3) \otimes f_D(x_1, x_4) \otimes f_E(x_2, x_5)$.

2.2 Algoritam slanja poruka nad faktor-grafom

Algoritam slanja poruka (*message passing algorithm, MP*) na efikasan način izračunava marginalne vrednosti funkcije i može biti opisan kao postupak slanja poruka nad grafom pridruženom funkciji. U ovom poglavlju razmatramo *MP* algoritam u slučaju funkcija koje su opisane faktor-grafom, ali je princip funkcionisanja algoritama isti za sve tipove grafičkih modela [1], [15], [49], [68]. U slučaju faktor-grafa bez ciklusa, *MP* algoritam daje tačan rezultat, ali u odredjenim poluprstenima i za odredjene strukture faktor-grafa, *MP* algoritam može biti primenjen i za grafove sa ciklusima, pri čemu se dobija aproksimativno rešenje [64], [86], [89]. U ovom radu razmatramo faktor-grafove bez ciklusa.

2.2.1 *MP* algoritam

MP algoritam može da se koristi za rešavanje sledećih problema:

- **Marginalizacija u čvorovima promenljivih:** Izračunati *marginale u čvorovima promenljvih*

$$Z_n(x_n) = \bigoplus_{x_{-n}} \bigotimes_{M \in \mathcal{M}} f_M(x_M), \tag{2.10}$$

  za sve čvorove promenljivih u faktor-grafu, pri čemu $\oplus_{x_{1:T} \setminus x_{-n}}$ označava sumiranje po svim promenljavama iz $x_{1:T}$ osim po x_n.

- **Marginalizacija u faktor-čvorovima:** Izračunati *marginale u faktor-čvorovima*

$$\widetilde{Z}_M(x_M) = \bigoplus_{x_{-M}} f(x_{1:T}), \tag{2.11}$$

  za sve faktor-čvorove u grafu, pri čemu $\oplus_{x_{-M}}$ označava sumiranje po svim promenljivama iz $x_{1:T}$ osim po $x_n, n \in M$.

- **Normalizacioni problem:** Izračunati *normalizacionu konstantu*

$$Z = \bigoplus_{x_{1:T}} \bigotimes_{M \in \mathcal{M}} f_M(x_M). \tag{2.12}$$

Algoritam se može opisati kao proces slanja poruka kroz grane i procesiranje poruka u čvorovima faktor-grafa. Postoje dva tipa poruka:

1. *poruke* $q_{n\to M}(x_n) : \mathbb{X} \to \mathbb{K}$, iz čvorova promenljivih u faktor-čvorove i
2. *poruke* $r_{M\to n}(x_n) : \mathbb{X} \to \mathbb{K}$, iz faktor-čvorova u čvor promenljive,

pri čemu su čvorovi promenljivih i faktor-čvorovi koji učestvuju u procesu slanja poruka označeni sa n i M, respektivno.

Poruke su funkcije promenljive koja odgovara čvoru promenljive koja učestvuje u procesu slanja, i nose informaciju o podstablu iz koga su poslate. Uklanjanjem grane izmedju faktor-čvora M i čvora promenljive n, faktor-graf se razbija na dva podstabla. Neka je $\mathcal{M}(M,n)$ skup svih faktor-čvorova u podstablu koje sadrži M, a $\mathcal{M}(n,M)$ skup svih faktor-čvorova u podstablu koje sadrži n. Tada je poruka iz čvora promenljive n u faktor-čvor M

$$q_{n\to M}(x_n) = \bigoplus_{x_{\mathcal{M}(n,M)\setminus n}} \bigotimes_{K\in\mathcal{M}(n,M)} f_K(x_K), \tag{2.13}$$

dok je poruka iz faktor-čvora M u čvor promenljive n data sa

$$r_{M\to n}(x_n) = \bigoplus_{x_{\mathcal{M}(M,n)\setminus n}} \bigotimes_{K\in\mathcal{M}(M,n)} f_K(x_K). \tag{2.14}$$

Algoritam se sastoji iz *kolekcione* i *distribucione* faze.

Kolekciona faza algoritma započinje po izboru jednog od listova za *koren stabla* i usmeravanjem grana od ostalih listova prema korenu. Poruke iz čvora promenljive se inicijalizuju na jediničini element u poluprstenu, $q_{n\to M}(x_n) = 1$, a poruka iz faktor-čvora u čvor promenljive na odgovarajući faktor $r_{M\to n}(x_n) = f_M(x_n)$, pri čemu smo uzeli u obzir da faktor kome odgovara list zavisi od jedne promenljive, $M = \{n\}$. Inicijalnizaija se obavalja za sve čvorove promenljivih n i faktor-čvorove M u listovima faktor-grafa, za sve moguće vrednosti x_n. Poruka iz čvora ka njegovom roditelju se izračunava po prijemu svih poruka od potomaka na sledeći način

$$q_{n\to M}(x_n) = \bigotimes_{M'\in ne(n)\setminus M} r_{M'\to n}(x_n), \tag{2.15}$$

$$r_{M\to n}(x_n) = \bigoplus_{x_{M\setminus n}} f_M(x_M) \otimes \bigotimes_{n'\in ne(M)\setminus n} q_{n'\to M}(x_{n'}). \tag{2.16}$$

Kolekciona faza algoritma se okončava pošto koren stabla n primi poruku od jedinog potomka M. U ovom trenutku, normalizaciona konstanta se na osnovu (2.14) može izračunati kao

$$Z = \bigoplus_{x_n} r_{M\to n}(x_n). \tag{2.17}$$

Distribuciona faza algoritma služi za izračunavanje svih marginala u faktor-čvorovima i čvorovima promenljivih. Izvršava se po okončanoj kolekcionoj fazi u kojoj su sve poslate poruke zapamćene. U distribucionoj fazi poruke se šalju od korena ka listovima. Pošto primi poruku od roditelja, čvor šalje poruke svim potomcima, pri čemu se poruke izračunavaju saglasno izrazima (2.15)-(2.16). Onog trenutka kada čvor primi poruku od roditelja, marginalna vrednost za taj čvor može se izračunati kao

$$Z_n(x_n) = \bigotimes_{M\in ne(n)} r_{M\to n}(x_n), \tag{2.18}$$

u slučaju čvora promenljive, odnosno kao

$$Z_M(x_M) = f(x_M) \otimes \bigotimes_{n \in ne(M)} q_{n \to M}(x_n), \tag{2.19}$$

u slučaju faktor-čvora. Distribuciona faza se okončava kada svi listovi prime poruke od roditelja, i u tom trenutku svi marginali su izračunati.

2.2.2 Vremenska i memorijska kompleksnost *MP* algoritma

U ovom odeljku razmatramo asimptotsku *vremensku* (T) i *memorijsku* (M) kompleksnost *MP* algoritma kada $|\mathcal{N}|$, $|\mathcal{M}|$, i $|\mathbb{X}|$ teže beskonačnosti. Vremenska kompleksnost algoritma definisana je kao asimptotski broj operacija u poluprstenu (sabiranja i množenja) potrebnih za izvršenje algoritma. Memorijska kompleksnost definiše se kao maksimalni memorijski prostor koji mora biti alociran za reprezentaciju poruka, meren u broju elemenata odgovarajućeg poluprstena. Prilikom analize koristićemo *Landau* $\mathcal{O}$-notaciju [12].

Normalizacioni problem

Vremenska kompleksnost algoritma za rešavanje normalizacionog problema dobija se kao asimptotski broj operacija potrebnih za izvršenje kolekcione faze, saglasno formulama (2.15)-(2.17). Množenje se obavljaja u koracima (2.15) i (2.16). Za izračunavanje (2.15) potrebno je $d(n) - 2$ množenja za svako x_n iz $\mathbb{X}$ i za svaki čvor promenljive iz $\mathcal{N}$, što nam daje gornju granicu

$$\mathcal{O}\Big(\sum_{n \in \mathcal{N}} d(n) \cdot |\mathbb{X}|\Big). \tag{2.20}$$

Za izračunavanje (2.16) potrebno je $d(n) - 1$ množenja za sve moguće vrednosti za $x_{M \setminus n}$, za svako x_n, i za sve faktor-čvorove iz $\mathcal{M}$, što nam daje gornju granicu

$$\mathcal{O}\Big(\sum_{M \in \mathcal{M}} d(M) \cdot |\mathbb{X}|^{d(M)}\Big). \tag{2.21}$$

Sabiranjem prethodna dva izraza dobijamo asimptotski broj množenja potreban za rešavanje normalizacionog problema

$$\mathrm{T}^{\mathrm{Z}}_{\otimes} = \mathcal{O}\Big(\sum_{M \in \mathcal{M}} d(M) \cdot |\mathbb{X}|^{d(M)}\Big). \tag{2.22}$$

Sabiranja se obavljaju u koracima (2.16) i (2.17). Za izračunavanje (2.16), potrebno je $|\mathbb{X}|^{d(M)-1} - 1$ sabiranja za svako x_n i svaki faktor-čvor iz $\mathcal{M}$, odnosno

$$\mathcal{O}\Big(\sum_{M \in \mathcal{M}} |\mathbb{X}|^{d(M)}\Big) \tag{2.23}$$

sabiranja. U terminacionoj fazi imamo $\mathcal{O}\big(|\mathbb{X}|\big)$ sabiranja, pa je

$$\mathrm{T}^{\mathrm{Z}}_{\oplus} = \mathcal{O}\Big(\sum_{M \in \mathcal{M}} |\mathbb{X}|^{d(M)}\Big). \tag{2.24}$$

Totalna vremenska kompleksnost se dobija kao zbir $\mathrm{T}^{\mathrm{Z}}_{\mathbb{K}} = \mathrm{T}^{\mathrm{Z}}_{\oplus} + \mathrm{T}^{\mathrm{Z}}_{\otimes}$, pa je

$$\mathrm{T}^{\mathrm{Z}}_{\mathbb{K}} = \mathcal{O}\Big(\sum_{M \in \mathcal{M}} d(M) \cdot |\mathbb{X}|^{d(M)}\Big). \tag{2.25}$$

Memorijska kompleksnost je maksimalni memorijski prostor, koji mora biti alociran za reprezentaciju poruka, meren u broju elemenata u poluprstenu kojim su poruke predstavljene. Za jednu poruku potrebno je čuvanje $|\mathbb{X}|$ elemenata. Ukoliko bi sve poruke bile pamćene za vreme izvršenja algoritma, memorijska kompleksnost bi bila $\mathcal{O}(|\mathcal{E}| \cdot |\mathbb{X}|)$. Medjutim, kada čvor pošalje poruku, sve poruke od svih potomaka mogu biti izbrisane, a novoizračunata poruka može biti upisana na mesto jedne od izbrisanih. Pošto se algoritam inicijalizuje porukama iz listova, maksimalan broj poruka ne prelazi broj listova $|\mathcal{V}_l|$, pa se memorijska kompleksnost redukuje na

$$\mathrm{M}_{\mathbb{K}}^{\mathrm{mpa}} = \mathcal{O}(|\mathcal{V}_l| \cdot |\mathbb{X}|), \tag{2.26}$$

elemenata poluprstena $\mathbb{K}$.

Izračunavanje svih marginala

Vremenska kompleksnost: Za izračunavanje svih marginala, pored kolekcione faze koja zahteva

$$\mathcal{O}\Big(\sum_{M \in \mathcal{M}} d(M) \cdot |\mathbb{X}|^{d(M)}\Big) \tag{2.27}$$

operacija, potrebna je i distribuciona faza. Analiza vremenske kompleksnosti za distribucionu fazu može se izvršiti na sličan način kao i za kolekcionu. Najpre primetimo da dominantni član u izrazu (2.27) za kompleksnost kolekcionog algoritma potiče od izračunavanja poruke iz faktor-čvorova, pri čemu je za svaku od ovih poruka potrebno $\mathcal{O}\Big(d(M) \cdot |\mathbb{X}|^{d(M)}\Big)$ operacija. U distribucionoj fazi se iz svakog faktor-čvora šalje $d(M) - 1$ poruka, tako da je ukupna vremenska kompleksnost za izračunavanje marginala

$$\mathcal{O}\Big(\sum_{M \in \mathcal{M}} d(M)^2 \cdot |\mathbb{X}|^{d(M)}\Big). \tag{2.28}$$

Vremenska kompleksnost može biti smanjena primenom metode predložene u [1]. Neka je $\{a_1, \ldots, a_d\}$ skup od d realnih brojeva. Tada se proizvod svih članova a_i može naći sa $3(d-2)$ množenja, umesto sa očiglednih $d(d-2)$, uz pomoć dinamičkog izračunavanja pomoćnih promenljivih b_i i c_i, na osnovu

$$b_1 = a_1, \qquad b_i = b_{i-1} \cdot a_i; \quad i = 2, \cdots d, \tag{2.29}$$

$$c_d = b_d, \qquad c_i = c_{i+1} \cdot a_i; \quad i = 1, \cdots d-1, \tag{2.30}$$

što se može obaviti uz pomoć $2(d-2)$ množenja. Sada se proizvodi

$$\hat{a}_j = \prod_{\substack{i=1 \\ i \neq j}}^{d} a_i; \quad j = 1, \ldots, d \tag{2.31}$$

mogu izračunati pomoću

$$\hat{a}_1 = c_2, \qquad \hat{a}_d = b_{d-1}, \qquad \hat{a}_i = b_{i-1} c_{i+1}; \quad i = 2, \ldots, d-1, \tag{2.32}$$

sa dodatnih $d-2$ množenja, što daje ukupno $3(d-2)$ množenja.

Ova ideja se može primeniti i na izračunavanje svih poruka iz faktor-čvora ka njegovim potomcima

$$r_{M \to n}(x_n) = \bigoplus_{x_{M \setminus n}} f_M(x_M) \otimes \bigotimes_{n' \in ne(M) \setminus n} q_{n' \to M}(x_{n'}), \tag{2.33}$$

za svako $n \in ne(M)$. Za konfiguraciju $x_{M \setminus n}$, svi prozivodi

$$\bigotimes_{n' \in ne(M) \setminus n} q_{n' \to M}(x_{n'}); \quad n \in ne(M) \tag{2.34}$$

mogu se izračunati sa $3(d(M)-2)$ množenja. Svaki od ovih $d(M)$ proizvoda treba pomnožiti jednom sa $f_M(x_M)$, tako da kompleksnost izračunavanja svih proizvoda u (2.33), za sve konfiguracije iz $x_M \in \mathbb{X}^{d(M)}$, ostaje $\mathcal{O}\Big(d(M) \cdot |\mathbb{X}|^{d(M)}\Big)$ kao u kolekcionoj fazi. Broj sabiranja u jednom faktor-čvoru je u odnosu na kolekcionu fazu povećan sa $\mathcal{O}(|\mathbb{X}|^{d(M)})$ na $\mathcal{O}(d(M)|\mathbb{X}|^{d(M)})$, ali i dalje ne utiče na ukupnu kompleksnost, tako da kompleksnost algoritma za izračunavanje marginala u faktor-čvorovima $\mathrm{T}_{\mathbb{K}}^{Z_M}$ i čvorovima promenljivih ostaje ista kao i za izračunavanje normalizacione konstante i iznosi

$$\mathrm{T}_{\mathbb{K}}^{Z_M} = \mathrm{T}_{\mathbb{K}}^{Z_n} = \mathcal{O}\Big(\sum_{M \in \mathcal{M}} d(M) \cdot |\mathbb{X}|^{d(M)}\Big). \tag{2.35}$$

Memorijska kompleksnost: Da bi se izračunali svi marginali, sve poruke od potomaka ka roditeljima iz kolekcione faze moraju biti sačuvane. U distribucionoj fazi, poruke teku ka potomcima, i pošto čvor i primi poruku od svog roditelja j može se izračunati marginalna vrednost za cvor i, kao i vrednosti poruka koje i šalje potomcima. Tada nam poruke iz kolekcione faze više nisu potrebne i mogu biti izbrisane, a na njihovo mesto mogu se upisati poruke iz distribucione faze. Na ovaj način memorijsko zauzeće ne prelazi $|\mathcal{E}| \cdot |\mathbb{X}|$ elemenata poluprstena.

Ukoliko želimo da ubrzamo algoritam izračunavanjem svih proizvoda (2.34) uz pomoć veličina b_i i c_i kao što smo opisali u prethodnim paragrafima, potrebno je zapamtiti $2 \cdot d(M)$ vrednosti za svaku konfiguraciju $x_M \in \mathbb{X}^{d(M)}$, u svakom faktor-čvoru $M \in \mathcal{M}$. Dakle, potrebno je ukupno $2 \cdot d(M)\mathbb{X}^{d(M)}|\mathcal{M}|$ elemenata poluprstena, tako da ukupna memorijska kompleksnost algoritma za izračunavanje marginala u faktor-čvorovima $\mathrm{M}_{Z_M}^{\mathbb{K}}$ i čvorovima promenljivih $\mathrm{M}_{Z_n}^{\mathbb{K}}$ iznosi

$$\mathrm{M}_{\mathbb{K}}^{Z_M} = \mathrm{M}_{\mathbb{K}}^{Z_n} = \mathcal{O}\Big(|\mathcal{E}| \cdot |\mathbb{X}| + \sum_{M \in \mathcal{M}} d(M) \cdot |\mathbb{X}|^{d(M)}\Big) \tag{2.36}$$

elemenata poluprstena $\mathbb{K}$.

2.2.3 Primer algoritma

Posmatrajmo problem marginalizacije funkcije

$$Z_1(x_1) = \sum_{x_2, x_3, x_4, x_5} f(x_1, x_2, x_3, x_4, x_5),$$

gde je $x_i \in \mathbb{X}$, a funkcija se faktoriše kao

$$f(x_1, x_2, x_3, x_4, x_5) = f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_1, x_4) f_E(x_2, x_5). \tag{2.37}$$

Za *brute-force* izračunavanje izraza

$$Z_1(x_1) = \sum_{x_2, x_3, x_4, x_5} f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_1, x_4) f_E(x_2, x_5),$$

potrebno je $\mathcal{O}(|\mathbb{X}|^5)$ sabiranja i $\mathcal{O}(|\mathbb{X}|^5)$ množenja. Posle primene distributivnog zakona, ovaj izraz transformiše se u

$$Z_1(x_1) = \underbrace{f_A(x_1)}_{r_{A\to 1}(x_1)} \cdot \underbrace{\sum_{x_4} f_D(x_1, x_4)}_{r_{D\to 1}(x_1)} \cdot \underbrace{\sum_{x_2,x_3,x_5} f_B(x_2) f_C(x_1, x_2, x_3) f_E(x_2, x_5)}_{r_{C\to 1}(x_1)},$$

i može se izračunati sa $\mathcal{O}(|\mathbb{X}|^4)$ sabiranja i $\mathcal{O}(|\mathbb{X}|)$ množenja. Posle uvodjenja poruka $r_{A\to 1}(x_1)$, $r_{C\to 1}(x_1)$ i $r_{D\to 1}(x_1)$, kao na slici 2.5a, marginalna vrednost se može dobiti kao proizvod

$$Z_1(x_1) = r_{A\to 1}(x_1) \cdot r_{C\to 1}(x_1) \cdot r_{D\to 1}(x_1).$$

Slično, za *brute-force* izračunavanje poruke iz faktor-čvora

$$r_{C\to 1}(x_1) = \sum_{x_2,x_3,x_5} f_B(x_2) f_C(x_1, x_2, x_3) f_E(x_2, x_5),$$

potrebno je $\mathcal{O}(|\mathbb{X}|^4)$ sabiranja i $\mathcal{O}(|\mathbb{X}|^4)$ množenja, dok se posle primene distributivnog zakona ovaj broj redukuje na $\mathcal{O}(|\mathbb{X}|^3)$ sabiranja i $\mathcal{O}(|\mathbb{X}|^3)$ množenja

$$r_{C\to 1}(x_1) = \sum_{x_2,x_3} f_C(x_1, x_2, x_3) \cdot \underbrace{\sum_{x_5} f_B(x_2) \cdot f_E(x_2, x_5)}_{q_{2\to C}(x_2)} \cdot \underbrace{1(x_3)}_{q_{3\to C}(x_3)}.$$

Članovi $q_{2\to C}(x_2)$, $q_{3\to C}(x_3)$ mogu se shvatiti kao poruke iz čvorova 2 i 3 u čvor C, a poruka iz faktor-čvora C u čvor promenljive 1 može da se izračuna kao proizvod primljenih poruka sa faktor-čvorom f_C, koji je marginalizovan po svim promenljivama, osim po x_1

$$r_{C\to 1}(x_1) = \sum_{x_2,x_3} f_C(x_1, x_2, x_3) q_{2\to C}(x_2) \cdot q_{3\to C}(x_3).$$

Ponovo, za *brute-force* izračunavanje poruke iz čvora promenljive 2 u faktor-čvor C

$$q_{2\to C}(x_2) = \sum_{x_5} f_B(x_2) f_E(x_2, x_5),$$

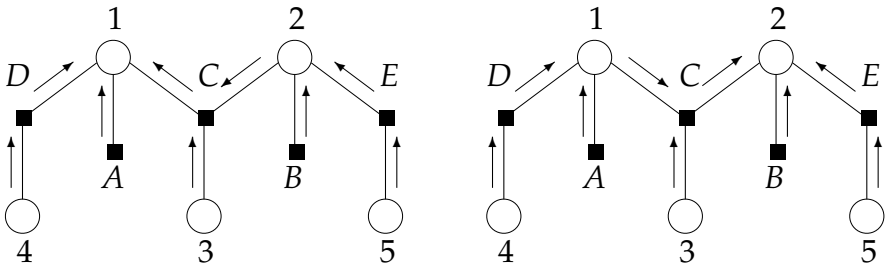
potrebno je $\mathcal{O}(|\mathbb{X}|^2)$ sabiranja i $\mathcal{O}(|\mathbb{X}|^2)$ množenja, što se može redukovati na $\mathcal{O}(|\mathbb{X}|^2)$ sabiranja i $\mathcal{O}(|\mathbb{X}|)$ množejna, posle korišćenja distributivnog zakona

$$q_{2\to C}(x_2) = \underbrace{f_B(x_2)}_{r_{B\to 2}(x_2)} \cdot \underbrace{\sum_{x_5} f_E(x_2, x_5)}_{r_{E\to 2}(x_2)}.$$

Poruke iz čvora promenljive mogu se izraziti kao proizvod poruka primljenih iz susednog faktor-čvora

$$q_{2\to C}(x_2) = r_{B\to 2}(x_2) \cdot r_{E\to 2}(x_2).$$

Na ovaj način problem efikasnog izračunavanja marginalne vrednosti može biti rešen slanjem poruka kroz faktor-graf i procesiranjem poruka u čvorovima, kao na slici 2.5a, po sledećem algoritmu.



(a) Z_1 (b) Z_2

Slika 2.5: Šeme izračunavanja marginala

Inicijalizacija:

$$q_{4\to D}(x_4) = q_{3\to C}(x_3) = q_{5\to E}(x_5) = 1,$$
$$r_{A\to 1}(x_1) = f_A(x_1), \quad r_{B\to 2}(x_2) = f_B(x_2).$$

Indukcija:

$$r_{D\to 1}(x_1) = \sum_{x_4} f_D(x_2, x_4) q_{4\to D}(x_4),$$
$$r_{E\to 2}(x_2) = \sum_{x_5} f_E(x_2, x_5) q_{5\to E}(x_5),$$
$$q_{2\to C}(x_2) = r_{B\to 2}(x_2) r_{E\to 2}(x_2),$$
$$r_{C\to 1}(x_1) = \sum_{x_2, x_3} f_C(x_1, x_2, x_3) \cdot q_{2\to C}(x_2) \cdot q_{3\to C}(x_3).$$

Terminacija:

$$Z_1(x_1) = q_{D\to 1}(x_D) \cdot q_{A\to 1}(x_A) \cdot q_{C\to 1}(x_C). \tag{2.38}$$

Slična propagaciona šema može se izvesti za izračunavanje marginala Z_2, sa jedinom razlikom u tome što se šalju poruke iz 1 u C, i iz C u 2, umesto poruka iz 2 u C, i iz C u 1, kao na slici 2.5b. Poruke mogu da se izračunaju kao

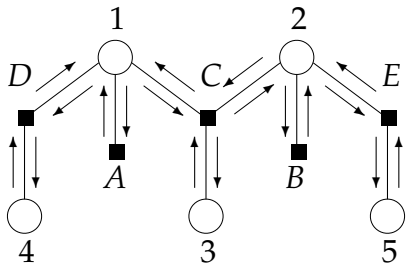
$$q_{1\to C}(x_1) = r_{A\to 1}(x_1) r_{D\to 1}(x_1)$$
$$r_{C\to 2}(x_2) = \sum_{x_2, x_3} f_C(x_1, x_2, x_3) \cdot q_{1\to C}(x_1) \cdot q_{3\to C}(x_3).$$

Konačno, pošto su sve poruke u faktor-grafu izračunate po propagacionoj šemi sa slike 2.6, problem izračunavanja marginala u svim čvorovima promenljivih

$$Z_n(x_n) = \sum_{x_{-n}} f(x_{1:T}), \;\; n = 1, \ldots, 5$$

može biti rešen pomoću

$$Z_1(x_1) = r_{C\to 1}(x_1) \cdot r_{A\to 1}(x_1) \cdot r_{C\to 1}(x_1),$$
$$Z_2(x_2) = r_{C\to 2}(x_2) \cdot r_{B\to 2}(x_2) \cdot r_{E\to 2}(x_2),$$
$$Z_3(x_3) = r_{D\to 3}(x_3) \quad Z_4(x_4) = r_{C\to 4}(x_4) \quad Z_5(x_5) = r_{E\to 5}(x_5),$$



Slika 2.6: Šema izračunavanja svih marginala.

a marginali u faktor-čvorovima

$$\widetilde{Z}_M(x_M) = \sum_{x_{-M}} f(x_{1:T}), \quad m = A, \ldots, D$$

mogu se izračunati kao

$$\begin{aligned}
\widetilde{Z}_A(x_1) &= q_{1\to A}(x_1) f_A(x_1) \\
\widetilde{Z}_B(x_2) &= q_{2\to B}(x_2) f_B(x_2) \\
\widetilde{Z}_D(x_1, x_4) &= f_D(x_1, x_4) \cdot r_{1\to D}(x_D) \cdot r_{4\to D}(x_D) \\
\widetilde{Z}_E(x_2, x_5) &= f_D(x_2, x_5) \cdot r_{2\to E}(x_E) \cdot r_{5\to E}(x_E) \\
\widetilde{Z}_C(x_1, x_2, x_3) &= f_C(x_1, x_2, x_3) \cdot q_{1\to C}(x_1) \cdot q_{2\to C}(x_2) \cdot q_{3\to C}(x_3).
\end{aligned}$$

2.2.4 *FB* algoritam nad komutativnim poluprstenom

Forward-backward (FB) algoritam nad komutativnim poluprstenom [83] je često korišćeni alat u zaključivanju.Pojavio se u dve nezavisne publikacije [5], [9], ali je poznatiji po radovima [3], [4]. Koristi dinamičko programiranje, ima vremensku kompleksnost $\mathcal{O}(N^2T)$ i memorijsku kompleksnost $\mathcal{O}(NT)$, gde je T dužina sekvence, a N broj stanja koja se klasifikuju. Uprkos vremenskoj efikasnosti, on postaje prostorno zahtevan kad je dužina sekvence velika [45].

Neka slučajna promenljiva $X_{0:T}$ uzima vrednosti iz skupa $\mathbb{X}^{T+1}$, i neka se funkcija $f : \mathbb{X}^{T+1} \to \mathbb{K}$ faktoriše u komutativnom poluprstenu $(\mathbb{K}, \oplus, \otimes, 0, 1)$ uz pomoć faktora $f_{A_0} : \mathbb{X} \to \mathbb{K}$ i $f_{A_t} : \mathbb{X}^2 \to \mathbb{K}$ kao

$$f(x_{0:T}) = f_{A_0}(x_0) \otimes \bigotimes_{t=1}^{T} f_{A_t}(x_{t-1}, x_t). \tag{2.39}$$

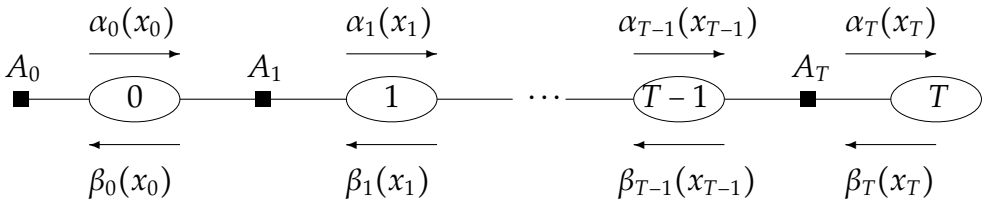
U ovom slučaju, faktor-graf ima strukturu lanca kao na slici (2.7), a marginalizacioni i normalizacioni problemi svode se na:

Marginalizacioni problem: Izračunati sumu

$$Z_{a:b}(x_{a:b}) = \bigoplus_{x_{-a:b}} u(x_{0:T}). \tag{2.40}$$

Normalizacioni problem: Izračunati sumu

$$Z = \bigoplus_{x_{0:T}} u(x_{0:T}). \tag{2.41}$$



Slika 2.7: Faktor-graf i poruke u slučaju lanca $f_{A_0}(x_0) \otimes \bigotimes_{t=1}^{T} f_{A_t}(x_{t-1}, x_t)$.

Rešavanje ova dva problema primenom *brute force* algoritma podrazumeva enumerisanje svih sekvenci $x_{0:T} \in \mathbb{X}^{T+1}$, za šta potrebno je $T \cdot |\mathbb{X}|^{T+1}$ sabiranja i množenja. Ovo postaje praktično neizvodljivo, čak i u slučaju malih vrednosti za $|\mathbb{X}|$ i T (za $|\mathbb{X}| = 10$ i $T = 20$, ukupan broj operacija je reda veličine 10^{22}). U daljem tekstu pokazujemo kako se ovi problemi mogu rešiti primenom *MP* algoritma uz pomoć $\mathcal{O}(T \cdot |\mathbb{X}|^2)$ operacija.

Kao što smo smo objasnili u odeljku (2.2.1), *MP* algoritam sastoji se iz dve faze: kolekciona faza, u kojoj se poruke šalju od listova ka unapred definisanom korenu, i distribuciona faza, u kojoj se poruke šalju od korena ka listovima.

U slučaju lanca, za koren stabla biramo čvor T, tako da se poruke u kolekcionoj fazi šalju od čvora A_0 ka čvoru T, a u distribucionoj fazi od čvora T ka čvoru A_0. Poruka od čvora ka njegovom susedu izračunava se na osnovu poruke primljene od drugog suseda. S obzirom na to da se poruka iz čvora promenljive dobija kao proizvod pristiglih poruka, poruka koju čvor promenljive šalje, ista je kao i poruka koju je primio, tako da u ovom slučaju možemo govoriti o slanju poruka izmedju faktor-čvorova sa zajedničkim susedom.

Kolekciona faza algoritma naziva se *forward algoritam*, a poruke $r_{A_t \to t} = q_{t \to A_{t+1}}$ *forward poruke* i označavaju se sa α_t, kao na slici 2.7. Na osnovu izlaganja iz odeljka 2.2.1 *forward* poruke imaju vrednost

$$\alpha_i(x_i) = \bigoplus_{x_{0:i-1}} f_{A_0}(x_0) \otimes \bigotimes_{t=1}^{i} f_{A_t}(x_{t-1}, x_t), \tag{2.42}$$

inicijalizuju se na

$$\alpha_0(x_0) = f_{A_0}(x_0), \tag{2.43}$$

i rekurzivno izračunavaju pomoću

$$\alpha_i(x_i) = \bigoplus_{x_{i-1}} f_{A_{i-1}}(x_{i-1}, x_i) \otimes \alpha_{i-1}(x_{i-1}). \tag{2.44}$$

Slično, distribuciona faza se naziva *backward algoritam*, a poruke $r_{A_{t+1} \to t} = q_{t \to A_t}$ nazivaju se *backward poruke* i označavaju se sa β_t. *Backward* poruke imaju vrednost

$$\beta_i(x_i) = \bigoplus_{x_{i+1:T}} \bigotimes_{t=i+1}^{T} f_{A_t}(x_{t-1}, x_t), \tag{2.45}$$

i inicijalizuju se na

$$\beta_T(x_T) = 1, \tag{2.46}$$

i rekurzivno se izračunavaju pomoću

$$\beta_t(x_t) = \bigoplus_{x_{t+1}} f_{A_{t+1}}(x_t, x_{t+1}) \otimes \beta_{t+1}(x_{t+1}). \tag{2.47}$$

Za rešavanje normalizacionog problema dovoljan je *forward* algoriam, po čijem okončanju normalizacionu konstantu nalazimo pomoću

$$\bigoplus_{x_{0:T}} f(x_{0:T}) = \bigoplus_{x_T} \alpha_T(x_T). \tag{2.48}$$

Za rešenje marginalizacionog problema, potrebno je izvršiti i *backward* algoritam, a marginali se mogu izračunati kao

$$Z_{l:r}(x_{l:r}) = \bigoplus_{x_{-l:r}} u(x_{0:T}) = \alpha_l(x_l) \otimes \bigotimes_{i=l+1}^{r} f_{A_i}(x_{i-1}, x_i) \otimes \beta_r(x_r). \tag{2.49}$$

Vremenska kompleksnost *forward* algoritma iznosi $\mathcal{O}(T \cdot |\mathbb{X}|^2)$, s obzirom na to da se u svakom od T koraka izvršava $\mathcal{O}(|\mathbb{X}|^2)$ množenja i sabiranja u poluprstenu. Slično važi i za *backward* algoritam, tako da su vremenske kompleksnosti za *FA* i *FBA*, iste i iznose $\mathcal{O}(T \cdot |\mathbb{X}|^2)$ operacija u poluprstenu.

Memorijska kompleksnost za *FA* iznosi $\mathcal{O}(1)$, s obzirom na to da *forward* poruka α_t može biti izbrisana po izračunavanju poruke α_{t+1}. U slučaju *FBA*, *forward* poruke moraju biti sačuvane tokom izvršavanja algoritma, tako da je memorijska kompleksnost za *FA* $\mathcal{O}(T \cdot |\mathbb{X}|)$ elemenata poluprstena.

Smanjenje memorijske kompleksnosti moguće je postići modifikacijom *FB* algoritma, nazvanom *checkpointing algoritam* [27], [80]. *Checkpointing* algoritam deli ulazni niz u $\sqrt{T}$ podsekvenci i tokom *forward* algoritma čuva samo prvu *forward* poruku u svakoj podsekvenci (kontrolne poruke). U *backward* algoritmu, *forward* poruke se za svaku podsekvencu ponovo redom izračunavaju, počevši od kontrolne poruke za tu podsekvencu. Na taj način kompleksnost izračunavanja *forward* poruka, potrebnih za izvršavanje *FBA*, povecana je na $\mathcal{O}(2T - N^2\sqrt{T})$, što dovodi do veće ukupne vremenske kompleksnosti. S druge strane, memorijska kompleksnost, iako svedena na $\mathcal{O}(N\sqrt{T})$, i dalje zavisi od dužine sekvence.

Druga mogućnost je upotreba *forward-only algoritma* [11], [63], [78], za koji matrice mogu biti izračunavane u *runtime*-u. Algoritam se izvršava uz konstantu memorijsku kompleksnost, ali je neefikasan sa stanovišta vremenske kompleksnosti, koja iznosi $\mathcal{O}(N^4T)$.

Za specijalni tip lanca, skriveni Markovljev model, koji razmatramo u glavi 3, moguće je primeniti algoritam koji su razvili *Khreich* i saradnici [45]. U njihovom radu predložen je algoritam za izračunavanje marginala *forward filtering backward smoothing (EFFBS)*, koji se izvršava uz vremensku kompleksnost $\mathcal{O}(N^2T)$, i sa memorijom nezavisnom od dužine sekvence $\mathcal{O}(N)$. Medjutim, algoritam zahteva da matrica f_t, koja odgovara faktorima, ne zavisi od t, što je slučaj kod skrivenog Markovljevog modela, ali postavlja i dodatni uslov - da ova matrica bude regularna, što u opštem slučaju ne važi.

2.3 Izračunavanje matematičkog očekivanja vektorske slučajne promenljive

U ovom poglavlju razmatramo izračunavanje matematičkog očekivanja višedimenzione slučajne promenljive primenom *MP* algoritma. Neka je $X_{1:T}$ višedimenziona slučajna promenljiva sa raspodelom $p_{X_{1:T}}$, i neka je $\boldsymbol{g} : \mathbb{X}^T \to \mathbb{R}^d$ funkcija promenljive $X_{1:T}$. Neka se još

$p_{X_{1:T}}$ i $\boldsymbol{g}$ mogu predstaviti faktor-grafom bez ciklusa kao proizvod, odnosno kao zbir faktora $\phi_M : \mathbb{X}^{d(M)} \to \mathbb{R}$ i $\boldsymbol{g}_M : \mathbb{R}^{d(M)} \to \mathbb{R}^d$ kao

$$p_{X_{1:T}}(x_{1:T}) = \prod_{M \in \mathcal{M}} \phi_M(x_M), \quad \boldsymbol{g}(x_{1:T}) = \sum_{M \in \mathcal{M}} \boldsymbol{g}_M(x_M). \tag{2.50}$$

U ovom slučaju matematičko očekivanje funkcije $\boldsymbol{g}$, u odnosu na $p_{X_{1:T}}$, ima oblik

$$\mathbb{E}[\boldsymbol{g}] = \sum_{x_{1:T}} p(x_{1:T}) \cdot \boldsymbol{g}(x_{1:T}) = \sum_{x_{1:T}} \prod_{M \in \mathcal{M}} \phi_M(x_M) \cdot \sum_{M \in \mathcal{M}} \boldsymbol{g}_M(x_M). \tag{2.51}$$

U narednom odeljku pokazujemo kako se očekivanje može izračunati primenom *MP* algoritma nad *sum-product* poluprstenom, poznatijim kao *sum-product algoritam* (*SPA*). *SPA* se koristi za nalaženje svih marginala u faktor-čvorovima faktor-grafa za $p_{X_{1:T}}$, koji se kasnije koriste za izračunavanje očekivanja. Kao što smo pomenuli, nalaženje svih marginala izvršava se uz memorijsku kompleksnost proporcionalnu broju faktor-čvorova u grafu, što u slučaju velikih grafova predstavlja nedostatak. Za prevazilaženje ovog problema, u odeljku 2.3.2 izvodimo algoritam za izračunavanje normalizacione konstante, koji funkcioniše kao *MP* algoritam nad poluprstenom očekivanja (*EMP* algoritam). Pokazujemo da *EMP* ima istu vremensku kompleksnost izračunavnja očekivanja kao i *SPA*. S druge strane, memorijska kompleksnost *EMP*-a je proporcionalna broju listova u faktor-grafu, za razliku od *SPA*, kod koga je, kao što smo pomenuli, proporcionalna broju faktor-čvorova u grafu.

2.3.1 Izračunavanje matematičkog očekivanja primenom *MP* algoritma nad *sum-product* poluprstenom

Da bismo preveli problem izračunavanja očekivanja u problem nalaženja svih marginala u faktor-čvorovima, izvršimo najpre transformaciju izraza (2.51)

$$\mathbb{E}[\boldsymbol{g}] = \sum_{x_{1:T}} \prod_{M \in \mathcal{M}} \phi_M(x_M) \cdot \sum_{k \in \mathcal{M}} \boldsymbol{g}_K(x_K) = \sum_{k \in \mathcal{M}} \sum_{x_M} \Big(\sum_{x_{-M}} \prod_{M \in \mathcal{M}} \phi_M(x_M) \Big) \cdot \boldsymbol{g}_K(x_K). \tag{2.52}$$

Sada, pošto su marginalne vrednosti u faktor-čvorovima date sa

$$\widetilde{Z}_K(x_K) = \sum_{x_{-M}} \prod_{M \in \mathcal{M}} \phi_M(x_M),$$

dobijamo

$$\mathbb{E}[\boldsymbol{g}] = \sum_{M \in \mathcal{M}} \sum_{x_M} \widetilde{Z}_M(x_M) \cdot \boldsymbol{g}_M(x_M). \tag{2.53}$$

Maginalne vrednosti $\widetilde{Z}_M(x_M)$ mogu se izračunati primenom *SPA*, a zatim se izračunava vrednost očekivanja (2.53).

Vremenska kompleksnost ovog algoritma $\mathrm{T}_{\mathbb{R}}^{\mathbb{E}[g]}$, merena u odnosu na realne operacije, predstvlja zbir vremenske kompleksnosti algoritma za izračunavanje svih marginala (2.35), $\mathcal{O}\big(\sum_{M \in \mathcal{M}} d(M) \cdot |\mathbb{X}|^{d(M)}\big)$ i kompleksnosti izračunavanja izraza (2.53) $\mathcal{O}\big(\sum_{M \in \mathcal{M}} d(M) \cdot |\mathbb{X}|^{d(M)} \cdot d\big)$

$$\mathrm{T}_{\mathbb{R}}^{\mathbb{E}[g]} = \mathcal{O}\Big(\sum_{M \in \mathcal{M}} d(M) \cdot |\mathbb{X}|^{d(M)} \cdot d \Big). \tag{2.54}$$

Memorijska kompleksnost ovog algoritma, merena u broju realnih brojeva potrebnom za skladištenje medjurezultata, jednaka je memorijskoj kompleksnosti algoritma za

izračunavanje svih marginala, pošto se izračunavanje (2.53) može izvršiti u fiksnom memorijskom prostoru, tako da je saglasno izrazu (2.36)

$$\mathrm{M}_{\mathbb{R}}^{\mathbb{E}[g]} = \mathcal{O}\Big(|\mathcal{E}| \cdot |\mathbb{X}| + \sum_{M \in \mathcal{M}} d(M) \cdot \mathbb{X}^{d(M)}\Big). \tag{2.55}$$

2.3.2 Izračunavanje matematičkog očekivanja primenom *EMP* algoritma

Poluprsten očekivanja

Neka je $\mathbb{R}^d$ vektorski prostor dimenzije $d \in \mathbb{N}$ nad poljem realnih brojeva $\mathbb{R}$.

Definicija 2 *[22], [23] Poluprsten očekivanja reda d definisan je kao petorka*

$$\big(\ \mathbb{R} \times \mathbb{R}^d,\ \oplus,\ \otimes,\ (0,\ \mathbf{0}), (1,\ \mathbf{0})\ \big), \tag{2.56}$$

pri čemu je $\mathbf{0}$ *nula vektor u* $\mathbb{R}^d$*, a operacije* $\oplus$ *i* $\otimes$ *su definisane sa*

$$(p_1, \boldsymbol{h}_1) \oplus (p_2, \boldsymbol{h}_2) = (p_1 + p_2, \boldsymbol{h}_1 + \boldsymbol{h}_2), \tag{2.57}$$
$$(p_1, \boldsymbol{h}_1) \otimes (p_2, \boldsymbol{h}_2) = (p_1 p_2, p_1 \boldsymbol{h}_2 + p_2 \boldsymbol{h}_1), \tag{2.58}$$

za sve parove $(p_1, \boldsymbol{h}_1), (p_2, \boldsymbol{h}_2) \in \mathbb{R} \times \mathbb{R}^d$.

Prva komponenta uredjenog para naziva se p-komponenta i označava se sa $w^{(p)}$, a druga se naziva h-komponenta i označava se sa $w^{(h)}$. Poluprsten očekivanja reda 1 naziva se *entropijski poluprsten*. Važi sledeća lema.

Lema 2.3.1 *Neka je* $\mathcal{M}$ *konačan skup indeksa i* $(p_m,\ \boldsymbol{h}_m) \in \mathbb{R} \times \mathbb{R}^d$, $m \in \mathcal{M}$. *Tada važe sledeće jednakosti*

$$\bigotimes_{m \in \mathcal{M}} (p_m, \boldsymbol{h}_m) = \Big(\sum_{m \in \mathcal{M}} p_m,\ \sum_{m \in \mathcal{M}} \boldsymbol{h}_m\Big), \tag{2.59}$$
$$\bigotimes_{m \in \mathcal{M}} (p_m, \boldsymbol{h}_m) = \Big(\prod_{m \in \mathcal{M}} p_m,\ \sum_{m \in \mathcal{M}} \prod_{j \in \mathcal{M} \setminus m} p_j \cdot \boldsymbol{h}_m\Big). \tag{2.60}$$

Dokaz. Lemu dokazujemo matematičkom indukcijom po kardinalnosti skupa $\mathcal{M}$. Jednakost (2.59) očigledno sledi iz definicije. U daljem tekstu dokazujemo jednakost (2.60). Bez gubitka opštosti možemo pretpostaviti da je $\mathcal{M} = \{1,\ 2,\ \ldots,\ k\}$, gde je $k \in \mathbb{N}$. Ako je $k = 2$, jednakost (2.60) se svodi na definiciju množenja u poluprstenu očekivanja

$$(a_1,\ b_1) \otimes (a_2,\ b_2) = (a_1 a_2,\ a_1 b_2 + a_2 b_1).$$

Neka sada jednakost (2.60) važi za k-elementni skup $\mathcal{M}_k = \{1,\ 2,\ \ldots,\ k\}$,

$$\bigotimes_{m \in \mathcal{M}_k} (a_m,\ b_m) = \left(\prod_{m \in \mathcal{M}_k} a_m,\ \sum_{m \in \mathcal{M}_k} b_m \prod_{j \in \mathcal{M}_k \setminus m} a_j\right).$$

Korišćenjem jednakosti za slučaj k, i korišćenjem $\mathcal{M}_{k+1} = \mathcal{M}_k \cup \{k+1\}$, lako se dokazuje da jednakost (2.60) važi i za $k+1$-elementni skup $\mathcal{M}_{k+1} = \{1,\ 2,\ \ldots,\ k+1\}$,

$$\bigotimes_{m \in \mathcal{M}_{k+1}} (a_m,\ b_m) = \bigotimes_{m \in \mathcal{M}_k} (a_m,\ b_m) \otimes (a_{m+1},\ b_{m+1}) =$$

$$= \left(\prod_{m \in \mathcal{M}_{k+1}} a_m,\ b_{k+1} \prod_{m \in \mathcal{M}_k} a_m + \sum_{m \in \mathcal{M}_k} b_m \prod_{j \in \mathcal{M}_{k+1} \setminus m} a_j \right)$$

$$= \left(\prod_{m \in \mathcal{M}_{k+1}} a_m,\ \sum_{m \in \mathcal{M}_{k+1}} b_m \prod_{j \in \mathcal{M}_{k+1} \setminus m} a_j \right),$$

čime je lema dokazana.

EMP algoritam

Neka je, kao na početku ovog poglavlja, $X_{1:T}$ višedimenziona slučajna promenljiva sa raspodelom $p_{X_{1:T}}$, i neka je $\boldsymbol{g} : \mathbb{X}^T \to \mathbb{R}^d$ funkcija promenljive $X_{1:T}$. Neka se $p_{X_{1:T}}$ i $\boldsymbol{g}$ mogu predstaviti faktor-grafom bez ciklusa kao proizvod, odnosno zbir faktora $\phi_M : \mathbb{X}^{d(M)} \to \mathbb{R}$ i $\boldsymbol{g}_M : \mathbb{R}^{d(M)} \to \mathbb{R}^d$, kao

$$p_{X_{1:T}}(x_{1:T}) = \prod_{M \in \mathcal{M}} \phi_M(x_M), \quad \boldsymbol{g}(x_{1:T}) = \sum_{M \in \mathcal{M}} \boldsymbol{g}_M(x_M). \tag{2.61}$$

Dalje, neka je $(\mathbb{K}, \oplus, \otimes, (0, \mathbf{0}), (1, \mathbf{0}))$ poluprsten očekivanja reda d, i neka se funkcija $w : \mathbb{X}^M \to \mathbb{K}$ faktoriše kao

$$w(x_{1:T}) = \bigoplus_{M \in \mathcal{M}} w_M(x_M), \tag{2.62}$$

gde su faktori $w_M : \mathbb{X}^{d(M)} \to \mathbb{R} \times \mathbb{R}^d$ dati sa

$$w_M(x_M) = \big(\phi_M(x_M), \phi_M(x_M)\boldsymbol{g}_M(x_M)\big), \tag{2.63}$$

za $x_M \in \mathbb{X}^{d(M)}$. Koristeći se lemom o množenju u poluprstenu očekivanja (2.60), lako se dobija

$$w(x_{1:T}) = \bigotimes_{M \in \mathcal{M}} w_M(x_M) = \Big(\prod_{M \in \mathcal{M}} \phi_M(x_M),\ \prod_{M \in \mathcal{M}} \phi_M(x_M) \sum_{K \in \mathcal{M}} \boldsymbol{g}_K(x_K) \Big), \tag{2.64}$$

odnosno

$$\bigoplus_{x_{1:T}} w(x_{1:T}) = \bigoplus_{x_{1:T}} \bigotimes_{M \in \mathcal{M}} w_M(x_M) = \big(Z,\ \mathbb{E}[\boldsymbol{g}] \big). \tag{2.65}$$

Na ovaj način, uz pomoć *MP* algoritma nad poluprstenom očekivanja, u jedinstvenom prolazu, dobijamo normalizacionu konstantu u *sum-product* poluprstenu i očekivanje funkcije $\boldsymbol{g}$. Primetimo da, s obzirom na to da je $p_{X_{1:T}}$ raspodela, normalizaciona konstanta ima vrednost $Z = \sum_{x_{1:T}} p_{X_{1:T}}(x_{1:T}) = 1$.

Originalna verzija ovog algoritma razvijena je u [38] za skalarne slučajne promenljive. Algoritam iz [38] funkcioniše kao *MP* algoritam nad entropijskim poluprstenom, koji je, kao što smo pomenuli, poluprsten očekivanja reda 1 i naziva se algoritam slanja entropijskih poruka (*entropy message passing, EMP*). U daljem tekstu ovaj naziv koristićemo i za *MP* algoritam nad poluprstenom očekivanja. *EMP* algoritam sledi.

Inicijalizacija: Izabrati proizvoljan list za koren stabla. Postaviti poruke iz čvorova promenljivih i faktor-čvorova u ostalim listovima na

$$q_{n \to M}(x_n) = (1, \mathbf{0}), \tag{2.66}$$

$$r_{M \to n}(x_n) = \big(\phi_M(x_n), \phi_M(x_n)\boldsymbol{g}_M(x_n)\big). \tag{2.67}$$

Indukcija: Kada čvor primi poruke od svih potomaka, šalje poruku roditelju, saglasno sledećim formulama

$$q_{n\to M}(x_n) = \bigotimes_{M'\in ne(n)\setminus M} r_{M'\to n}(x_n), \tag{2.68}$$

$$r_{M\to n}(x_n) = \bigoplus_{x_{M\setminus n}} \big(\phi_M(x_M), \phi_M(x_M)\boldsymbol{g}_M(x_M)\big) \bigoplus_{n'\in ne(M)\setminus n} q_{n'\to M}(x_{n'}). \tag{2.69}$$

Terminacija: Proces se završava u listu n, koji je izabran za koren stabla po prijemu poruke iz jedinog susednog čvora M. Ovde je

$$\big(\ Z_p,\ \mathbb{E}_p(\boldsymbol{g})\ \big) = \bigoplus_{x_n} r_{M\to n}(x_n). \tag{2.70}$$

Vremenska kompleksnost: U odeljku 2.2.2 razmatrali smo vremensku kompleksnost *MP* algoritma, $\mathrm{T}^{\mathrm{mpa}}$ kao asimptotski broj operacija u poluprstenu, $\mathrm{T}^{\mathrm{mpa}}_{\oplus} + \mathrm{T}^{\mathrm{mpa}}_{\otimes}$, gde su $\mathrm{T}^{\mathrm{mpa}}_{\oplus}$ i $\mathrm{T}^{\mathrm{mpa}}_{\otimes}$ vremenske kompleksnosti, u odnosu na operacije u poluprstenu, definisane kao asimptotski broj sabiranja i oduzimanja, potrebnih za izvršavanje algoritma kada $|\mathcal{N}|$, $|\mathcal{M}|$ i $|\mathbb{X}|$ teže beskonačnosti. Pokazano je da važi

$$\mathrm{T}^{\mathrm{mpa}}_{\oplus} = \mathcal{O}\Big(\sum_{M\in\mathcal{M}} |\mathbb{X}|^{d(M)}\Big), \tag{2.71}$$

$$\mathrm{T}^{\mathrm{mpa}}_{\otimes} = \mathcal{O}\Big(\sum_{M\in\mathcal{M}} d(M)\cdot|\mathbb{X}|^{d(M)}\Big). \tag{2.72}$$

U ovom odeljku razmatramo vremensku kompleksnost kao asimptotski broj realnih operacija (sabiranja, oduzimanja i množenja), potrebnih za izvršenje algoritma, $\mathrm{T}^{\mathrm{mpa}}$ sa dodatnim zahtevom da $|A_{\boldsymbol{\nu}}|$ teži beskonačnosti. Vremenska kompleksnost data je sa

$$\mathrm{T}^{\mathrm{mpa}} = \mathrm{T}^{\mathrm{mpa}}_{+} + \mathrm{T}^{\mathrm{mpa}}_{\times}, \tag{2.73}$$

gde su $\mathrm{T}^{\mathrm{sr}}_{+}$ i $\mathrm{T}^{\mathrm{sr}}_{\times}$ asimptotski brojevi realnih sabiranja i množenja, koji su dati sa

$$\mathrm{T}^{\mathrm{mpa}}_{+} = \mathrm{T}^{\mathrm{mpa}}_{\oplus}\cdot\mathrm{T}^{\oplus}_{+} + \mathrm{T}^{\mathrm{mpa}}_{\otimes}\cdot\mathrm{T}^{\otimes}_{+}, \tag{2.74}$$

$$\mathrm{T}^{\mathrm{mpa}}_{\times} = \mathrm{T}^{\mathrm{mpa}}_{\oplus}\cdot\mathrm{T}^{\oplus}_{\times} + \mathrm{T}^{\mathrm{mpa}}_{\otimes}\cdot\mathrm{T}^{\otimes}_{\times}, \tag{2.75}$$

gde su $\mathrm{T}^{\oplus}_{+}$ i $\mathrm{T}^{\otimes}_{+}$ brojevi realnih sabiranja, a $\mathrm{T}^{\oplus}_{\times}$ i $\mathrm{T}^{\otimes}_{\times}$ brojevi realnih množenja, potrebnih za izvršenje odgovarajuće operacije u poluprstenu.

Za sabiranje u poluprstenu očekivanja potrebno je $d+1$ realnih sabiranja i nisu potrebna množenja, pa je $\mathrm{T}^{\oplus}_{+} = \mathcal{O}(d)$ i $\mathrm{T}^{\oplus}_{\times} = 0$, dok je za množenje u poluprstenu očekivanja potrebno d sabiranja i $2d+1$ množenja, tako da se za vremensku kompleksnost operacija u *EMP* algoritmu dobija

$$\mathrm{T}^{\mathrm{emp}}_{+} = \mathcal{O}\Big(\sum_{M\in\mathcal{M}} |\mathbb{X}|^{d(M)}\cdot K\Big), \tag{2.76}$$

$$\mathrm{T}^{\mathrm{emp}}_{\times} = \mathcal{O}\Big(\sum_{M\in\mathcal{M}} d(M)\cdot|\mathbb{X}|^{d(M)}\cdot d\Big), \tag{2.77}$$

pa je totalna vremenska kompleksnost ista kao i kod *SPA*

$$\mathrm{T}^{\mathrm{emp}} = \mathcal{O}\Big(\sum_{M\in\mathcal{M}} d(M)\cdot|\mathbb{X}|^{d(M)}\cdot d\Big). \tag{2.78}$$

Memorijska kompleksnost: Memorijska kompleksnost data je maksimalnim brojem realnih brojeva, potrebnih za izvršenje algoritma, kada $|\mathcal{V}_l|$ i $|A_v|$ teže beskonačnosti, i može se izračunati kao

$$\mathrm{M}_{\mathbb{R}}^{\mathrm{mpa}} = \mathrm{M}_{\mathrm{gsr}}^{\mathrm{mpa}} \cdot \mathrm{M}_{\mathbb{R}}^{\mathrm{gsr}} \tag{2.79}$$

gde je $\mathrm{M}_{\mathbb{K}}^{\mathrm{mpa}}$ memorija potrebna za izvršenje algoritma, kada je osnovna memorijska jedinica element poluprstena, koja je po formuli (2.26) jednaka $\mathcal{O}(|\mathcal{V}_l| \cdot |\mathbb{X}|)$. $\mathrm{M}_{\mathbb{R}}^{\mathrm{gsr}}$ je broj realnih brojeva potrebnih za predstavljanje jednog elementa poluprstena, koji je u slučaju poluprstena očekivanja jednak $d + 1$. Na osnovu ovoga, memorijska kompleksnost algoritma je

$$\mathrm{M}_{\mathbb{R}}^{\mathrm{mpa}} = \mathcal{O}(|\mathcal{V}_l| \cdot |\mathbb{X}| \cdot d). \tag{2.80}$$

Dakle, za razliku od *SPA*, koji zahteva memoriju proporcionalnu broju čvorova u stablu, memorijska kompleksnost *EMP*-a je proporcionalna broju listova. Ova razlika naročito dolazi do izražaja u slučaju grafova sa strukturom lanca, koji imaju samo dva lista, dok ukupan broj njihovih čvorova može biti veliki. Ovaj slučaj će detaljno biti razmatran u glavama 3 i 4.

Glava 3

Izračunavanje entropije skrivenog Markovljevog modela

Skriveni Markovljev model (*hidden Markov model, HMM*) [8], [24] je probabilistički koji, na jednostavan način modeluje sekvence parova stanje-opservacija. Estimacija parametara *HMM*-a se najčešće obavlja kombinacijom *FB* algoritma i *EM* (*expectation-maximization*) algoritma [19]. Ova kombinacija je poznata kao *BaumWelch* (*BW*) algoritam [4], [5]. *FB* algoritam je i poznat po radu koji su objavili *Baum* i *Welch* i kome je uveden kao algoritam koji se primenjuje za *HMM*, a kasnije je generalisan kao algoritam nad komutativnim poluprstenom, kao što je objašnjeno u odeljku 2.2.4.

Prilikom procesiranja sekvenci, dva glavna problema kod *HMM*-a su odredjivanje verovatnoće sekvence i odredjivanje verovatnoće stanja u odredjenom trenutku u sekvenci. Prvi od ova dva problema rešava se primenom čuvenog Viterbijevog algoritma, a *FB* algoritam se koristi za drugi problem. Ova dva algoritma daju mogućnost za efikasno dekodovanje sekvenci. Ukoliko se koriste dugačke sekvence, naročito je važno voditi računa kako o numeričkoj stabilnosti, tako i o memorijskim zahtevima za izvršenje algoritma. Ovakav slučaj se javlja u bezbedosti računara [52], [85], bioinformatici [48], [62] i robotici [47].

Entropija sekvence stanja za zadatu sekvencu opservacija *HMM*-a predstavlja meru njegove neizvesnosti i može biti korišćena kao pokazatelj *HMM* performansi [31]. U ovoj glavi izveden je numerički stabilan algoritam za izračunavanje entropije *HMM*-a [39], koji je primenjiv kako za kratke, tako i za dugačke sekvence. U poglavlju 3.1 data je rekapitulacija *FB* algoritma nad komutativnim poluprstenom. U poglavlju 3.2 dajemo definiciju *HMM*-a i pokazujemo kako se *FB* algoritam može primeniti na njega. U poglavlju 3.3 razmatramo problem izračunavanja entropije skrivenog Markovljevog modela. Najpre dajemo pregled dva algoritma koji se mogu primeniti u ovu svrhu: algoritam koji su dali *Mann* i *McCallum* [60] i algoritam koji su dali *Hernando* i saradnici [20], [31]. Zatim izvodimo novi algoritam za izračunavanje entropije, koji funkcioniše kao *forward* algoritam nad entropijskim poluprstenom i može se smatrati numerički stabilnom verzijom *EMP* algoritma iz odeljka 2.3.2 primenjenog na *HMM*. Za razliku od algoritma koji su dali *Mann* i *McCallum* [60], *EMP* ne izračunava svaki marginal posebno, već izračunava gradijent u jedinstvenom *forward* prolazu koristeći se dvostrukom rekurzijom. Zahvaljujući ovome, *EMP* ima znatno manju memorijsku kompleksnost nego algoritam koji su dali *Mann* i *McCallum*. Takodje, *EMP* može biti transformisan u algoritam koji su dali *Hernando* i saradnici [31], pri čemu, za razliku od njega, može da izračuna i podsekvencom ograničenu entropiju. Izračunavanje

podsekvencom ograničene entropije razmatramo u poglavlju 3.4.

Pomenimo još, da iako se diskusija iz ove glave odnosi na tradicionalni *HMM* [8], [24], [72], ona lako može biti uopštena i za kompleksnije modele, kao što su takozvani *pairwise* i *triplet* Markovljevi lanci [70], [71].

3.1 *FB* algoritam- rekapitulacija

Preglednosti radi, ponovo dajemo *FB* algoritam opisan u odeljku 2.2.4. Dakle, neka promenljiva $x_{0:T} = (x_0, \cdots, x_T)$ uzima vrednosti iz skupa $\mathbb{X}^{T+1}$, i neka se funkcija $u : \mathbb{X}^T \to \mathbb{K}$ faktoriše u komutativnom poluprstenu $(\mathbb{K}, \oplus, \otimes, 0, 1)$ uz pomoć faktora $u_{A_0} : \mathbb{X} \to \mathbb{K}$ i $u_{A_t} : \mathbb{X}^2 \to \mathbb{K}$ kao

$$u(x_{0:T}) = u_{A_0}(x_0) \otimes \bigotimes_{t=1}^{T} u_{A_t}(x_{t-1}, x_t). \tag{3.1}$$

FB algoritam se koristi za rešavanje sledeća dva problema:

Marginalizacioni problem: Izračunati sumu

$$Z_{a:b}(x_{a:b}) = \bigoplus_{x_{-a:b}} u(x_{0:T}). \tag{3.2}$$

Normalizacioni problem: Izračunati sumu

$$Z = \bigoplus_{x_{0:T}} u(x_{0:T}). \tag{3.3}$$

Forward poruke definisane su sa

$$\alpha_i(x_i) = \bigoplus_{x_{0:i-1}} f_{A_0}(x_0) \otimes \bigotimes_{t=1}^{i} f_{A_t}(x_{t-1}, x_t), \tag{3.4}$$

a *backward* poruke sa

$$\beta_i(x_i) = \bigoplus_{x_{i+1:T}} \bigotimes_{t=i+1}^{T} f_{A_t}(x_{t-1}, x_t). \tag{3.5}$$

Algoritam je sledeći.

Forward inicijalizacija: Za $x_0 \in \mathbb{X}$

$$\alpha_0(x_0) = u_{A_0}(x_0). \tag{3.6}$$

Forward rekurzija: Za $1 \leq i \leq l$ i $x_i \in \mathbb{X}$

$$\alpha_i(x_i) = \bigoplus_{x_{i-1}} u_{A_{i-1}}(x_{i-1}, x_i) \otimes \alpha_{i-1}(x_{i-1}). \tag{3.7}$$

Backward inicijalizacija: Za $x_T \in \mathbb{X}$

$$\beta_T(x_T) = 1. \tag{3.8}$$

Backward rekurzija: Za $T \geq i \geq r$ i $x_i \in \mathbb{X}$

$$\beta_t(x_t) = \bigoplus_{x_{t+1}} u_{A_{t+1}}(x_t, x_{t+1}) \otimes \beta_{t+1}(x_{t+1}). \tag{3.9}$$

Izračunavanje svih marginala: Za $x_{l:r} \in \mathbb{X}^{r-l+1}$

$$Z_{l:r}(x_{l:r}) = \bigoplus_{x_{-l:r}} u(x_{0:T}) = \alpha_l(x_l) \otimes \bigotimes_{i=l+1}^{r} u_{A_i}(x_{i-1}, x_i) \otimes \beta_r(x_r). \tag{3.10}$$

Ukoliko se rešava normalizacioni problem, izvršava se samo *forward* prolaz za $1 \leq i \leq T$, a zatim se izvršava

Normalizacija:

$$Z = \bigoplus_{x_T} \alpha_T(x_T). \tag{3.11}$$

3.2 Skriveni Markovljev model i *FB* algoritam

3.2.1 Skriveni Markovljev model

Skriveni Markovljev model (*hidden Markov model, HMM*) definišemo kao petorku

$$(\mathbb{X}, \mathcal{O}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\pi}), \tag{3.12}$$

gde je

- $\mathbb{X}$ konačan skup stanja modela,
- $\mathcal{O}$ konačan skup opservacija,
- $\boldsymbol{\pi} : \mathbb{X} \to \mathbb{R}$ iniciajlni vektor stanja,
- $\boldsymbol{A} : \mathbb{X}^2 \to \mathbb{R}$ matrica tranzicija, gde sa a_{ij} označavamo $\boldsymbol{A}(i, j)$ i za svako $x_{t-1} \in \mathbb{X}$ važi
$$\sum_{x_t \in \mathbb{X}} a_{x_{t-1}, x_t} = 1, \tag{3.13}$$
- $\boldsymbol{B} : \mathbb{X} \times \mathcal{O} \to \mathbb{R}$ je matrica emisija, gde sa $b_{x_t}(o_t)$ označavamo $\boldsymbol{B}(x_t, o_t)$ i za svako $x_t \in \mathbb{X}$ važi
$$\sum_{o_t} b_{x_t}(o_t) = 1. \tag{3.14}$$

Neka su $X_0, \ldots, X_T$ i $O_0, \ldots, O_T$ nizovi slučajnih promenljivih koje uzimaju vrednosti iz skupa stanja $\mathbb{X}$ i opservacija $\mathcal{O}$, respektivno, i neka je

$$p(x_{0:T}, o_{0:T}) = \pi_{x_0} b_{x_0}(o_0) \prod_{t=1}^{T} a_{x_{t-1} x_t} b_{x_t}(o_t). \tag{3.15}$$

Tada se, na osnovu (3.13)-(3.15), lako pokazuju sledeće jednakosti

$$\pi_i = P(X_0 = i), \quad a_{ij} = P(X_t = j | X_{t-1} = i), \quad b_i(o_t) = P(O_t = o_t | X_t = i), \tag{3.16}$$

kao i

$$p(x_{0:i}, o_{0:i}) = \pi_{x_0} b_{x_0}(o_0) \prod_{t=1}^{i} a_{x_{t-1} x_t} b_{x_t}(o_t), \tag{3.17}$$

$$p(x_{i+1:T}, o_{i+1:T} | x_t) = \prod_{t=i+1}^{T} a_{x_{t-1} x_t} b_{x_t}(o_t). \tag{3.18}$$

U primenama je od interesa uslovna verovatnoća sekvence $X_{0:T}$ za datu sekvencu opservacija $O_{0:T}$

$$p(x_{0:T}|o_{0:T}) = \frac{p(x_{0:T}, o_{0:T})}{p(o_{0:T})}. \tag{3.19}$$

Ukoliko uvedemo oznake

$$c_0 = p(o_0), \quad c_t = p(o_t|o_{0:t-1}), \tag{3.20}$$

za opservacionu verovatnoću imamo $p(o_{0:T}) = c_0 \cdot \prod_{t=1}^{T} c_t$, pa se uslovna verovatnoća $p(x_{0:T}|o_{0:T})$ može predstaviti u obliku

$$p(x_{0:T}|o_{0:T}) = \frac{\pi_{x_0} b_{x_0}(o_0)}{c_0} \prod_{t=1}^{T} \frac{a_{x_{t-1}x_t} b_{x_t}(o_t)}{c_t}, \tag{3.21}$$

koji ćemo koristiti u daljem tekstu. Takodje, koristićemo normalizovani oblik jednakosti (3.22) i (3.23)

$$p(x_{0:i}|o_{0:i}) = \frac{\pi_{x_0} b_{x_0}(o_0)}{c_0} \prod_{t=1}^{i} \frac{a_{x_{t-1}x_t} b_{x_t}(o_t)}{c_t}, \tag{3.22}$$

$$p(x_{i+1:T}|o_{i+1:T}|x_t) = \prod_{t=i+1}^{T} \frac{a_{x_{t-1}x_t} b_{x_t}(o_t)}{c_t}. \tag{3.23}$$

3.2.2 *HMM* ***forward-backward*** **algoritam**

Jedan od glavnih problema kod *HMM*-a je efikasno izračunavanje verovatnoće

$$p(x_{l:r}|o_{0:T}) = \sum_{x_{-l:r}} p(x_{0:T}|o_{0:T}). \tag{3.24}$$

Ovaj problem može biti rešen uz pomoć *FBA* nad *sum-product* poluprstenom, s obzirom na to da *HMM* verovatnoća $p(x_{0:T}|o_{0:T})$ može biti predstavljena faktorizacijom (3.1)

$$p(x_{0:i}|o_{0:i}) = z_0(x_0) \cdot \prod_{t=1}^{i} z_t(x_{t-1}, x_t), \tag{3.25}$$

gde su

$$z_0(x_0) = \frac{\pi_{x_0} b_{x_0}(o_0)}{c_0}, \quad \text{i} \quad z_t(x_{t-1}, x_t) = \frac{a_{x_{t-1}x_t} b_{x_t}(o_t)}{c_t}. \tag{3.26}$$

Na osnovu jednačine (3.22), uslovna verovatnoća može se predstaviti kao

$$p(x_{0:i}|o_{0:i}) = z_0(x_0) \cdot \prod_{t=1}^{i} z_t(x_{t-1}, x_t), \tag{3.27}$$

a *forward* poruka (3.4) ima oblik

$$\alpha_i(x_i) = \sum_{x_{0:i-1}} z_0(x_0) \cdot \prod_{i=t}^{i} z_t(x_{t-1}, x_t) = p(x_i|o_{0:i}). \tag{3.28}$$

Na osnovu (3.6), *forward* poruka se inicijalizuje na

$$\alpha_0(x_0) = z_0(x_0) = \frac{\pi_{x_0} b_{x_0}(o_0)}{c_0}, \tag{3.29}$$

a rekurzivna jednačina (3.7) se, u slučaju *FA*, svodi na

$$\alpha_t(x_t) = \sum_{x_{t-1}} z_t(x_{t-1}, x_t) \cdot \alpha_{t-1}(x_{t-1}) = \frac{\sum_{x_{t-1}} a_{x_{t-1}x_t} b_{x_t}(o_t)\alpha_{t-1}(x_{t-1})}{c_t}. \tag{3.30}$$

Normalizacioni faktori c_t nisu dati u specifikaciji modela, ali se mogu izračunati korišćenjem uslova

$$\sum_{x_t} \alpha_t(x_t) = \sum_{x_t} p(x_t|o_{0:t}) = 1, \tag{3.31}$$

što nam daje

$$c_0 = \sum_{j=1}^{N} \pi_j b_j(o_0) \quad \text{i} \quad c_t = \sum_{j=1}^{N}\sum_{i=1}^{N} \alpha_{t-1}^{(z)}(i) a_{ij} b_j(o_t). \tag{3.32}$$

Takodje, saglasno jednačini (3.23)

$$p(x_{i+1:T}, o_{i+1:T}|x_t) = \prod_{t=i+1}^{T} z_t(x_{t-1}, x_t), \tag{3.33}$$

backward poruka je

$$\beta_i(x_i) = \sum_{x_{i+1:T}} \prod_{t=i+1}^{T} z_t(x_{t-1}, x_t) = \frac{p(o_{t+1:T}|x_t)}{p(o_{t+1:T}|o_{0:t})}, \tag{3.34}$$

a njena rekurzivna jednačina

$$\beta_T(x_T) = 1, \tag{3.35}$$

$$\beta_t(x_t) = \frac{\sum_{x_{t+1}} a_{x_t x_{t+1}} b_{x_t}(o_t)\alpha_{t-1}(x_{t-1})}{c_{t+1}}. \tag{3.36}$$

Konačno, na sličan način, jednačina (3.10) svodi se na

$$p(x_{l:r}|o_{0:T}) = \alpha_l(x_l) \cdot \prod_{t=l+1}^{r} \frac{a_{x_{t-1},x_t} b_{x_t}(o_t)}{c_t} \cdot \beta_r(x_r), \tag{3.37}$$

čime je rešen marginalizacioni problem u *HMM*-u. U daljem tekstu, *FBA* primenjen za *HMM* nazivaćemo *HMMFBA*.

Napominjemo da se osnovna verzija *FBA* za *HMM* izvodi marginalizacijom združene raspodele $p(x_{0:T}, o_{0:T})$ umesto $p(x_{0:T}|o_{0:T})$, kao što smo činili u prethodnim paragrafima. U ovom slučaju *forward* poruke imaju oblik $\alpha_t(x_t) = p(x_t, o_{0:t})$, a backward poruke oblik $\beta_i(x_i) = p(o_{t+1:T}|x_t)$. U slučaju dugačkih sekvenci, vrednosti za poruke postaju male, tako da algoritam postaje numerički nestabilan (videti [6], [45], [74]), što nije slučaj kod algoritma izvedenog u ovom odeljku, u kome se propagiraju poruke normalizovane faktorima c_t.

Forward inicijalizacija: Za $1 \le j \le N$

$$c_0 = \sum_{j=1}^{N} \pi_j b_j(o_0), \tag{3.38}$$

$$\alpha_0(j) = \frac{\pi_j b_j(o_0)}{c_0}. \tag{3.39}$$

Forward rekurzija: Za $0 \leq t \leq T$, $1 \leq j \leq N$

$$c_t = \sum_{j=1}^{N} \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t), \tag{3.40}$$

$$\alpha_t(j) = \frac{\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)}{c_t}. \tag{3.41}$$

Backward inicijalizacija: Za $1 \leq i \leq N$

$$\beta_T(i) = 1. \tag{3.42}$$

Backward rekurzija: Za $T - 1 \geq t \geq 0$, $1 \leq i \leq N$

$$\beta_t(i) = \frac{\sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{c_{t+1}}. \tag{3.43}$$

Izračunavanje marginala: Za sve vrednosti $x_{l:r}$, izračunati marginalne vrednosti

$$p(x_{l:r}|o_{0:T}) = \alpha_l(x_l) \cdot \prod_{t=l+1}^{r} \frac{a_{x_{t-1},x_t} b_{x_t}(o_t)}{c_t} \cdot \beta_r(x_r). \tag{3.44}$$

Dva najčešće izračunavana marginala $p(x_{t-1}, x_t|o_{0:T})$ i $p(x_t|o_{0:T})$ mogu se dobiti pomoću formula

$$p(x_{t-1}, x_t|o_{0:T}) = \frac{\alpha_{t-1}(x_{t-1}) a_{x_{t-1}x_t} b_{x_t}(o_t) \beta_t(o_t)}{c_t}, \tag{3.45}$$

$$p(x_t|o_{0:T}) = \alpha_t(x_t) \cdot \beta_t(x_t). \tag{3.46}$$

Najveći deo izračunavanja se izvodi se u *forward* i *backward* algoritmu, što rezultuje vremenskom kompleksnošću $\mathcal{O}(N^2T)$. Čuvanje svih *forward* i *backward* promenljivih zajedno sa normalizacionim konstantama zahteva $\mathcal{O}(NT)$ memorije.

3.3 Izračunavanje entropije skrivenog Markovljevog modela

3.3.1 Entropija skrivenog Markovljevog modela

Uslovna entropija *HMM*-a je data sa

$$H(X_{0:T} \mid o_{0:T}) = -\sum_{x_{0:T}} p(x_{0:T} \mid o_{0:T}) \ln p(x_{0:T} \mid o_{0:T}), \tag{3.47}$$

a podsekvencna entropija sa

$$H(X_{-l:r}|x_{l:r}, o_{0:T}) = -\sum_{x_{-l:r}} p(x_{-l:r}|x_{l:r}, o_{0:T}) \cdot \ln p(x_{-l:r}|x_{l:r}, o_{0:T}). \tag{3.48}$$

Ukoliko uvedemo

$$H(X_{-l:r}, x_{l:r}|o_{0:T}) = -\sum_{x_{-l:r}} p(x_{0:T}|o_{0:T}) \cdot \ln p(x_{0:T}|o_{0:T}), \tag{3.49}$$

možemo izvesti sledeću jednakost

$$H(X_{-l:r}|x_{l:r}, o_{0:T}) = \frac{H(X_{-l:r}, x_{l:r}|o_{0:T}) + \ln p(x_{l:r}|o_{0:T})}{p(x_{l:r}|o_{0:T})}. \tag{3.50}$$

U narednom izlaganju razmatramo algoritme za efikasno izračunavanje entropije. U naredna dva odeljka, izvodimo dva algoritma bazirana na dekompozicionim pravilima za entropiju [14]

$$H(X, Y) = H(X) + H(Y|X), \tag{3.51}$$
$$H(Y|X) = \sum_x p(x) \cdot H(Y|X = x). \tag{3.52}$$

Nakon toga, izvodimo algoritme za izračunavanje entropije i podsekvencnom ograničene entropije bazirane na *FBA* nad entropijskim poluprstenom.

3.3.2 *Mann-McCallum* algoritam

Mann i *McCallum* su dali algoritam za izračunavanje gradijenta entropije *uslovnih slučajnih polja* [60], koji takodje može da se koristi za *HMM*. Algoritam koristi uslovne verovatnoće

$$p_{t|t+1}(i|j) = p(x_t|x_{t+1}, o_{0:T}) = \frac{p(x_t, x_{t+1}|o_{0:T})}{p(x_{t+1}|o_{0:T})}, \tag{3.53}$$
$$p_{t|t-1}(i|j) = p(x_t|x_{t-1}, o_{0:T}) = \frac{p(x_{t-1}, x_t|o_{0:T})}{p(x_{t-1}|o_{0:T})}, \tag{3.54}$$

koje se izračunavaju primenom *HMMFBA*, i *forward* i *backward* entropije, koje se izračunavaju rekurzivnim postupkom baziranim na formulama dekompozicije entropije (3.51). *Forward* entropija $H_t^\alpha(x_t)$ u trenutku t definisana je kao entropija sekvence stanja $X_{0:t-1}$ koja se završava u stanju x_t, za datu sekvencu opservacija $o_{0:T}$

$$H_t^\alpha(x_t) = H(X_{0:t-1}|x_t, o_{0:T}), \tag{3.55}$$

dok je *backward* entropija $H_t^\beta(x_t)$, entropija sekvence stanja $X_{t+1:T}$, koja startuje u x_t

$$H_t^\beta(x_t) = H(X_{t+1:T}|x_t, o_{0:T}). \tag{3.56}$$

Koristeći se *forward* i *backward* entropijom, podsekvencom ograničena *HMM* entropija može se izračunati sledećim algoritmom:

Forward-backward algoritam: Izračunati i sačuvati *forward* i *backward* poruke uz pomoć *FBA*.

Forward entropijska inicijalizacija: Za $1 \le j \le N$

$$H_0^\alpha(j) = 0. \tag{3.57}$$

Forward entropijska indukcija: Za $0 \le t \le T-1$, $1 \le j \le N$

$$H_{t+1}^\alpha(j) = \sum_{i=1}^{N} p_{t|t+1}(i|j)\Big(H_t^\alpha(i) - \ln p_{t|t+1}(i|j)\Big), \tag{3.58}$$

gde se $p_{t|t+1}(i|j)$ izračunava pomoću (3.45), (3.46) i (3.53).

Backward entropijska inicijalizacija: Za $1 \leq j \leq N$

$$H_T^{\beta}(j) = 0. \tag{3.59}$$

Backward entropijska rekurzija: Za $0 \leq t \leq T-1$, $1 \leq j \leq N$

$$H_{t-1}^{\beta}(j) = \sum_{i=1}^{T} p_{t|t-1}(i|j)\Big(H_t^{\beta}(i) - \ln p_{t|t-1}(i|j)\Big), \tag{3.60}$$

gde se $p_{t|t+1}(i|j)$ izračunava pomoću (3.45), (3.46) i (3.54).

Terminacija:

$$H(X_{-l:r}, x_{l:r}|o_{0:T}) = p(x_{l:r}|o_{0:T})(H_l^{\alpha}(x_l) + H_r^{\beta}(x_r) + \ln p(x_{l:r}|o_{0:T})), \tag{3.61}$$

$$H(X_{-l:r}|x_{l:r}, o_{0:T}) = \frac{H(X_{-l:r}, x_{l:r}|o_{0:T}) + \ln p(x_{l:r}|o_{0:T})}{p(x_{l:r}|o_{0:T})}. \tag{3.62}$$

Vremenska kompleksnost algoritma je $\mathcal{O}(N^2T + N^{r-l})$, gde je $\mathcal{O}(N^2T)$ za izračunavanje *forward* i *backward* entropija i $\mathcal{O}(N^{r-l})$ za terminacionu fazu. Memorijska kompleksnost zavisi od dužine sekvence, pošto su sve *forward* i *backward* poruke dostupne u fazama izračunavanja *forward* i *backward* entropija, a s obzirom na $\mathcal{O}(N^{r-l})$ prostora potrebnog za skladištenje rezultata u terminacionoj fazi, ukupna memorijska kompleksnost je $O(NT + N^{r-l})$.

Algoritam takodje može biti iskorišćen za izračunavanje entropije korišćenjem

$$H(X_{0:T}|o_{0:T}) = H(X_T|o_{0:T}) + \sum_{x_{0:T-1}} p(x_T|o_{0:T}) \cdot H_T^{(\alpha)}(x_T), \tag{3.63}$$

što sledi iz entropijskih dekompozicionih formula i definicije *forward* entropije. U ovom slučaju, *backward* prolaz za izračunavanje entropije nije potreban, ali se vremenska i memorijska kompleksnost ne smanjuje, pošto je još uvek potrebno izračunati *forward* i *backward* poruke. U narednom odeljku dajemo pregled algoritma za izračunavanje entropije uz memorijsku kompleksnost nezavisnom od dužine sekvence koji su izveli Hernando i njegovi saradnici u [31] .

3.3.3 Algoritam Hernanda i saradnika

Hernando i saradnici su razvili rekurzivni algoritam za izračunavanje *HMM* entropije [31]. Algoritam koristi: *HMM forward* verovatnoću

$$\alpha_t(x_t) = p(x_t|o_{1:t}), \tag{3.64}$$

uslovnu verovatnoću

$$p_{t|t-1}(x_t|x_{t-1}) = p(x_{t-1}|x_t, o_{1:t}), \tag{3.65}$$

i uslovnu entropiju

$$H_t(x_t) = H(X_{0:t-1}|x_t, o_{1:t}). \tag{3.66}$$

Algoritam sledi.

Inicijalizacija: Za $1 \le j \le N$

$$H_0(j) = 0, \tag{3.67}$$

$$\alpha_0(j) = \frac{\pi_j b_j(o_0)}{\sum_{i=1}^{N} \pi_i b_i(o_1)}. \tag{3.68}$$

Indukcija: Za $1 \le t \le T$ i $1 \le i, j \le N$

$$\alpha_t(j) = \frac{\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij} b_j(o_t)}{\sum_{k=1}^{N} \sum_{i=1}^{N} \alpha_{t-1}(i) a_{ik} b_k(o_t)}, \tag{3.69}$$

$$p_{t-1|t}(i|j) = \frac{\alpha_{t-1}(i) a_{ij}}{\sum_{k=1}^{N} \sum_{i=1}^{N} \alpha_{t-1}(k) a_{kj}}, \tag{3.70}$$

$$H_t(j) = \sum_{i=1}^{N} p_{t-1|t}(i|j) \Big(H_{t-1}(i) - \ln p_{t-1|t}(i|j) \Big). \tag{3.71}$$

Terminacija:

$$H(X_{0:T} \mid o_{0:T}) = \sum_{j=1}^{N} \alpha_T(j) \Big(H_T(j) - \ln \alpha_T(j) \Big). \tag{3.72}$$

Algoritam se izvršava uz linearnu vremensku kompleksnost $O(N^2T)$ i u fiksnom memorijskom prostoru nezavisnom od dužine sekvence $O(N^2)$, pošto se vektori α_{t-1}, H_{t-1} i matrica $p_{t-1|t}$ koriste samo u t – 1-oj iteraciji i, pošto se iskoriste za izračunavanje H_t, mogu biti izbrisani.

3.4 Izračunavanje *HMM* entropije i podsekvencom ograničene entropije primenom *EMP* algoritma

U ovom poglavlju razmatramo *FB* algoritam nad entropijskim poluprstenom i njegovu primenu za izračunavanje *HMM* entropije. Algoritam predstavlja instancu *EMP* algoritma iz odeljka 2.3.2 primenjenog za izračunavanje *HMM* entropije.

U odeljku 2.3.2 pomenuli smo da se poluprsten očekivanja prvog reda svodi na entropijski poluprsten. Radi preglednosti ponavljamo definiciju entropijskog poluprstena.

Definicija 3 *Entropijski poluprsten definisan je kao petorka*

$$\big(\mathbb{R}^2, \oplus, \otimes, (0, 0), (1, 0) \big), \tag{3.73}$$

gde su operacije $\oplus$ *i* $\otimes$ *definisane sa*

$$(z_1, h_1) \oplus (z_2, h_2) = (z_1 + z_2, z_1 + z_2), \tag{3.74}$$

$$(z_1, h_1) \otimes (z_2, h_2) = (z_1 z_2, z_1 h_2 + z_2 h_1), \tag{3.75}$$

za sve parove $(z_1, h_1), (z_2, h_2) \in \mathbb{R}^2$.

Kao što smo pomenuli, prva komponenta uredjenog para $w \in \mathbb{R}^2$ naziva se z-komponenta i označava se sa $w^{(z)}$, a druga komponenta naziva se h-komponenta i označava se sa $w^{(h)}$. U slučaju lanca, lema se svodi na sledeću lemu.

Lema 3.4.1 *Neka su* $(z_i, z_i h_i) \in \mathbb{R}^2$ *za* $0 \le i \le T$. *Tada važi sledeća jednakost*

$$\bigotimes_{i=0}^{T}(z_i, z_i h_i) = \left(\prod_{i=0}^{T} z_i \, , \, \prod_{i=0}^{T} z_i \cdot \sum_{j=0}^{T} h_j \right). \tag{3.76}$$

Neka promenljiva $x_{0:T} = (x_0, \cdots, x_T)$ uzima vrednosti iz skupa, $\mathbb{X}^{T+1}$ i neka se funkcija $u : \mathbb{X}^T \to \mathbb{R}^2$ faktoriše u entropijskom poluprstenu $(\mathbb{R}^2, \oplus, \otimes, 0, 1)$ uz pomoć faktora $u_{A_0} : \mathbb{X} \to \mathbb{R}^2$ i $u_{A_t} : \mathbb{X}^2 \to \mathbb{R}^2$ kao

$$u(x_{0:T}) = u_{A_0}(x_0) \otimes \bigotimes_{i=1}^{T} u_{A_i}(x_{i-1}, x_i). \tag{3.77}$$

pri čemu faktori imaju oblik

$$u_{A_0}(x_0) = \big(z_0(x_0), z_0(x_0) h_0(x_0)\big), \tag{3.78}$$

$$u_{A_i}(x_{i-1}, x_i) = \big(z_i(x_{i-1}, x_i), z_i(x_{i-1}, x_i) h_i(x_{i-1}, x_i)\big), \tag{3.79}$$

gde je

$$z_0(x_0) = \frac{\pi_{x_0} b_{x_0}(o_0)}{c_0}, \quad \text{i} \quad z_t(x_{t-1}, x_t) = \frac{a_{x_{t-1}x_t} b_{x_t}(o_t)}{c_t}, \tag{3.80}$$

sa $c_0 = p(o_0)$, $c_t = p(o_t | o_{0:t-1})$ i

$$h_0(x_0) = \ln z_0(x_0), \quad \text{i} \quad h_t(x_{t-1}, x_t) = \ln z_t(x_{t-1}, x_t). \tag{3.81}$$

Iz leme 3.4.1 sledi da su z i h-komponente izraza

$$u(x_{0:T}) = u_{A_0}(x_0) \otimes \bigotimes_{i=1}^{T} u_i(x_{i-1}, x_i), \tag{3.82}$$

date sa

$$u(x_{0:T})^{(z)} = z_0(x_0) \prod_{i=1}^{T} z_i(x_{i-1}, x_i), \tag{3.83}$$

$$u(x_{0:T})^{(h)} = z_0(x_0) \prod_{i=1}^{T} z_i(x_{i-1}, x_i) \cdot \Big(h_0(x_0) + \sum_{j=1}^{T} h_j(x_{j-1}, x_j)\Big). \tag{3.84}$$

Primetimo da

$$h_0(x_0) + \sum_{j=1}^{T} h_j(x_{j-1}, x_j) = \ln\Big(z_0 \cdot \prod_{j=1}^{T} z_j(x_{j-1}, x_j)\Big), \tag{3.85}$$

i na osnovu faktorizacije (3.21) za *HMM* uslovnu verovatnoću

$$p(x_{0:T} | o_{0:T}) = z_0 \cdot \prod_{i=1}^{T} z_i(x_{i-1}, x_i), \tag{3.86}$$

dobija se

$$u(x_{0:T}) = \big(p(x_{0:T} | o_{0:T}) \, , \, p(x_{0:T} | o_{0:T}) \ln p(x_{0:T} | o_{0:T}) \big). \tag{3.87}$$

Sada se entropije $H(X_{0:T}|o_{0:T})$ i $H(X_{-l:r}, x_{l:r}|o_{0:T})$ mogu dobiti sumiranjem (3.87) u entropijskom poluprstenu kao h-komponente suma

$$\Big(\bigoplus_{x_{1:T}} u(x_{1:T})\Big)^{(h)} = -H(X_{0:T}|o_{0:T}), \tag{3.88}$$

$$\Big(\bigoplus_{x_{-l:r}} u(x_{0:T})\Big)^{(h)} = -H(X_{l:r}, x_{l:r}|o_{0:T}). \tag{3.89}$$

Dva različita načina za sumiranje, koja odgovaraju normalizacionom i marginalizacionom problemu mogu biti izvršena *FB* algoritmom nad entropijskim poluprstenom, pri čemu z i h-komponente *forward* i *backward* poruka mogu biti izvedene uz pomoć leme 3.4.1.

Za *forward* poruku

$$\alpha_t(x_t) = \bigoplus_{x_{0:t-1}} u_{A_0}(x_0) \otimes \bigotimes_{i=1}^{t} u_{A_i}(x_{i-1}, x_i), \tag{3.90}$$

imamo

$$\alpha_t^{(z)}(x_t) = \sum_{x_{0:t-1}} z_0(x_0) \cdot \prod_{i=1}^{t} z_i(x_{i-1}, x_i), \tag{3.91}$$

$$\alpha_t^{(h)}(x_t) = \sum_{x_{0:t-1}} z_0(x_0) \cdot \prod_{i=1}^{t} z_i(x_{i-1}, x_i) \cdot \Big(h_0(x_0) + \sum_{j=1}^{t} h_j(x_{j-1}, x_j)\Big), \tag{3.92}$$

i korišćenjem jednakosti

$$p(x_{0:t}|o_{0:t}) = z_0(x_0) \prod_{i=1}^{t} z_i(x_{i-1}, x_i), \tag{3.93}$$

dobijamo

$$\alpha_t^{(z)}(x_t) = \sum_{x_{o:t}} p(x_{0:t}|o_{0:t}) = p(x_t|o_{0:t}), \tag{3.94}$$

$$\alpha_t^{(h)}(x_t) = \sum_{x_{o:t}} p(x_{0:t}|o_{0:t}) \ln p(x_{0:t}|o_{0:t}). \tag{3.95}$$

Z-komponenta *ESR forward* poruke je *HMM forward* promenljiva, definisana u odeljku 3.2.2, dok se informacija o podsekvencnim entropijama prenosi kroz h-komponentu, tako da u svakom koraku imamo

$$H(X_{0:t}|o_{0:t}) = \sum_{x_t} \alpha_t^{(h)}(x_t). \tag{3.96}$$

Forward poruka se inicijalizuje na $f_{A_0}(x_0)$ i, s obzirom na (3.78), dobija se

$$\alpha_0^{(z)}(x_0) = z_0(x_0) = \frac{\pi_{x_o} b_{x_o}(o_0)}{c_0}, \tag{3.97}$$

$$\alpha_0^{(h)}(y_0) = z_0(x_0) h_0(x_0) = \frac{\pi_{x_o} b_{x_o}(o_0)}{c_0} \ln \frac{\pi_{x_o} b_{x_o}(o_0)}{c_0}. \tag{3.98}$$

Z i h-komponente *forward* jednačine

$$\alpha_i(x_i) = \bigoplus_{x_{i-1}} u_{i-1}(x_{i-1}, x_i) \otimes \alpha_{i-1}(x_{i-1}), \tag{3.99}$$

mogu se odrediti na osnovu definicije entropijskog poluprstena kao

$$\alpha_i^{(z)}(x_i) = \sum_{x_{i-1}} z_i(x_{i-1}, x_i)\alpha_{i-1}^{(z)}(x_{i-1}), \tag{3.100}$$

$$\alpha_i^{(h)}(x_i) = \sum_{x_{i-1}} z_i(x_{i-1}, x_i)\big(\alpha_{i-1}^{(h)}(x_{i-1}) + h_i(x_{i-1}, x_i)\alpha_{i-1}^{(z)}(x_{i-1})\big), \tag{3.101}$$

ili, ekvivalentno

$$\alpha_t^{(z)}(j) = \sum_{i=1}^{N} \frac{a_{ij}b_j(o_t)}{c_t} \cdot \alpha_{t-1}^{(z)}(i), \tag{3.102}$$

$$\alpha_i^{(h)}(j) = \sum_{i=1}^{N} \frac{a_{ij}b_j(o_t)}{c_t}\big(\alpha_{t-1}^{(h)}(i) + \alpha_{t-1}^{(z)}(i) \ln \frac{a_{ij}b_j(o_t)}{c_t}\big). \tag{3.103}$$

Faktori c_t se mogu naći normalizacijom z-komponente kao u odeljku (3.2.2)

$$c_0 = \sum_{j=1}^{N} \pi_j b_j(o_0), \tag{3.104}$$

$$c_t = \sum_{j=1}^{N}\sum_{i=1}^{N} \alpha_{t-1}^{(z)}(i)a_{ij}b_j(o_t). \tag{3.105}$$

Backward poruka

$$\beta_i(x_i) = \bigoplus_{x_{i+1:T}} \bigotimes_{t=i+1}^{T} f_{A_t}(x_{t-1}, x_t), \tag{3.106}$$

ima z i h-komponente

$$\beta_t^{(z)}(x_t) = \sum_{x_{t+1:T}} \prod_{i=t+1}^{T} z_i(x_{i-1}, x_i), \tag{3.107}$$

$$\beta_t^{(h)}(x_t) = \sum_{x_{t+1:T}} \prod_{i=t+1}^{T} z_i(x_{i-1}, x_i) \cdot \sum_{j=t+1}^{T} h_j(x_{j-1}, x_j). \tag{3.108}$$

Jednakost (3.23) implicira

$$\prod_{i=t+1}^{T} z_i(x_{i-1}, x_i) = \frac{p(x_{t+1:T}, o_{t+1:T}|x_t)}{p(o_{t+1:T}|o_{0:t})}, \tag{3.109}$$

i dobijamo

$$\beta_t^{(z)}(x_t) = \frac{p(o_{t+1:T}|x_t)}{p(o_{t+1:T}|o_{0:t})}, \tag{3.110}$$

$$\beta_t^{(h)}(x_t) = \sum_{x_{t+1:T}} \frac{p(x_{t+1:T}, o_{t+1:T}|x_t)}{p(o_{t+1:T}|o_{0:t})} \ln \frac{p(x_{t+1:T}, o_{t+1:T}|x_t)}{p(o_{t+1:T}|o_{0:t})}, \tag{3.111}$$

iz čega zaključujemo da je z-komponenta *ESR backward* poruke ista kao *HMM backward* promenljiva iz odeljka 3.2.2.

Backward poruka se inicijalizuje na $\beta_T(x_T) = 1$, pa je

$$\beta_T(x_T)^{(z)} = 1, \tag{3.112}$$
$$\beta_T(x_T)^{(h)} = 0, \tag{3.113}$$

dok se rekurzivna jednačina

$$\beta_t(x_t) = \bigoplus_{x_{t+1}} u_{t+1}(x_t, x_{t+1}) \otimes \beta_{t+1}(x_{t+1}), \tag{3.114}$$

svodi na

$$\beta_i^{(z)}(x_i) = \sum_{x_{i+1}} z(x_i, x_{i+1}) \beta_{i+1}^{(z)}(x_{i+1}), \tag{3.115}$$
$$\beta_i^{(h)}(x_i) = \sum_{x_{i+1}} z(x_i, x_{i+1}) \big(\beta_{i+1}^{(h)}(x_{i+1}) + h(x_i, x_{i+1}) \alpha_{i+1}^{(z)}(x_{i+1}) \big), \tag{3.116}$$

ili ekvivalentno

$$\beta_t^{(z)}(i) = \sum_j \frac{a_{ij} b_j(o_t)}{c_{t+1}} \beta_{t+1}^{(z)}(j), \tag{3.117}$$
$$\beta_t^{(h)}(i) = \sum_j \frac{a_{ij} b_j(o_t)}{c_{t+1}} \Big(\beta_{t+1}^{(h)}(j) + \beta_{t+1}^{(z)}(j) \ln \frac{a_{ij} b_j(o_t)}{c_{t+1}} \Big), \tag{3.118}$$

gde se normalizacione konstante c_t izračunavaju u *forward* prolazu.

3.4.1 Izračunavanje *HMM* entropije primenom *EMP* algoritma

Ako je sumacija faktorizacije (3.87) izvršena nad celom sekvencom

$$\bigoplus_{x_{0:T}} u(x_{0:T}) = \bigoplus_{x_{0:T}} \big(\, p(x_{0:T}|o_{0:T}) \, , \, p(x_{0:T}|o_{0:T}) \ln p(x_{0:T}|o_{0:T}) \, \big), \tag{3.119}$$

z i h-komponente sume svode se na

$$\Big(\bigoplus_{x_{1:T}} u(x_{1:T}) \Big)^{(z)} = \sum_{x_{0:T}} p(x_{0:T}|o_{0:T}) = 1, \tag{3.120}$$
$$\Big(\bigoplus_{x_{1:T}} u(x_{1:T}) \Big)^{(h)} = \sum_{x_{0:T}} p(x_{0:T}|o_{0:T}) \ln p(x_{0:T}|o_{0:T}) = -H(X_{0:T}|o_{0:T}). \tag{3.121}$$

H-komponenta sume odgovara *HMM* entropiji i može se naći kao rešenje normalizacionog problema

$$\Big(\bigoplus_{x_{1:T}} u(x_{1:T}) \Big)^{(h)} = \sum_{x_T} \alpha_T^{(h)}(x_T), \tag{3.122}$$

korišćenjem samo *forward* prolaza, na osnovu jednačina (3.97)-(3.98), (3.102)-(3.105). Algoritam sledi.

Inicijalizacija: Za $1 \leq j \leq N$

$$c_0 = \sum_{j=1}^{N} \pi_j b_j(o_0), \tag{3.123}$$

$$\alpha_0^{(z)}(j) = \frac{\pi_j b_j(o_0)}{c_0}, \tag{3.124}$$

$$\alpha_0^{(h)}(j) = \frac{\pi_j b_j(o_0)}{c_0} \ln \frac{\pi_j b_j(o_0)}{c_0}. \tag{3.125}$$

Indukcija: Za $1 \leq t \leq T$, $1 \leq j \leq N$

$$c_t = \sum_{j=1}^{N} \sum_{i=1}^{N} \alpha_{t-1}^{(z)}(i) a_{ij} b_j(o_t), \tag{3.126}$$

$$\alpha_t^{(z)}(j) = \sum_{i=1}^{N} \frac{a_{ij} b_j(o_t)}{c_t} \cdot \alpha_{t-1}^{(z)}(i), \tag{3.127}$$

$$\alpha_i^{(h)}(j) = \sum_{i=1}^{N} \frac{a_{ij} b_j(o_t)}{c_t} \Big(\alpha_{t-1}^{(h)}(i) + \alpha_{t-1}^{(z)}(i) \ln \frac{a_{ij} b_j(o_t)}{c_t} \Big). \tag{3.128}$$

Terminacija:

$$H(X_{0:T}|o_{0:T}) = -\sum_{j=1}^{N} \alpha_T^{(h)}(j). \tag{3.129}$$

EMP algoritam ima vremensku kompleksnost $\mathcal{O}(N^2T)$ i memorijsku kompleksnost $\mathcal{O}(N)$, kao i algoritam koji su dali *Hernando* i saradnici. Dva algoritma su uskoj vezi, kao što je opisano u sledećem tekstu.

EMP i algoritam Hernanda i saradnika

Oba algoritma izračunavaju *forward* verovatnoću

$$\alpha_t^{(z)}(x_t) = p(x_t|o_{0:t}). \tag{3.130}$$

Razlika izmedju ova dva algoritma je u drugoj veličini koja se izračunava. U algoritmu koji su dali *Hernando* i saradnici to je uslovna entropija

$$H_t(x_t) = H(X_{0:t-1}|x_t, o_{1:t}) = -\sum_{x_{0:t-1}} p(x_{0:t-1}|x_t, o_{1:t}) \ln p(x_{0:t-1}|x_t, o_{1:t}), \tag{3.131}$$

dok je u *EMP* algoritmu to h-komponenta *forward* poruke:

$$\alpha_t^{(h)}(x_t) = \sum_{x_{0:t-1}} p(x_{0:t}|o_{0:t}) \ln p(x_{0:t}|o_{0:t}). \tag{3.132}$$

Veza izmedju ovih veličina

$$\alpha_t^{(h)} = \alpha_t^{(z)} H_t(x_t) + \alpha_t^{(z)} \ln \alpha_t^{(z)}, \tag{3.133}$$

može se izvesti jednostavno korišćenjem elementarnih transformacija verovatnoće.

Pored toga, na osnovu faktorizacije za *HMM* združenu raspodelu verovatnoće (3.15), možemo izvesti Markovljeva svojstva

$$p(o_t|x_t, x_{t-1}, o_{0:t-1}) = p(o_t|x_t), \qquad p(x_t|x_{t-1}, o_{0:t-1}) = p(x_t|x_{t-1}), \tag{3.134}$$

koja impliciraju jednakosti

$$\begin{aligned}\frac{a_{x_{t-1}x_t} b_{x_t}(o_t)}{c_t} &= \frac{p(x_t|x_{t-1})p(o_t|x_t)}{p(o_t|o_{0:t-1})} = \\ &= \frac{p(o_t, x_t|x_{t-1}, o_{0:t-1})}{p(o_t|o_{0:t-1})} = \frac{p_{t-1|t}(x_{t-1}|x_t) \cdot \alpha_t^{(z)}(x_t)}{\alpha_{t-1}^{(z)}(x_{t-1})},\end{aligned} \tag{3.135}$$

gde je $p_{t-1|t}(x_{t-1}|x_t) = p(x_{t-1}|x_t, o_{0:t})$ kao što su definisali *Hernando* i saradnika. Na osnovu ovoga, rekurzivna jednačina za $H_t(x_t)$, koju su izveli *Hernando* i saradnici, može da se dobije iz *EMP* algoritma zamenom izraza (3.135) i (3.133) u jednačinu za $\alpha_t^{(h)}$ u *EMP* algoritmu, čime je uspostavljena veza izmedju dva algoritma.

3.4.2 Izračunavanje *HMM* podsekvencom ograničene entropije primenom *EMP* algoritma

Ako se sumacija faktorizacije (3.77) izvrši nad podsekvencom $x_{-l:r}$

$$\bigoplus_{x_{-l:r}} u(x_{0:T}) = \bigoplus_{x_{-l:r}} \left(\, p(x_{0:T}|o_{0:T}) \, , \, p(x_{0:T}|o_{0:T}) \ln p(x_{0:T}|o_{0:T}) \, \right), \tag{3.136}$$

z i h-komponente suma su

$$\left(\bigoplus_{x_{-l:r}} u(x_{0:T}) \right)^{(z)} = p(x_{l:r}), \tag{3.137}$$

$$\left(\bigoplus_{x_{-l:r}} u(x_{0:T}) \right)^{(h)} = -H(X_{l:r}, x_{l:r}|o_{0:T}). \tag{3.138}$$

H-komponenta sume odgovara *HMM* podsekvencom ograničenoj entropiji i može se dobiti kao rešenje marginalizacionog problema

$$Z_{l:r}(x_{l:r}) = \alpha_l(x_l) \otimes \bigotimes_{i=l+1}^{r} u_i(x_{i-1}, x_i) \otimes \beta_r(x_r). \tag{3.139}$$

Z i h-komponente mogu se izračunati uz pomoć definicije operacija u entropijskom poluprstenu:

$$\left(\bigoplus_{x_{-l:r}} u(x_{0:T}) \right)^{(z)} = \alpha_l^{(z)}(x_l)\beta_r^{(z)}(x_r) \prod_{i=l+1}^{r} z_i(x_{i-1}, x_i), \tag{3.140}$$

$$\begin{aligned}\left(\bigoplus_{x_{-l:r}} u(x_{0:T}) \right)^{(h)} &= \prod_{i=l+1}^{r} z_i(x_{i-1}, x_i) \cdot \\ &\left(\alpha_l^{(z)}(x_l)\beta_r^{(h)}(x_r) + \alpha_l^{(h)}(x_l)\beta_r^{(z)}(x_r) + \alpha_l^{(z)}(x_l)\beta_r^{(z)}(x_r) \sum_{j=l+1}^{r} h_j(x_{j-1}, x_j) \right)\end{aligned} \tag{3.141}$$

Da bismo izračunali h-komponentu marginala, potrebne su l-ta *forward* i r-ta *backward* poruke. l-ta *forward* poruka može se izračunati uz pomoć *ESR forward* algoritma datog jednačinama (3.123)-(3.128). Medjutim, induktivni koraci (3.126)-(3.128) za normalizacione konstante c_t i z-komponente *forward* poruka moraju biti izvršeni za svako t, pošto normalizacione konstante $c_t, r < t \leq T$ moraju biti dostupne u *backward* prolazu. Pošto su normalizacione konstate izračunate, *backward* prolaz može da se izvrši pomoću jednačina (3.112)-(3.113), (3.117)-(3.118), i posle toga možemo izračunati podsekvencom ograničenu entropiju, koristeći se jednakostima (3.140)-(3.141) i (3.50). Algoritam sledi.

Forward inicijalizacija: Za $1 \leq j \leq N$,

$$c_0 = \sum_{j=1}^{N} \pi_j b_j(o_0), \tag{3.142}$$

$$\alpha_0^{(z)}(j) = \frac{\pi_j b_j(o_0)}{c_0}, \tag{3.143}$$

$$\alpha_0^{(h)}(j) = \frac{\pi_j b_j(o_0)}{c_0} \ln \frac{\pi_j b_j(o_0)}{c_0}. \tag{3.144}$$

Potpuna forward rekurzija: Za $1 \leq t \leq l$, $1 \leq j \leq N$,

$$c_t = \sum_{j=1}^{N} \sum_{i=1}^{N} \alpha_{t-1}^{(z)}(i) a_{ij} b_j(o_t), \tag{3.145}$$

$$\alpha_t^{(z)}(j) = \sum_{i=1}^{N} \frac{a_{ij} b_j(o_t)}{c_t} \cdot \alpha_{t-1}^{(z)}(i), \tag{3.146}$$

$$\alpha_i^{(h)}(j) = \sum_{i=1}^{N} \frac{a_{ij} b_j(o_t)}{c_t} \cdot \Big(\alpha_{t-1}^{(h)}(i) + \alpha_{t-1}^{(z)}(i) \ln \frac{a_{ij} b_j(o_t)}{c_t}\Big). \tag{3.147}$$

Forward rekurzija za izračunavanje z-komponente: Za $l+1 \leq t \leq T$, $1 \leq j \leq N$,

$$c_t = \sum_{j=1}^{N} \sum_{i=1}^{N} \alpha_{t-1}^{(z)}(i) a_{ij} b_j(o_t), \tag{3.148}$$

$$\alpha_t^{(z)}(j) = \sum_{i=1}^{N} \frac{a_{ij} b_j(o_t)}{c_t} \cdot \alpha_{t-1}^{(z)}(i). \tag{3.149}$$

Backward inicijalizacija: Za $1 \leq i \leq N$,

$$\beta_T^{(z)}(j) = 1, \tag{3.150}$$

$$\beta_T^{(h)}(j) = 0. \tag{3.151}$$

Backward rekurzija: Za $T-1 \geq t \geq r$, $1 \leq j \leq N$,

$$\beta_t^{(z)}(i) = \sum_{j} \frac{a_{ij} b_j(o_t)}{c_{t+1}} \beta_{t+1}^{(z)}(j), \tag{3.152}$$

$$\beta_t^{(h)}(i) = \sum_{j} \frac{a_{ij} b_j(o_t)}{c_{t+1}} \cdot \Big(\beta_{t+1}^{(h)}(j) + \beta_{t+1}^{(z)}(j) \ln \frac{a_{ij} b_j(o_t)}{c_{t+1}}\Big). \tag{3.153}$$

Terminacija: Za $l \le t \le r$, $1 \le x_t \le N$, izračunati podsekvencom ograničenu entopiju

$$p(x_{l:r}) = \alpha_l^{(z)}(x_l)\beta_r^{(z)}(x_r) \prod_{t=l+1}^{r} \frac{a_{x_{t-1}x_t} b_{x_t}(o_t)}{c_t}, \tag{3.154}$$

$$\begin{aligned} -H(X_{l:r}, x_{l:r}|o_{0:T}) &= \prod_{t=l+1}^{r} \frac{a_{x_{t-1}x_t} b_{x_t}(o_t)}{c_t} \cdot \\ &\left(\alpha_l^{(z)}(x_l)\beta_r^{(h)}(x_r) + \alpha_l^{(h)}(x_l)\beta_r^{(z)}(x_r) + \alpha_l^{(z)}(x_l)\beta_r^{(z)}(x_r) \sum_{j=l+1}^{r} h_j(x_{j-1}, x_j)\right), \end{aligned} \tag{3.155}$$

$$H(X_{-l:r}|x_{l:r}, o_{0:T}) = \frac{H(X_{-l:r}, x_{l:r}|o_{0:T}) + \ln p(x_{l:r}|o_{0:T})}{p(x_{l:r}|o_{0:T})}. \tag{3.156}$$

Vremenska kompleksnost algoritma je $\mathcal{O}(N^2T + N^{r-l})$, gde je $\mathcal{O}(N^2T)$ za *forward* i *backward* rekurzije, i $\mathcal{O}(N^{r-l})$ za terminacionu fazu, što je ista kompleksnost kao i u algoritmu koji su dali *Mann* i *MacCallum*.

S druge strane, potpuna *forward* faza može biti realizovana u $\mathcal{O}(N^2 l)$ vremenu i memoriji fiksne veličine $\mathcal{O}(N)$, s obzirom na to da $\alpha_{t-1}^{(z)}$, $\alpha_{t-1}^{(h)}$ i c_{t-1} mogu biti obrisani, pošto se iskoriste za izračunavanje $\alpha_t^{(z)}$, $\alpha_t^{(h)}$ i c_t. Slično, za *forward* indukciju za z-komponenta i i *backward* rekurziju potrebno je $\mathcal{O}(N)$ prostora. Jedini dodatni prostor koji zavisi od dužine sekvence $\mathcal{O}(T - l)$ potreban je za normalizacione konstante koje se izračunavaju u *forward* rekurzionoj fazi za z-komponente, pošto one moraju biti dostupne u *backward* i terminacionoj fazi. Konačno, s obzirom na $\mathcal{O}(N^{r-l})$ prostora potrebnog za čuvanje rezultata u terminacionoj fazi, ukupna memorijska kompleksnost je $\mathcal{O}(T - l + N^{r-l})$, što se sporije uvećava sa T u odnosu na $O(NT + N^{r-l})$, što je slučaj kod algoritma koji su dali *Mann* i *McCallum*.

Glava 4

Izračunavanje gradijenta uslovnih slučajnih polja

Uslovna slučajna polja (*conditional random fields, CRF*) [51] su probabilistički model koji predstavlja alternativu skrivenom Markovljevom modelu za labeliranje sekvenci. Prednost *CRF*-a u odnosu na *HMM* ogleda se u tome što se kod *CRF*-a ne pretpostavlja nezavisnost tekućeg stanja od prethodnih opservacija, kao što je to slučaj kod *HMM*-a. Medjutim, ovo unapredjenje je praćeno uvećanjem potrebnog vremena i prostora za estimaciju parametara modela, posebno u slučajevima kada se koristi za jako dugačke sekvence, koje se mogu javiti u bezbednosti računara [52], [85], bioinformatici [48], [62] i robotici [47].

Estimacija *CRF* parametara obično se izvodi nekim od gradijentnih metoda, kao što su: *iterative scaling*, konjugovani gradijent ili memorijom ograničene kvazi-Njutnove metode [28], [51], [76], [79], [84]. Sve ovi metode zahtevaju izračunavanje gradijenta verodostojnosti, što postaje zahtevno sa porastom dužine sekvence i broja klasa. Standardni metod za izračunavanje gradijenta [51] baziran je na internim izračunavanjima marginalnih *CRF* verovatnoća uz pomoć *FB* algoritma.

U ovoj glavi dajemo algoritam za egzaktno izračunavanje *CRF* gradijenta. Algoritam je izveden kao *forward* algoritam nad log-domen poluprstenom očekivanja, i može se smatrati numerički stabilnom verzijom *EMP* algoritma iz odeljka 2.3.2 primenjenog na *CRF*. Pošto se izvršava sa jednim *forward* prolazom, *EMP* se može implementirati uz memorijsku kompleksnost nezavisnu od dužine sekvence i na taj način ostvaruje prednost u odnosu na *FB* u slučaju dugačkih sekvenci.

Glava je organizovana na sledeći način. U poglavlju 4.1 dajemo rekapitulaciju *FB* algoritma. *CRF* model razmatramo u poglavlju 4.2, u kome su objašnjeni i standardni algoritmi za izračunavanje gradijenta, bazirani na *FB* algoritmu nad *sum – product* i log-domen *sum-product* poluprstenom. Algoritme bazirane na *EMP* algoritmu razmatramo u poglavlju 4.3.

4.1 *FB* algoritam - rekapitulacija

Kao i u poglavlju 3.1, najpre dajemo rekapitulaciju *FB* algoritma, za nešto drugačiju faktorizaciju u odnosu na onu u odeljku 2.2.4. Neka promenljiva $x_{0:T}$ uzima vrednosti iz skupa $\mathbb{X}^{T+1}$ i neka se funkcija $u : \mathbb{X}^{T+1} \to \mathbb{K}$ faktoriše u komutativnom poluprstenu $(\mathbb{K}, \oplus, \otimes, 0, 1)$ uz

pomoć faktora $u_{A_t} : \mathbb{X}^2 \to \mathbb{K}$ kao

$$u(x_{0:T}) = \bigotimes_{t=1}^{T} u_{A_t}(x_{t-1}, x_t). \tag{4.1}$$

FB algoritam se koristi za rešavanje sledeća dva problema:

Marginalizacioni problem: Izračunati sumu

$$Z_{i-1:i}(x_{i-1}, x_i) = \bigoplus_{x_{-i-1:i}} u(x_{0:T}). \tag{4.2}$$

Normalizacioni problem: Izračunati sumu

$$Z = \bigoplus_{x_{0:T}} u(x_{0:T}). \tag{4.3}$$

Forward poruke definisane su sa

$$\alpha_i(x_i) = \bigoplus_{x_{0:i-1}} \bigotimes_{t=1}^{i} u_{A_t}(x_{t-1}, x_t), \tag{4.4}$$

za $i = 1, \ldots, T$ i $\alpha_0(x_0) = 1$, a *backward* poruke definisane su sa

$$\beta_i(x_i) = \bigoplus_{x_{i+1:T}} \bigotimes_{t=i+1}^{T} u_{A_t}(x_{t-1}, x_t), \tag{4.5}$$

za $i = 0, \ldots, T-1$ i $\beta_T(x_T) = 1$. Algoritam je sledeći.

Forward inicijalizacija: Za $x_0 \in \mathbb{X}$

$$\alpha_0(x_0) = 1. \tag{4.6}$$

Forward indukcija: Za $1 \leq i \leq T$ i $x_i \in \mathbb{X}$

$$\alpha_i(x_i) = \bigoplus_{x_{i-1}} u_{A_{i-1}}(x_{i-1}, x_i) \otimes \alpha_{i-1}(x_{i-1}). \tag{4.7}$$

Backward inicijalizacija: Za $x_T \in \mathbb{X}$

$$\beta_T(x_T) = 1. \tag{4.8}$$

Backward indukcija: Za $T \geq i \geq 1$ i $x_i \in \mathbb{X}$

$$\beta_t(x_t) = \bigoplus_{x_{t+1}} u_{A_{t+1}}(x_t, x_{t+1}) \otimes \beta_{t+1}(x_{t+1}). \tag{4.9}$$

Izračunavanje svih marginala: Za $1 \leq i \leq T$ i $x_{i-1}, x_i \in \mathbb{X}$

$$Z_{i-1:i}(x_{i-1}, x_i) = \bigoplus_{x_{-i-1:i}} u(x_{0:T}) = \alpha_{i-1}(x_{i-1}) \otimes u_{A_i}(x_{i-1}, x_i) \otimes \beta_i(x_i). \tag{4.10}$$

Ukoliko se rešava normalizacioni problem, izvršava se samo *forward* prolaz za $1 \leq i \leq T$ i normalizacioni korak.

Normalizacija: Za $x_T \in \mathbb{X}$

$$Z = \bigoplus_{x_T} \alpha_T(x_T). \tag{4.11}$$

4.2 Problem izračunavanja gradijenta uslovnih slučajnih polja

Neka su $X_{0:T}$ i $O_{0:T}$ višedimenzione slučajne promenljive koje uzimaju vrednosti iz skupova $\mathbb{X}$ i $\mathbb{O}$, respektivno. Promenljivu $X_{0:T}$ ćemo zvati *sekvenca stanja*, a promenljivu $O_{0:T}$ *sekvenca opservacija*. Uslovna slučajna polja (*conditional random fieldsm CRF*) modeluju uslovnu verovatnoću p promenljive $X_{0:T}$ pri uslovu da sekvenca $O_{0:T}$ uzima vrednost $o_{0:T}$ na sledeći način

$$p(x_{0:T}|o_{0:T};\boldsymbol{\theta}) = \frac{1}{Z(o_{0:T};\boldsymbol{\theta})}\prod_{i=1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\, \boldsymbol{f}(x_{i-1},x_i,o_{0:T},i)\rangle}, \tag{4.12}$$

pri čemu simbol $\langle\cdot,\cdot\rangle$ označava skalarni proizvod izmedju d-dimenzionalnog vektora parametara

$$\boldsymbol{\theta} = \left[\theta_1,\ldots,\theta_d\right] \in \mathbb{R}^d, \tag{4.13}$$

i i-te *feature*-funkcije, $\boldsymbol{f} : \mathbb{X}^2 \times \mathcal{O}^T \times \{1,\ldots,T\} \to \mathbb{R}^d$,

$$\boldsymbol{f}(x_{i-1},x_i,o_{0:T},i) = \left[f_1(x_{i-1},x_i,o_{0:T},i),\ldots,f_d(x_{i-1},x_i,o_{0:T},i)\right]. \tag{4.14}$$

Normalizacioni faktor

$$Z(o_{0:T};\boldsymbol{\theta}) = \sum_{x_{0:T}}\prod_{i=1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\, \boldsymbol{f}(x_{i-1},x_i,o_{0:T},i)\rangle} \tag{4.15}$$

naziva se particiona funkcija.

Cilj *CRF* treninga je pravljenje modela (4.12) na osnovu skupa podataka $\{(o_{0:T}^{(l)}, x_{0:T}^{(l)}\}_{l=1}^{L}$. Standardna metoda je maksimiziranje logaritma verodostojnosti raspodele (4.12)

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{l=1}^{L} \ln p(x_{0:T}^{(l)}|o_{0:T}^{(l)};\boldsymbol{\theta}), \tag{4.16}$$

po vektoru parametara $\boldsymbol{\theta}$ za dat skup *feature*-funkcija $\boldsymbol{f}(x_{i-1},x_i,o_{0:T},i)$. Maksimum se može naći uz pomoć neke od gradijentnih metoda [28], [51], [76], [79], [84], koje zahtevaju izračunavanje gradijenta $\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta})$. Gradijent se, na osnovu (4.12) i (4.16), može izraziti kao

$$\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}) = \sum_{l=1}^{L}\sum_{i=1}^{T^{(l)}} \boldsymbol{f}(x_{i-1}^{(l)},x_i^{(l)},o_{0:T}^{(l)},i) - \sum_{l=1}^{L}\frac{\nabla_{\boldsymbol{\theta}}Z(o_{0:T};\boldsymbol{\theta})}{Z(o_{0:T};\boldsymbol{\theta})}, \tag{4.17}$$

gde $T^{(l)}$ označava dužinu l-te opservacione sekvence. Dva glavna problema u izračunavanju izraza(4.17) su izračunavanje particione funkcije date sa (4.15) i njenog gradijenta

$$\nabla_{\boldsymbol{\theta}}Z(o_{0:T};\boldsymbol{\theta}) = [\nabla_{\theta_1}Z(o_{0:T};\boldsymbol{\theta}),\ldots,\nabla_{\theta_d}Z(o_{0:T};\boldsymbol{\theta})], \tag{4.18}$$

gde $\nabla_{\theta_m}Z(o_{0:T};\boldsymbol{\theta})$ označava m-ti parcijalni izvod. Gradijent se može dobiti iz (4.15), posle korišćenja Lajbnicovog pravila za izvod proizvoda

$$\nabla_{\theta}Z(o_{0:T};\boldsymbol{\theta}) = \sum_{x_{0:T}}\prod_{i=1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\, \boldsymbol{f}(x_{i-1},x_i,o_{0:T},i)\rangle}\sum_{k=1}^{T}\boldsymbol{f}(x_{k-1},x_k,o_{0:T},k). \tag{4.19}$$

Standardna metoda za izračunavanje particione funkcije i njenog gradijenta [51] bazira se na *FB* algoritmu, koji razmatramo u narednom odeljku.

4.2.1 Izračunavanje particione funkcije i njenog gradijenta primenom *FB* algoritma nad *sum-product* poluprstenom

Particiona funkcija (4.15) može se dobiti kao rešenje normalizacionog problema (4.11) za faktorizaciju

$$\prod_{i=1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\, f(x_{i-1}, x_i, o_{0:T}, i)\rangle}, \tag{4.20}$$

kao

$$Z(o_{0:T}; \boldsymbol{\theta}) = \sum_{x_{0:T}} \prod_{i=1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\, f(x_{i-1}, x_i, o_{0:T}, i)\rangle}. \tag{4.21}$$

Gradijent se može izračunati rešavanjem marginalizacionog problema (4.10) u *sum-product* poluprstenu. Najpre, promenom redosleda u (4.19) razbijamo sumu nad $x_{0:T}$, na sume $x_{k-1:k}$ i $x_{-k-1:k}$, transformišući (4.19) u

$$\nabla_\theta Z(o_{0:T}; \boldsymbol{\theta}) = \sum_{k=1}^{T} \sum_{x_{k-1:k}} \Big(\sum_{x_{-k-1:k}} \prod_{i=1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\, f(x_{i-1}, x_i, o_{0:T}, i)\rangle} \Big) \cdot \boldsymbol{f}(x_{k-1}, x_k, o_{0:T}, k). \tag{4.22}$$

Problem izračunavanja izraza

$$\sum_{x_{-k-1:k}} \prod_{i=1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\, f(x_{i-1}, x_i, o_{0:T}, i)\rangle} \tag{4.23}$$

može se rešiti kao marginalizacioni problem nad *sum-product* poluprstenom.

FB algoritam nad *sum-product* poluprstenom je numerički nestabilan pošto eksponencijalni članovi mogu da ispadnu izvan opsega računarske preciznosti, pa se umesto njega obično koristi njegova stabilna varijanta koja, funkcioniše kao *FB* algoritam nad log-domen *sum-product* poluprstenom.

4.2.2 Izračunavanje particione funkcije i njenog gradijenta primenom *FB* algoritma nad log-domen *sum-product* poluprstenom

Definicija 4 *Log-domen sum-product poluprsten je petorka* $\left(\mathbb{R}^*, \oplus, \otimes, -\infty, 0\right)$ *gde je* $\mathbb{R}^*$ *prošireni skup realnih brojeva a operacije (log-sabiranje,* $\oplus$*, i log-množenje,* $\otimes$*) su definisane sa*

$$a \oplus b = \ln(\mathrm{e}^a + \mathrm{e}^b), \tag{4.24}$$

$$a \otimes b = a + b, \tag{4.25}$$

za svako $a, b \in \mathbb{R}$.

Sledeće leme mogu se dokazati indukcijom direktno iz definicije log-domen *sum-product* poluprstena.

Lema 4.2.1 *Neka su* $a_i \in \mathbb{R}$ *za svako* $1 \leq i \leq T$*. Tada, u log-domen sum-product poluprstenu važe sledeće jednakosti*

$$\ln\Big(\sum_{i=1}^{T} a_i\Big) = \bigoplus_{i=1}^{T} \ln a_i, \quad \ln\Big(\prod_{i=1}^{T} a_i\Big) = \bigotimes_{i=1}^{T} \ln a_i. \tag{4.26}$$

U log-domenu, faktori imaju oblik

$$u_i(x_{i-1}, x_i) = \langle \boldsymbol{\theta},\ \boldsymbol{f}(x_{i-1}, x_i, o_{0:T}, i)\rangle, \tag{4.27}$$

za $i = 1, \ldots, T$. Saglasno lemi 4.2.1 i izrazu (4.4), *forward* vektor u log-domenu je logaritam *forward* vektora u *sum-product* poluprstenu:

$$\alpha_i(x_i) = \bigoplus_{x_{0:i-1}} \bigotimes_{t=1}^{i} u_t(x_t, x_{t-1}) = \ln\Big(\sum_{x_{0:i-1}} \prod_{t=1}^{i} \mathrm{e}^{\langle \boldsymbol{\theta},\, f(x_{t-1}, x_t, o_{0:T}, t)\rangle} \Big). \tag{4.28}$$

Forward promenljiva α_0 inicijalizovana je na 0 što predstavlja jedinični element za $\otimes$

$$\alpha_0(x_0) = 0, \tag{4.29}$$

i izračunava se na osnovu

$$\alpha_i(x_i) = \bigoplus_{x_{i-1}} \big(u_i(x_{i-1}, x_i) + \alpha_{i-1}(x_{i-1}) \big). \tag{4.30}$$

Slično, na osnovu leme 4.2.1 i izraza (4.5), *backward* promenljiva u log-domenu je logaritam *backward* promenljive u *sum-product* poluprstenu

$$\beta_i(x_i) = \bigoplus_{x_{i+1:T}} \bigotimes_{t=i+1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\, f(x_{t-1}, x_t, o_{0:T}, t)\rangle} = \ln \sum_{x_{i+1:T}} \prod_{t=i+1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\, f(x_{t-1}, x_t, o_{0:T}, t)\rangle}, \tag{4.31}$$

inicijalizovana je na

$$\beta_T(x_T) = 0, \tag{4.32}$$

i izračunava se pomoću

$$\beta_i(x_i) = \bigoplus_{x_{i+1}} \big(u_{i+1}(x_i, x_{i+1}) + \beta_{i+1}(x_{i+1}) \big). \tag{4.33}$$

Ukoliko se sabiranje u log-domenu izvršava na osnovu definicije, $a \oplus b = \ln(\mathrm{e}^a + \mathrm{e}^b)$, gubi se numerička preciznost prilikom izračunavanja e^a i e^b. Medjutim, kao što je napomenuto u ([79]), $\oplus$ se može izvršiti na numerički stabilan način

$$a \oplus b = a + \ln\big(1 + \mathrm{e}^{(b-a)}\big) = b + \ln\big(1 + \mathrm{e}^{(a-b)}\big), \tag{4.34}$$

pri čemu se za izračunavanje bira izraz sa manjim eksponentom.

Logaritam normalizacione funkcije (4.21) je na osnovu leme 4.2.1

$$\ln Z(o_{0:T}; \boldsymbol{\theta}) = \ln \sum_{x_{0:T}} \prod_{i=1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\, f(x_{i-1}, x_i, o_{0:T}, i)\rangle} = \bigoplus_{x_{0:T}} \bigotimes_{i=1}^{T} u_i(x_i, x_{i-1}), \tag{4.35}$$

i može se izračunati uz pomoć rešenja za normalizacioni problem u log-domenu pomoću *forward* algoritma, saglasno formulama (4.11)

$$\ln Z(o_{0:T}; \boldsymbol{\theta}) = \bigoplus_{x_T} \alpha_T(x_T). \tag{4.36}$$

Na osnovu leme 4.2.1, marginalne vrednosti (4.23) u log-domenu imaju oblik

$$v_k(x_{k-1}, x_k) = \ln \sum_{x_{-k-1:k}} \prod_{i=1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\, \boldsymbol{f}(x_{i-1}, x_i, o_{0:T}, i)\rangle} = \bigoplus_{x_{-k-1:k}} \bigotimes_{i=1}^{T} u_i(x_i, x_{i-1}). \tag{4.37}$$

Marginalne vrednosti mogu biti efikasno izračunate na osnovu rešenja za marginalizacioni problem (4.10)

$$v_k(x_{k-1}, x_k) = \alpha_{k-1}(x_{k-1}) \otimes u_k(x_{k-1}, x_k) \otimes \beta_k(x_k), \tag{4.38}$$

pri čemu se $\alpha_{k-1}(x_{k-1})$ i $\beta_k(x_k)$ izračunavaju uz pomoć *FB* algoritma nad log-domen *sum-product* poluprstenom na osnovu jednakosti (4.29)-(4.33). Logaritmovanjem m-te komponente gradijenta u izrazu (4.22), dobijamo

$$\ln \nabla_{\theta_m} Z(o_{0:T}; \boldsymbol{\theta}) = \bigoplus_{k=1}^{T} \bigoplus_{x_{\{k-1,k\}}} v_k(x_{k-1}, x_k) \otimes \ln f_m(x_{k-1}, x_k, o_{0:T}, k), \tag{4.39}$$

za $m = 1, \ldots, d$. Konačno, količnik izmedju gradijenta i particione funkcije može se izračunati na osnovu

$$\frac{\nabla_\theta Z(o_{0:T}; \boldsymbol{\theta})}{Z(o_{0:T}; \boldsymbol{\theta})} = \mathrm{e}^{\ln \nabla_\theta Z(o_{0:T}; \boldsymbol{\theta}) - \ln Z(o_{0:T}; \boldsymbol{\theta})}. \tag{4.40}$$

Algoritam sledi.

Inicijalizacija matrica: Za $1 \le i \le T$, $x_{i-1}, x_i \in \mathbb{X}$

$$u_i(x_{i-1}, x_i) = \langle \boldsymbol{\theta},\ \boldsymbol{f}(x_{i-1}, x_i, o_{0:T}, i)\rangle, \tag{4.41}$$

za $i = 1, \cdots, T$.

Forward inicijalizacija: Za $x_0 \in \mathbb{X}$

$$\alpha_0(x_0) = 0. \tag{4.42}$$

Forward indukcija: Za $1 \le i \le T$, $x_i \in \mathbb{X}$

$$\alpha_i(x_i) = \bigoplus_{x_{i-1}} \big(u_i(x_{i-1}, x_i) + \alpha_{i-1}(x_{i-1})\big). \tag{4.43}$$

Backward inicijalizacija: Za $x_T \in \mathbb{X}$

$$\beta_T(x_T) = 0. \tag{4.44}$$

Backward indukcija: Za $T - 1 \ge i \ge 0$, $x_i \in \mathbb{X}$

$$\beta_i(x_i) = \bigoplus_{x_{i+1}} \big(u_{i+1}(x_i, x_{i+1}) + \beta_{i+1}(x_{i+1})\big). \tag{4.45}$$

Izračunavanje logaritma particione funkcije: Za $1 \le m \le d$

$$\ln Z(o_{0:T}; \boldsymbol{\theta}) = \bigoplus_{x_T} \alpha_T(x_T). \tag{4.46}$$

Izračunavanje svih marginala: Za $1 \le i \le T$ i $x_{i-1}, x_i \in \mathbb{X}$

$$v_k(x_{k-1}, x_k) = \alpha_{k-1}(x_{k-1}) \otimes u_k(x_{k-1}, x_k) \otimes \beta_k(x_k). \tag{4.47}$$

Izračunavanje logaritma particione funkcije: Za $1 \le m \le d$

$$\ln \nabla_{\theta_m} Z(o_{0:T}; \boldsymbol{\theta}) = \bigoplus_{k=1}^{T} \bigoplus_{x_{\{k-1,k\}}} v_k(x_{k-1}, x_k) \otimes \ln f_m(x_{k-1}, x_k, o_{0:T}, k) \tag{4.48}$$

Vremenska i memorijska kompleksnost *FB* algoritma nad log-domen *sum-product* poluprstenom

Kao i do sada, vremenska kompleksnost definiše se kao asimptotski broj operacija potrebnih za izvršavanje algoritma. U analizi uzimamo samo najzahtevniju operaciju, log-sabiranje, koja zahteva izračunavanje logaritma i eksponenta. Memorijska kompleksnost definiše se kao broj realnih brojeva potrebnih za čuvanje promenljivih tokom izvršenja algoritma.

U primenama, *feature*-funkcije preslikavaju ulazni prostor u skup retko popunjenih vektora, koji imaju nenula vrednosti jedino na pozicijama

$$\mathcal{A}_k(x_{k-1}, x_k) = \left\{ m \;\text{ ako je }\; f_m(x_{k-1}, x_k, o_{0:T}, k) \neq 0 \right\}, \tag{4.49}$$

tako da je moguće smanjiti kompleksnost izračunavanja obavljanjem operacija samo za elemente različite od nule. U našoj analizi koristićemo prosečan broj elemenata različit od nule, definisan kao

$$A = \frac{\sum_{k=1}^{T} \sum_{x_{k-1}, x_k} |\mathcal{A}_k(x_{k-1}, x_k)|}{|\mathbb{X}|^2 T}. \tag{4.50}$$

Najzahtevniji deo za izračunavanje u algoritmu je poslednja faza, izračunavanje logaritma particione funkcije, u kojoj se izvršava $\mathcal{O}(|\mathbb{X}|^2 TA)$ log-sabiranja. Memorijska kompleksnost algoritma je $\mathcal{O}(|\mathbb{X}|^2 T + d)$, i odredjena je memorijskim prostorom potrebnim za izračunavanje matrica u_i. Zavisnost memorijske kompleksnosti od dužine sekvence može značajno smanjiti računarske performanse, ukoliko se algoritam koristi za dugačke sekvence, jer onda dolazi do prepisivanja sa interne memorije na hard disk, što potvrdjuju eksperimenti sprovedeni u radu [35]. U narednom poglavlju pokazaćemo kako se gradijent može izračunati primenom *EMP* algoritma uz nešto veću vremensku kompleksnost, $\mathcal{O}(|\mathbb{X}|^2 (d+A)T)$, zadržavajući memorijsku kompleksnost nezavisnu od dužine sekvence.

4.3 Izračunavanje particione funkcije i njenog gradijenta primenom *EMP* algoritma

U ovom poglavlju ćemo razviti memorijski efikasan algoritam za računanje *CRF* gradijenta, koji predstavlja numerički stabilnu verzijju *EMP* algoritma za izračunavanje *CRF* gradijenta, a funkcioniše kao *forward* algoritam nad log-domen poluprstenom očekivanja.

4.3.1 Izračunavanje particione funkcije i njenog gradijenta primenom standardnog *EMP* algoritma

Sledeća lema predstavlja specijalan slučaj leme 2.3.1 za lance.

Lema 4.3.1 *Neka je* $(z_i, \boldsymbol{h}_i) \in \mathbb{R} \times \mathbb{R}^d$ *za* $1 \leq i \leq T$. *Tada, važi sledeća jednakost u poluprstenu očekivanja*

$$\bigoplus_{i=1}^{T} (z_i, z_i \boldsymbol{h}_i) = \left(\sum_{i=1}^{T} z_i \,,\, \sum_{i=1}^{T} \boldsymbol{h}_i \right). \tag{4.51}$$

Ukoliko parovi imaju formu $(z, z\boldsymbol{h})$, množenje se izvršava kao

$$(z_1, \boldsymbol{h}_1) \odot (z_2, \boldsymbol{h}_2) = (z_1 z_2,\ z_1 z_2(\boldsymbol{h}_1 + \boldsymbol{h}_2), \tag{4.52}$$

što se generalizuje sledećom lemom.

Lema 4.3.2 *Neka je* $(z_i, z_i\boldsymbol{h}_i) \in \mathbb{R} \times \mathbb{R}^d$ *za* $1 \leq i \leq T$. *Tada, važi sledeća jednakost*

$$\bigodot_{i=1}^{T} (z_i, z_i\boldsymbol{h}_i) = \left(\prod_{i=1}^{T} z_i\ ,\ \prod_{i=1}^{T} z_i\ \cdot \sum_{j=1}^{T} \boldsymbol{h}_j \right). \tag{4.53}$$

Saglasno izrazu 4.3.2, ukoliko faktori imaju oblik

$$\begin{aligned} u_i(x_{i-1}, x_i) = \big(&\mathrm{e}^{\langle \boldsymbol{\theta},\ f(x_{i-1}, x_i, o_{0:T}, i)\rangle}, \\ &\mathrm{e}^{\langle \boldsymbol{\theta},\ f(x_{i-1}, x_i, o_{0:T}, i)\rangle} \cdot \boldsymbol{f}(x_{i-1}, x_i, o_{0:T}, i)\big), \end{aligned} \tag{4.54}$$

za $i = 1, \cdots, T$, njihov proizvod je, saglasno lemi 4.3.2

$$\begin{aligned} \bigotimes_{i=1}^{T} u_i(x_{i-1}, x_i) = \big(&\prod_{i=1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\ f(x_{i-1}, x_i, o_{0:T}, i)\rangle}, \\ &\prod_{i=1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\ f(x_{i-1}, x_i, o_{0:T}, i)\rangle} \cdot \sum_{j=1}^{T} \boldsymbol{f}(x_{j-1}, x_j, o_{0:T}, j)\big). \end{aligned} \tag{4.55}$$

Primenom 4.3.1, i izraza (4.55), možemo dobiti particionu funkciju (4.15)

$$Z(o_{0:T}; \boldsymbol{\theta}) = \sum_{x_{0:T}} \prod_{i=1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\ f(x_{i-1}, x_i, o_{0:T}, i)\rangle} \tag{4.56}$$

i njen gradient (4.19)

$$\nabla_{\theta_m} Z(o_{0:T}; \boldsymbol{\theta}) = \sum_{x_{0:T}} \prod_{i=1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\ f(x_{i-1}, x_i, o_{0:T}, i)\rangle} \cdot \sum_{j=1}^{T} \boldsymbol{f}(x_{j-1}, x_j, o_{0:T}, j), \tag{4.57}$$

kao z i h-komponente sume

$$\bigoplus_{x_{0:T}} \bigotimes_{i=1}^{T} u_i(x_{i-1}, x_i) = \big(\ Z(o_{0:T}; \boldsymbol{\theta}),\ \nabla_{\theta} Z(o_{0:T}; \boldsymbol{\theta})\big). \tag{4.58}$$

Izraz (4.58) se može izračunati kao normalizacioni problem (4.11) primenom *forward* algoritma nad poluprstenom očekivanja. Kao što smo pomenuli, z-komponente zbira i proizvoda u poluprstenu očekivanja su realno množenje i sabiranje z-komponenata. Shodno tome, z-komponente *forward* vektora biće iste kao *forward* vektori u *sum – product* poluprstenu, a njihovo izračunavanje je numerički nestabilno. U narednom odeljku razvijamo numerički stabilan *forward* algoritam koji se izvršava u log-domen poluprstenu očekivanja.

4.3.2 Izračunavanje particione funkcije i njenog gradijenta primenom log-domen *EMP* algoritma

Log-domen poluprsten očekivanja je kombinacija log-domen *sum-product* poluprstena i poluprstena očekivanja. Može se dobiti ako se realno sabiranje i množenje u definiciji poluprstena očekivanja zamene sabiranjem i množenjem u log-domenu. Pre nego što definišemo log-domen poluprsten očekivanja, uvodimo sledeću notaciju. Prvo, setimo se da je log-domen sabiranje i množenje definisano sa

$$a \oplus b = \ln(\mathrm{e}^a + \mathrm{e}^b), \tag{4.59}$$

$$a \otimes b = a + b. \tag{4.60}$$

Log-proizvod izmedju skalara $z \in \mathbb{R}$ i vektora $\boldsymbol{h} = (h[1], \ldots, h[d]) \in \mathbb{R}^d$ definiše se kao vektor $z \otimes h$

$$z \otimes \boldsymbol{h} = z \otimes (h[1], \ldots, h[d]) = (z \otimes h[1], \ldots, z \otimes h[d]). \tag{4.61}$$

Logaritam vektora $[h_1, \ldots, h_d] \in \mathbb{R}^d$ definisan je kao

$$\ln[h_1, \ldots, h_d] = [\ln h_1, \ldots, \ln h_d]. \tag{4.62}$$

Vektor $-\boldsymbol{\infty}$ definiše se kao vektor čije su sve koordinate $-\infty$.

Definicija 5 *Log-domen poluprsten očekivanja reda d je petorka*

$$\big(\mathbb{R} \times \mathbb{R}^d,\ \oplus,\ \odot,\ (-\infty,\ -\boldsymbol{\infty}), (0,\ -\boldsymbol{\infty}) \big),$$

pri čemu su operacije $\oplus$ *i* $\odot$ *definisane sa*

$$(z_1, \boldsymbol{h}_1) \oplus (z_2, \boldsymbol{h}_2) = \big(z_1 \oplus z_2,\ \boldsymbol{h}_1 \oplus \boldsymbol{h}_2 \big), \tag{4.63}$$

$$(z_1, \boldsymbol{h}_1) \odot (z_2, \boldsymbol{h}_2) = \big(z_1 \otimes z_2,\ (z_1 \otimes \boldsymbol{h}_2)\ \oplus\ (z_2 \otimes \boldsymbol{h}_1) \big), \tag{4.64}$$

za svako $(z_1, \boldsymbol{h}_1)$, $(z_2, \boldsymbol{h}_2)$ *iz* $\mathbb{R} \times \mathbb{R}^d$. *Slično kao i kod poluprstena očekivanja, prva komponenta uredjenog para naziva se z-komponenta, dok se druga naziva h-komponenta.*

Sledeća lema je log-domen verzija leme 4.3.1.

Lema 4.3.3 *Neka je* $(z_i, z_i \boldsymbol{h}_i) \in \mathbb{R} \times \mathbb{R}^d$ *za svako* $1 \le i \le T$. *Važi sledeća jednakost u log-domen poluprstenu očekivanja*

$$\bigoplus_{i=1}^{T} (z_i, \boldsymbol{h}_i) = \Big(\bigoplus_{i=1}^{T} z_i\ ,\ \bigoplus_{i=1}^{T} \boldsymbol{h}_i \Big), \tag{4.65}$$

gde je

$$\bigoplus_{i=1}^{T} a_i = \ln \Big(\sum_{i=1}^{T} \boldsymbol{e}^{a_i} \Big). \tag{4.66}$$

Slično kao u poluprstenu očekivanja, ako parovi imaju oblik $(z,\ z \otimes \boldsymbol{h})$, množenje se svodi na

$$(z_1, z_1 \otimes \boldsymbol{h}_1) \otimes (z_2, z_2 \otimes \boldsymbol{h}_2) = (\ z_1 \otimes z_2,\ z_1 \otimes z_2 \otimes (\boldsymbol{h}_1 \oplus \boldsymbol{h}_2)). \tag{4.67}$$

Sledeća lema je log-domen verzija leme 4.3.2.

Lema 4.3.4 *Neka je* $(z_i,\ z_i \otimes \boldsymbol{h}_i) \in \mathbb{R} \times \mathbb{R}^d$ *za svako* $1 \leq i \leq T$. *Tada važi sledeća jednakost u log-domen poluprstenu očekivanja*

$$\bigotimes_{i=1}^{T}(z_i,\ z_i \otimes \boldsymbol{h}_i) = \Big(\ \bigotimes_{i=1}^{T} z_i\ ,\ \bigotimes_{i=1}^{T} z_i\ \otimes \bigoplus_{j=1}^{T} \boldsymbol{h}_j\ \Big), \tag{4.68}$$

gde je

$$\bigotimes_{i=1}^{T} a_i = \sum_{i=1}^{T} a_i. \tag{4.69}$$

Neka je za $i = 1, \ldots, T$

$$\psi_i(x_i, x_{i-1}) = \langle \boldsymbol{\theta},\ \boldsymbol{f}(x_{i-1}, x_i, o_{0:T}, i) \rangle. \tag{4.70}$$

Tada se logaritam particione funkcije (4.15) može predstaviti kao

$$\ln Z(o_{0:T}; \boldsymbol{\theta}) = \bigoplus_{x_{0:T}} \bigotimes_{i=1}^{T} \psi_i(x_i, x_{i-1}). \tag{4.71}$$

Logaritam m-tog parcijalnog izvoda može se zapisati kao

$$\ln \nabla_{\theta_m} Z(o_{0:T}; \boldsymbol{\theta}) = \ln \Big(\sum_{x_{0:T}} \prod_{i=1}^{T} \mathrm{e}^{\langle \boldsymbol{\theta},\ f(x_{i-1}, x_i, o_{0:T}, i) \rangle} \sum_{k=1}^{T} f_m(x_{k-1}, x_k, o_{0:T}, k) \Big), \tag{4.72}$$

ili korišćenjem operacija u log-domen *sum-product* poluprstenu

$$\ln \nabla_{\theta} Z(o_{0:T}; \boldsymbol{\theta}) = \bigoplus_{x_{0:T}} \bigotimes_{i=1}^{T} \psi_i(x_i, x_{i-1}) \otimes \bigoplus_{k=1}^{T} \ln \boldsymbol{f}(x_{k-1}, x_k, o_{0:T}, k). \tag{4.73}$$

Ako faktori imaju oblik

$$u_i(x_{i-1}, x_i) = \big(\psi_i(x_i, x_{i-1}),\ \ \psi_i(x_i, x_{i-1}) \otimes \ln \boldsymbol{f}(x_{i-1}, x_i, o_{0:T}, i)\big), \tag{4.74}$$

za $i = 1, \cdots, T$, njihov proizvod je, saglasno lemi 4.3.4

$$\bigodot_{i=1}^{T} u_i(x_{i-1}, x_i) = \big(\bigotimes_{i=1}^{T} \psi_i(x_i, x_{i-1}), \bigotimes_{i=1}^{T} \psi_i(x_i, x_{i-1}) \otimes \bigoplus_{j=1}^{T} \ln \boldsymbol{f}(x_{j-1}, x_j, o_{0:T}, j)\big). \tag{4.75}$$

Na osnovu leme (4.3.3) za sabiranje u log-domen poluprstenu očekivanja, suma uredejnih parova je uredjeni par suma, tako da se particiona funkcija i njen gradijent mogu dobiti kao z i h-komponenta

$$\bigoplus_{x_{0:T}} \bigodot_{i=1}^{T} u_i(x_{i-1}, x_i) = \Big(\ \ln Z(o_{0:T}; \boldsymbol{\theta}),\ \ln \nabla_{\theta} Z(o_{0:T}; \boldsymbol{\theta})\Big). \tag{4.76}$$

Izraz (4.76) može se izračunati kao normalizacioni problem (4.11) uz pomoć *forward* algoritma nad log-domen poluprstenom očekivanja (*log-domen EMP algoritam*). *Forward* algoritam se inicijalizuje na jedinični element za množenje u log-domen poluprstenu očekivanja

$$\alpha_0(x_0) = (0, -\infty), \tag{4.77}$$

za svako $x_0 \in \mathbb{X}$. Nakon toga, izračunavamo *forward* promenljive pomoću formule

$$\alpha_i(x_i) = \bigoplus_{x_{i-1}} u_i(x_{i-1}, x_i) \odot \alpha_{i-1}(x_{i-1}), \tag{4.78}$$

gde je su faktori dati sa (4.74).

Na osnovu pravila za sabiranje i množenje u poluprstenu očekivanja z i h-komponente izraza (4.78) su

$$\alpha_i^{(z)}(x_i) = \bigoplus_{x_{i-1}} \psi_i(x_i, x_{i-1}) \otimes \alpha_{i-1}^{(z)}(x_{i-1}), \tag{4.79}$$

$$\begin{aligned}\boldsymbol{\alpha}_i^{(h)}(x_i) = &\bigoplus_{x_{i-1}} \psi_i(x_i, x_{i-1}) \otimes \boldsymbol{\alpha}_{i-1}^{(h)}(x_{i-1}) \oplus \\ &\bigoplus_{x_{i-1}} \psi_i(x_i, x_{i-1}) \otimes \alpha_{i-1}^{(z)}(x_{i-1}) \otimes \ln \boldsymbol{f}(x_{i-1}, x_i, o_{0:T}, i),\end{aligned} \tag{4.80}$$

za svako $x_i \in \mathbb{X}$, $i = 1, \ldots, T$. Konačno, normalizacioni problem može se rešiti sumiranjem

$$\bigoplus_{x_{0:T}} \bigodot_{i=1}^{T} u_i(x_{i-1}, x_i) = \bigoplus_{x_T} \alpha_T(x_T), \tag{4.81}$$

čija je z-komponenta logaritam particione funkcije, a h-komponenta logaritam gradijenta

$$\ln Z(o_{0:T}; \boldsymbol{\theta}) = \bigoplus_{x_T} \alpha_T^{(z)}(x_T), \quad \ln \nabla_\theta Z(o_{0:T}; \boldsymbol{\theta}) = \bigoplus_{x_T} \boldsymbol{\alpha}_T^{(h)}(x_T). \tag{4.82}$$

Dakle, algoritam se sastoji iz dve faze: 1) *forward* algoritam, u kome se *forward* promenljiva inicijalizuje na osnovu (4.77) i induktivno izračunavaju pomoću (4.79)-(4.80), pri čemu se matrice u_i izračunavaju u svakom koraku i 2) *terminacija*, u kojoj se izvršava finalno sumiranje *forward* vektora na osnovu (4.82) i izračunavaju se logaritmi particione funkcije i gradijenta.

Forward inicijalizacija: Za $x_0 \in \mathbb{X}$

$$\alpha_0^{(z)}(x_0) = 0, \tag{4.83}$$

$$\boldsymbol{\alpha}_0^{(h)}(x_0) = 0. \tag{4.84}$$

Forward indukcija: Za $1 \leq i \leq T$, $x_{i-1}, x_i \in \mathbb{X}$

$$u_i(x_{i-1}, x_i) = \langle \boldsymbol{\theta},\ \boldsymbol{f}(x_{i-1}, x_i, o_{0:T}, i) \rangle, \quad \text{za } x_i \in \mathbb{X}, \tag{4.85}$$

$$\alpha_i^{(z)}(x_i) = \bigoplus_{x_{i-1}} \psi_i(x_i, x_{i-1}) \otimes \alpha_{i-1}^{(z)}(x_{i-1}), \tag{4.86}$$

$$\begin{aligned}\boldsymbol{\alpha}_i^{(h)}(x_i) = &\bigoplus_{x_{i-1}} \psi_i(x_i, x_{i-1}) \otimes \boldsymbol{\alpha}_{i-1}^{(h)}(x_{i-1}) \oplus \\ &\bigoplus_{x_{i-1}} \psi_i(x_i, x_{i-1}) \otimes \alpha_{i-1}^{(z)}(x_{i-1}) \otimes \ln \boldsymbol{f}(x_{i-1}, x_i, o_{0:T}, i).\end{aligned} \tag{4.87}$$

Izračunavanje logaritma particione funkcije i njenog gradijenta

$$\begin{aligned}\ln Z(o_{0:T}; \boldsymbol{\theta}) &= \bigoplus_{x_T} \alpha_T^{(z)}(x_T), \\ \ln \nabla_\theta Z(o_{0:T}; \boldsymbol{\theta}) &= \bigoplus_{x_T} \boldsymbol{\alpha}_T^{(h)}(x_T).\end{aligned} \tag{4.88}$$

FB algoritam zahteva izračunavanje i čuvanje svih *forward* i *backward* promenljivih, sve do izračunavanja particione funkcije i njenog gradijenta u terminacionom koraku. Kod *EMP* algoritma, izračunavanje se završava kada se izračuna poslednja *forward* promenljiva korišćenjem formula (4.79) i (4.80). Ovo se može realizovati u fiksnom memorijskom prostoru veličine nezavisne od dužine sekvence, pošto promenljive $\alpha_{i-1}^{(z)}$, $\boldsymbol{\alpha}_{i-1}^{(h)}$ i matrice u_i treba da se izračunaju samo jednom u $i-1$-voj iteraciji, a pošto se iskoriste za izračunavanje promenljivih $\alpha_i^{(z)}$ i $\boldsymbol{\alpha}_i^{(h)}$ u i-toj iteraciji, mogu biti izbrisani.

U odnosu na *FB* algoritam, kome je potrebno $\mathcal{O}(|\mathbb{X}|^2T+d)$ memorije, *EMP* ima memorijsku kompleksnost $\mathcal{O}(|\mathbb{X}|^2+|\mathbb{X}|\cdot d$, nezavisnu od dužine sekvence T, sto nije slučaj kod *FB* algoritma. S druge strane, najzahtevniji deo izračunavanja kod *EMP* algoritma obavlja se prilikom *forward* indukcije za h-komponentu, pri čemu se izvršava $\mathcal{O}(|\mathbb{X}|^2T(A+d))$ log-sabiranja. Dakle, u odnosu na *FB* algoritam koji zahteva $\mathcal{O}(|\mathbb{X}|^2TA)$ log-sabiranja, imamo uvećanje vremenske kompleksnosti za $\mathcal{O}(|\mathbb{X}|^2TA)$, što predstavlja cenu smanjenja memorijskih zahteva.

Setimo se da A u izrazu za vremensku složenost *FB* algoritma, $\mathcal{O}(|\mathbb{X}|^2TA)$, predstavlja prosečan broj nenultih elemenata u vektorima $\boldsymbol{f}(x_{k-1}, x_k, o_{0:T}, k)$, za koji je obično $A << d$. Medjutim, ovo nije slučaj kod uslovno obučavanog skrivenog Markovljevog modela [25]. U ovom slučaju, vremenska kompleksnost *EMP*-a postaje bliža *FB* algoritmu. Sa praktične strane, *EMP* ima prednost kada se koriste dugačke sekvence, pošto u tom slučaju *FB* algoritam mora da koristi eksternu memoriju, kao što je ekserimentalno pokazano u [35].

Glava 5

Izračunavanje kros-momenata na faktor-grafovima

U ovoj glavi algoritmi za izračunavanje momenata prvog reda uopšteni su za generalni slučaj kros-momenata proizvoljnog reda, u slučaju kada *MGF* može biti predstavljena faktor-grafom bez ciklusa [37]. Algoritmi iz ove glave u osnovi predstavljaju *MP* algoritam nad poluprstenom stepenih redova, a kros-momenti se mogu dobiti kao parcijalni izvodi *MGF*-a odgovarajućeg reda, ili, ekvivalentno, kao umnošci koeficijenata u Tejlorovom razvoju za *MGF*. Kada se *MP* algoritam kombinuje sa poluprstenom stepenih redova, poruke su stepeni redovi, i mogu se predstaviti kao beskonačno-dimenzione n-torke Tejlorovih koeficijenta, ili, ekvivalentno, kao beskonačno-dimenzione n-torke parcijalnih izvoda u nuli. Ovakva propagaciona šema naziva se *MGFMP* (*moment generating function message passing*) i obradjena je u poglavlju 5.1.

U praksi nas interesuju kros-momenti zaključno sa odredjenim redom $(\nu_1, \ldots, \nu_d)$, tako da je dovoljno čuvati informaciju o Tejlorovim koeficijentima (parcijalnim izvodima), zaključno sa redom $(\nu_1, \ldots, \nu_d)$; dakle, koeficijenti uz članove $t_1^{\alpha_1} \cdots t_d^{\alpha_d}$, gde je $\alpha_i \leq \nu_i$, za svako i. Razmatramo dve mogućnosti:

1. Prva mogućnost - u kojoj su poruke predstavljene n-torkama Tejlorovih koeficijenata zaključno sa redom $(\nu_1, \ldots, \nu_d)$. Ova mogućnost može se realizovati kao *MP* algoritam nad poluprstenom polinoma (*polynomial semiring message passing, PSMP*). *PSMP* algoritam razmatran je u poglavlju 5.2 i predstavlja generalizaciju algoritma za izračunavanje skalarnih momenata nad faktor-grafom bez ciklusa, koji su razvili *Cowell* i saradnici [15].

2. Druga mogućnost - u kojoj su poruke predstavljene n-torkama parcijalnih izvoda zaključno sa redom $(\nu_1, \ldots, \nu_d)$. Ova mogućnost može se realizovati kao *MP* algoritam nad binomnim poluprstenom (*binomial semiring message passing, BSMP*). *BSMP* algoritam razmatran je u poglavlju 5.3 i predstavlja generalizaciju algoritma za izračunavanje skalarnih momenata nad faktor-grafom sa strukturom lanca koji su razvili *Heim* i saradnici [30] (videti odeljak 5.5.2).

PSMP i *BSMP* se poklapaju kada se koriste za izračunavanje kros-momenata reda (1) i (1,1), i mogu se shvatiti kao generalizacija *EMP* algoritma iz odeljka 2.3.2, u slučaju reda (1), i kao generalizacija *MP* algoritma drugog reda, koji su razvili *Kulesza* i *Taskar* [50], u slučaju reda (1,1) (odeljak 5.5.1). Kao što ćemo videti u poglavlju 5.4, za kros-momente višeg reda oba algoritma imaju istu vremensku kompleksnost, ali za razliku od *PSMP*-a, *BSMP* ne zahteva deljenje. S druge strane, za *BSMP* se plaća dodatna cena u nešto uvećanoj

memorijskoj kompleksnosti.

5.1 Izračunavanje funkcije generatrise momenta uz pomoć *MP* algoritma

5.1.1 Kros-momenti i funkcija generatrise

Neka je, kao i do sada, $X_{1:T}$ višedimenziona slučajna promenljiva sa raspodelom $p_{X_{1:T}}$, i neka je $\boldsymbol{g} : \mathbb{X}^T \to \mathbb{R}^d$ funkcija promenljive $X_{1:T}$. Neka se još $p_{X_{1:T}}$ i $\boldsymbol{g}$ mogu predstaviti faktor-grafom bez ciklusa kao proizvod, odnosno zbir faktora $\phi_M : \mathbb{X}^{d(M)} \to \mathbb{R}$ i $\boldsymbol{g}_M : \mathbb{R}^{d(M)} \to \mathbb{R}^d$, kao

$$\mu_{p,\,\boldsymbol{g}}^{(\boldsymbol{\alpha})} = \sum_{x_{1:T}} p(x_{1:T}) \cdot \boldsymbol{g}(x_{1:T})^{\boldsymbol{\alpha}}. \tag{5.1}$$

MGF funkcije $\boldsymbol{g}$, u odnosu na f, je realna funkcija $M_{p,\boldsymbol{g}} : \mathbb{R}^d \to \mathbb{R}$ definisana sa

$$M_{p,\boldsymbol{g}}(\boldsymbol{t}) = \sum_{x_{1:T}} p(x_{1:T}) \cdot \boldsymbol{e}^{\,\boldsymbol{g}(x_{1:T})\cdot \boldsymbol{t}}, \tag{5.2}$$

za svako $\boldsymbol{t}$. Kros-moment reda $\boldsymbol{\alpha}$ može se izračunati kao parcijalni izvod *MGF*-a

$$\mu_{p,\,\boldsymbol{g}}^{(\boldsymbol{\alpha})} = \mathcal{D}^{(\boldsymbol{\alpha})} \big\{\, M_{p,\boldsymbol{g}}(\boldsymbol{t}) \,\big\}_{\boldsymbol{t}=0}. \tag{5.3}$$

Ekvivalentno, *MGF* se može izraziti preko Tejlorovog reda

$$M_{p,\boldsymbol{g}}(\boldsymbol{t}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \frac{\mu_{p,\boldsymbol{g}}^{(\boldsymbol{\alpha})}}{\boldsymbol{\alpha}!} \cdot \boldsymbol{t}^{(\boldsymbol{\alpha})}. \tag{5.4}$$

Poluprsten stepenih redova

Formalni stepeni red dimenzije d definiše se kao preslikavanje $s : \mathbb{N}_0^d \to \mathbb{R}$ i može se predstaviti beskonačnom n-torkom

$$\boldsymbol{s} = \big(\, s^{(\boldsymbol{\alpha})} \,\big)_{\boldsymbol{\alpha}\in\mathbb{N}_0^d}. \tag{5.5}$$

Neka je sa $\mathbb{R}\big[\mathbb{N}_0^d\big]$ označen skup svih formalnih stepenih redova dimenzije d

$$\mathbb{R}\big[\mathbb{N}_0^d\big] = \Big\{ s : \mathbb{N}_0^d \to \mathbb{R} \Big\}. \tag{5.6}$$

Da bismo kreirali strukturu poluprstena, zapisaćemo elemente od $\mathbb{R}\big[\mathbb{N}_0^d\big]$ kao

$$\boldsymbol{s}(\boldsymbol{t}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} s^{(\boldsymbol{\alpha})} \boldsymbol{t}^{\boldsymbol{\alpha}}, \tag{5.7}$$

gde je $\boldsymbol{t} \in \mathbb{R}^d$, i koristićemo standardna pravila za množenje stepenih redova

$$(\boldsymbol{s}_1 + \boldsymbol{s}_2)(\boldsymbol{t}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} (s_1^{(\boldsymbol{\alpha})} + s_2^{(\boldsymbol{\alpha})}) \boldsymbol{t}^{\boldsymbol{\alpha}}, \tag{5.8}$$

$$(\boldsymbol{s}_1 \cdot \boldsymbol{s}_2)(\boldsymbol{t}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \Big(\sum_{\boldsymbol{\beta}+\boldsymbol{\gamma}=\boldsymbol{\alpha}} s_1^{(\boldsymbol{\beta})} s_2^{(\boldsymbol{\gamma})} \Big) \boldsymbol{t}^{\boldsymbol{\alpha}}. \tag{5.9}$$

Komutativni poluprsten stepenih redova više promenljivih možemo sada da definišemo kao petorku $\big(\mathbb{R}\big[\mathbb{N}_0^d\big], +, \cdot, 0, 1\big)$, pri čemu su sabiranje i množenje definisani sa (5.8)-(5.9), i neutralni elementi 0 i 1 su iz $\mathbb{R}$.

5.1.2 *MP* algoritam nad poluprstenom stepenih redova

Direktno izračunavanje prethodnog izraza enumerisanjem svih mogućih vrednosti za $x_{1:T}$ zahteva $\mathcal{O}(|\mathbb{X}|^T)$ operacija. U ovom odeljku razmatramo problem efikasnog izračunavanja kros-momenata za funkcije

$$p(x_{1:T}) = \prod_{M\in\mathcal{M}} \phi_M(x_M), \quad \boldsymbol{g}(x_{1:T}) = \sum_{M\in\mathcal{M}} \boldsymbol{g}_M(x_M), \tag{5.10}$$

kojima odgovara faktor-graf bez ciklusa. U ovom slučaju, kros-momenti imaju oblik

$$\mu^{(\boldsymbol{\alpha})}_{p,\,\boldsymbol{g}} = \sum_{x_{1:T}} \prod_{M\in\mathcal{M}} \phi_M(x_M) \cdot \Big(\sum_{M\in\mathcal{M}} \boldsymbol{g}_M(x_M) \Big)^{\boldsymbol{\alpha}}, \tag{5.11}$$

dok *MGF* ima oblik

$$M_{p,\boldsymbol{g}}(\boldsymbol{t}) = \sum_{x_{1:T}} \prod_{m\in\mathcal{M}} \phi_M(x_{1:T}) \cdot \boldsymbol{e}^{\,\boldsymbol{g}_M(x_{1:T})\cdot\boldsymbol{t}}. \tag{5.12}$$

Na ovaj način, *MGF* može da se predstavi kao particiona funkcija u poluprstenu stepenih redova više promenljivih, pri čemu su faktori dati sa

$$\phi_M(x_M) \cdot \boldsymbol{e}^{\boldsymbol{g}_M(x_M)\cdot\boldsymbol{t}} = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \frac{\phi_M(x_M)\boldsymbol{g}_M(x_M)^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \cdot \boldsymbol{t}^{\boldsymbol{\alpha}}, \tag{5.13}$$

pa *MGF* može da se izračuna primenom *MP* algoritma. Poruka iz faktor-čvora M ka čvoru promenljive n je

$$r_{M\to n}(x_n)(x_n,\boldsymbol{t}) = \sum_{x_{\mathcal{M}(M,n)\setminus n}} \phi_{\mathcal{M}(M,n)}(x_{\mathcal{M}(M,n)}) \cdot \boldsymbol{e}^{\boldsymbol{g}_{\mathcal{M}(M,n)}(x_{\mathcal{M}(M,n)})\cdot\boldsymbol{t}}, \tag{5.14}$$

dok je poruka iz čvora promenljive n u faktor-čvor M

$$q_{n\to M}(x_n,\boldsymbol{t}) = \sum_{x_{\mathcal{M}(n,M)\setminus n}} \phi_{\mathcal{M}(n,M)}(x_{\mathcal{M}(n,M)}) \cdot \boldsymbol{e}^{\boldsymbol{g}_{\mathcal{M}(n,M)}(x_{\mathcal{M}(n,M)})\cdot\boldsymbol{t}}, \tag{5.15}$$

gde je

$$\phi_{\mathcal{M}(i,j)}(x_{\mathcal{M}(i,j)}) = \prod_{k\in\mathcal{M}(i,j)} \phi_k(x_K), \quad \boldsymbol{g}_{\mathcal{M}(i,j)}(x_{\mathcal{M}(i,j)}) = \sum_{k\in\mathcal{M}(i,j)} \boldsymbol{g}_K(x_K).$$

Parcijalni izvodi

$$r^{(\boldsymbol{\alpha})}_{M\to n}(x_n,\mathbf{0}) = \mathcal{D}^{(\boldsymbol{\alpha})}\big\{\, r_{M\to n}(x_n,\boldsymbol{t}) \,\big\}_{\boldsymbol{t}=0}, \quad q^{(\boldsymbol{\alpha})}_{n\to M}(x_n,\mathbf{0}) = \mathcal{D}^{(\boldsymbol{\alpha})}\big\{\, q_{n\to M}(x_n,\boldsymbol{t}) \,\big\}_{\boldsymbol{t}=0}, \tag{5.16}$$

imaju vrednost

$$r^{(\boldsymbol{\alpha})}_{M\to n}(x_n,\mathbf{0}) = \sum_{x_{\mathcal{M}(M,n)\setminus n}} \phi_{\mathcal{M}(M,n)}(x_{\mathcal{M}(M,n)})\boldsymbol{g}_{\mathcal{M}(M,n)}(x_{\mathcal{M}(M,n)})^{\boldsymbol{\alpha}}, \tag{5.17}$$

$$q^{(\boldsymbol{\alpha})}_{n\to M}(x_n,\mathbf{0}) = \sum_{x_{\mathcal{M}(n,M)\setminus n}} \phi_{\mathcal{M}(n,M)}(x_{\mathcal{M}(n,M)})\boldsymbol{g}_{\mathcal{M}(n,M)}(x_{\mathcal{M}(n,M)})^{\boldsymbol{\alpha}}, \tag{5.18}$$

a kros-momenti se mogu izračunati kao

$$\mu^{(\boldsymbol{\alpha})}_{\phi_{\mathcal{M}(M,n)},\,\boldsymbol{g}_{\mathcal{M}(M,n)}} = \sum_{x_n} r^{(\boldsymbol{\alpha})}_{M\to n}(x_n,\mathbf{0}), \quad \mu^{(\boldsymbol{\alpha})}_{\boldsymbol{g}_{\mathcal{M}(n,M)},\,\boldsymbol{g}_{\mathcal{M}(n,M)}} = \sum_{x_n} q^{(\boldsymbol{\alpha})}_{n\to M}(x_n,\mathbf{0}). \tag{5.19}$$

MP algoritam za izračunavanje *MGF*-a označavaćemo sa *MGFMP* (*moment generating function message passing*). Algoritam sledi.

Inicijalizacija: Postaviti poruke iz čvorova promenljivih i faktor-čvorova u listovima na

$$q_{n\to m}(x_n, \boldsymbol{t}) = 1, \tag{5.20}$$

$$r_{M\to n}(x_n, \boldsymbol{t}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \frac{\phi_M(x_M)\boldsymbol{g}_M(x_M)^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}\cdot \boldsymbol{t}^{\boldsymbol{\alpha}}. \tag{5.21}$$

Indukcija: Kada čvor primi poruke od svih potomaka, šalje poruku roditelju, saglasno sledećim formulama

$$q_{n\to M}(x_n, \boldsymbol{t}) = \prod_{m'\in\mathcal{N}(n)\setminus m} r_{M'\to n}(x_n, \boldsymbol{t}), \tag{5.22}$$

$$r_{M\to n}(x_n, \boldsymbol{t}) = \sum_{x_{M\setminus n}} \Big(\sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \frac{\phi_M(x_M)\boldsymbol{g}_M(x_M)^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}\cdot \boldsymbol{t}^{\boldsymbol{\alpha}} \Big) \prod_{n'\in\mathcal{N}(M)\setminus n} q_{n'\to M}(x_{n'}, \boldsymbol{t}). \tag{5.23}$$

Terminacija: Proces se završava u listu koji je izabran za koren stabla. *MGF* se izračunava kao

$$M_{p,\boldsymbol{g}}(\boldsymbol{t}) = \sum_{x_n} r_{M\to n}(x_n, \boldsymbol{t}). \tag{5.24}$$

Poruke su predstavljene beskonaćnim redovima

$$r_{M\to n}(x_n, \boldsymbol{t}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \frac{r^{(\boldsymbol{\alpha})}_{M\to n}(x_n, \boldsymbol{0})}{\boldsymbol{\alpha}!}\cdot \boldsymbol{t}^{\boldsymbol{\alpha}}, \tag{5.25}$$

$$q_{n\to m}(x_n, \boldsymbol{t}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \frac{q^{(\boldsymbol{\alpha})}_{n\to M}(x_n, \boldsymbol{0})}{\boldsymbol{\alpha}!}\cdot \boldsymbol{t}^{\boldsymbol{\alpha}}. \tag{5.26}$$

Ukoliko nas interesuju samo kros-momenti do odredjenog reda, recimo $\boldsymbol{\nu}$, dovoljno je da čuvamo samo članove reda manjeg ili jednakog $\boldsymbol{\nu}$ u izrazima za poruke i faktore. Ovo može biti realizovano ukoliko čuvamo informaciju, bilo o Tejlorovim koeficijentima, bilo o *MGF* izvodima u nuli. Na primer, za poruku $r_{M\to n}(\,\cdot\,, \boldsymbol{t})$ imamo dve mogućnosti:

1. Propagacija $|A_{\boldsymbol{\nu}}|$-torki
$$\Big(\frac{r^{(\boldsymbol{\alpha})}_{M\to n}(\,\cdot\,, \boldsymbol{0})}{\boldsymbol{\alpha}!} \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}},$$
što je ekvivalentno *MP* algoritmu nad poluprstenom polinoma reda $\boldsymbol{\nu}$, i

2. Propagacija $|A_{\boldsymbol{\nu}}|$-torki
$$\Big(r^{(\boldsymbol{\alpha})}_{M\to n}(\,\cdot\,, \boldsymbol{0}) \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}},$$
što je ekvivalentno *MP* algoritmu nad binomnim poluprstenom reda $\boldsymbol{\nu}$.

U narednom poglavlju detaljnije se bavimo diskusijom na ovu temu.

5.2 *MP* algoritam nad poluprstenom polinoma

Definicija 6 *Poluprsten polinoma reda* $\boldsymbol{\nu}$ *je petorka* $(\mathbb{R}^{|\mathcal{A}_\nu|}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$ *gde su* $\oplus$ *i* $\otimes$ *definisani sa*

$$u \oplus v = \left(u^{(\boldsymbol{\alpha})} + v^{(\boldsymbol{\alpha})}\right)_{\boldsymbol{\alpha} \in A_\nu}$$
$$u \otimes v = \Big(\sum_{\boldsymbol{\beta}+\boldsymbol{\gamma}=\boldsymbol{\alpha}} u^{(\boldsymbol{\beta})} \cdot v^{(\boldsymbol{\gamma})} \Big)_{\boldsymbol{\alpha} \in A_\nu}$$

za svako $u, v \in \mathbb{R}^{|\mathcal{A}_\nu|}$. *Neutralni elementi za* $\oplus$ *i* $\otimes$ *su dati sa*

$$\mathbf{0} = (\underbrace{0, 0, \ldots, 0}_{|\mathcal{A}_\nu| \text{ times}}), \tag{5.27}$$
$$\mathbf{1} = (1, \underbrace{0, \ldots, 0}_{|\mathcal{A}_\nu|-1 \text{ times}}). \tag{5.28}$$

Primetimo da se poluprsten polinoma reda 1 svodi na entropijski poluprsten, koji je definisan u poglavlju 3.4. Sledeća lema daje formule za izračunavanje zbira i proizvoda proizvoljnog broja elemenata poluprstena polinoma i jednostavno se dokazuje indukcijom.

Lema 5.2.1 *Neka je* $w_n \in \mathbb{R}^{|\mathcal{A}_\nu|}$; $n = 1, \ldots, N$. *Tada važi sledeća jednakost*

$$\Big(\bigoplus_{n=1}^{N} w_n\Big)^{(\boldsymbol{\alpha})} = \sum_{n=1}^{N} w_n^{(\boldsymbol{\alpha})}, \tag{5.29}$$
$$\Big(\bigotimes_{n=1}^{N} w_n\Big)^{(\boldsymbol{\alpha})} = \sum_{\boldsymbol{\beta}_1+\cdots+\boldsymbol{\beta}_N=\boldsymbol{\alpha}} \prod_{n=1}^{N} w_n^{(\boldsymbol{\beta}_n)}. \tag{5.30}$$

5.2.1 *PSMP* algoritam

Neka je $w(x_{1:T})$ funkcija više promenljivih , ciji je kodomen poluprsten polinoma reda $\boldsymbol{\nu}$, i neka faktorizacija

$$w(x_{1:T}) = \bigotimes_{M \in \mathcal{M}} w_m(x_m) \tag{5.31}$$

važi za indeksni skup $\mathcal{M}$, pri čemu svaki faktor $w_M(x_M)$ zavisi od $x_M \subset x_{1:T}$, i podskupovi x_M pokrivaju $x_{1:T}$. Dalje, neka faktori imaju oblik

$$w(x_{1:T}) = \left(w_M(x_M)^{(\boldsymbol{\alpha})}\right)_{\boldsymbol{\alpha} \in A_\nu}, \tag{5.32}$$

gde je

$$w_M(x_M)^{(\boldsymbol{\alpha})} = \frac{\phi_M(x_M) \boldsymbol{g}_M(x_M)^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!}. \tag{5.33}$$

Koristeći se izrazom (5.30) i multinomnom teoremom, lako se dobija

$$w(x_{1:T})^{(\boldsymbol{\alpha})} = \frac{1}{\boldsymbol{\alpha}!} \cdot \prod_{M \in \mathcal{M}} \phi_M(x_M) \cdot \Big(\sum_{M \in \mathcal{M}} \boldsymbol{g}_M(x_M) \Big)^{\boldsymbol{\alpha}}, \tag{5.34}$$

a korišćenjem (5.11), dobija se kros-moment reda $\boldsymbol{\alpha}$

$$\frac{\mu_{p,\,g}^{(\boldsymbol{\alpha})}}{\boldsymbol{\alpha}!} = \sum_{x_{1:T}} w(x_{1:T})^{(\boldsymbol{\alpha})}. \tag{5.35}$$

Prema tome, svi kros-momenti do reda $\boldsymbol{\nu}$ mogu se izračunati kao suma u poluprstenu polinoma (5.31)

$$\bigoplus_{x_{1:T}} w(x_{1:T}) = \Big(\sum_{x_{1:T}} w(x_{1:T})^{(\boldsymbol{\alpha})} \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}} = \Big(\frac{\mu_{p,\,g}^{(\boldsymbol{\alpha})}}{\boldsymbol{\alpha}!} \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}}. \tag{5.36}$$

Sumiranje se može obaviti uz pomoć *MP* algoritma nad poluprstenom polinoma (polynomial semiring message passing, *PSMP*) reda $\boldsymbol{\nu}$, koji sledi.

Inicijalizacija: Postaviti poruke iz čvorova promenljivih i faktor-čvorova u listovima na

$$q_{n\to M}(x_n) = \Big(1, \underbrace{0, \ldots, 0}_{|\mathcal{A}_{\boldsymbol{\nu}}|-1 \text{ times}} \Big), \tag{5.37}$$

$$r_{M\to n}(x_n) = \Big(\frac{\phi_M(x_M)\boldsymbol{g}_M(x_M)^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}}. \tag{5.38}$$

Indukcija: Kada čvor primi poruke od svih potomaka, šalje poruku roditelju, saglasno sledećim formulama

$$q_{n\to M}(x_n) = \bigotimes_{m'\in\mathcal{N}(n)\setminus m} r_{M'\to n}(x_n), \tag{5.39}$$

$$r_{M\to n}(x_n) = \bigoplus_{x_M\setminus x_n} \Big(\frac{\phi_M(x_M)\boldsymbol{g}_M(x_M)^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}} \otimes \bigotimes_{n'\in\mathcal{N}(M)\setminus n} q_{n'\to M}(x_{n'}). \tag{5.40}$$

Terminacija: Proces se završava u listu koji je izabran za koren stabla. Uredjena n-torka normalizovanih kros-momenata se izračunava kao

$$\Big(\frac{\mu_{p,\,g}^{(\boldsymbol{\alpha})}}{\boldsymbol{\alpha}!} \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}} = \bigoplus_{x_n} r_{M\to n}(x_n). \tag{5.41}$$

5.2.2 *PSMP* kao $\mathcal{P}$-slika od *MGFMP*

U odeljku 5.1.2 smo pomenuli da se kros-momenti mogu izračunati primenom *MP* algoritma, u kome se šalju samo prvih $\boldsymbol{\nu}$ koeficijenata poruka

$$r_{M\to n}(x_n, \boldsymbol{t}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \hat{r}_{M\to n}^{(\boldsymbol{\alpha})} \cdot \boldsymbol{t}^{(\boldsymbol{\alpha})}, \tag{5.42}$$

gde se $\hat{r}_{M\to n}(x_n) = r_{M\to n}(x_n, \mathbf{0})^{(\boldsymbol{\alpha})}/\boldsymbol{\alpha}!$, šalju kao $|A_{\boldsymbol{\nu}}|$-torke

$$\Big(\hat{r}_{M\to n}^{(\boldsymbol{\alpha})}(x_n) \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}}, \tag{5.43}$$

iz poluprstena polinoma. Da bismo opravdali ovo, uvodimo preslikavanje $\mathcal{P}^{(\boldsymbol{\nu})} : \mathbb{R}(\boldsymbol{t}) \to \mathbb{R}^{|A_{\boldsymbol{\nu}}|}$, tako da,

$$\mathcal{P}^{(\boldsymbol{\nu})}\Big\{ \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} z^{(\boldsymbol{\alpha})} \cdot \boldsymbol{t}^{(\boldsymbol{\alpha})} \Big\} = \Big(z^{(\boldsymbol{\alpha})} \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}}. \tag{5.44}$$

Preslikavanje $\mathcal{P}$ slika neutralni element iz poluprstena stepenih redova, u jedinični element iz poluprstena polinoma

$$\mathcal{P}^{(\nu)}\{\ 1\ \} = \big(\ 1, \underbrace{0, \ \dots\ , 0}_{|\mathcal{A}_\nu|-1\text{times}}\ \big) \tag{5.45}$$

i za bilo koja dva reda

$$z_1(\boldsymbol{t}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \hat{z}_1^{(\boldsymbol{\alpha})} \cdot \boldsymbol{t}^{(\boldsymbol{\alpha})}, \quad z_2(\boldsymbol{t}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \hat{z}_2^{(\boldsymbol{\alpha})} \cdot \boldsymbol{t}^{(\boldsymbol{\alpha})}$$

preslikava operacije u odgovarajuće operacije u poluprstenu polinoma

$$\mathcal{P}^{(\nu)}\{\ z_1(\boldsymbol{t}) + z_2(\boldsymbol{t})\ \} = \big(\ \hat{z}_1^{(\boldsymbol{\alpha})}\ \big)_{\boldsymbol{\alpha}\in A_\nu} \oplus \big(\ \hat{z}_2^{(\boldsymbol{\alpha})}\ \big)_{\boldsymbol{\alpha}\in A_\nu}, \tag{5.46}$$

$$\mathcal{P}^{(\nu)}\{\ z_1(\boldsymbol{t}) \cdot z_2(\boldsymbol{t})\ \} = \big(\ \hat{z}_1^{(\boldsymbol{\alpha})}\ \big)_{\boldsymbol{\alpha}\in A_\nu} \otimes \big(\ \hat{z}_2^{(\boldsymbol{\alpha})}\ \big)_{\boldsymbol{\alpha}\in A_\nu}, \tag{5.47}$$

što sledi direktno iz definicije. Uzimajući u obzir razvoj u red za $\phi_M(x_M)\boldsymbol{g}_M(x_M)$, $r_{M\to n}(x_n, \boldsymbol{t})$ i $q_{n\to M}(x_n, \boldsymbol{t})$, možemo izračunati

$$\mathcal{P}^{(\nu)}\{\ \ \phi_M(x_M) \cdot \boldsymbol{e}^{\boldsymbol{g}_M(x_M)\cdot \boldsymbol{t}}\ \} = \Big(\ \frac{\phi_M \cdot \boldsymbol{g}_M}{\boldsymbol{\alpha}}^{(\boldsymbol{\alpha}!)}\ \Big)_{\boldsymbol{\alpha}!\in A_\nu}, \tag{5.48}$$

$$\mathcal{P}^{(\nu)}\{\ r_{M\to n}(x_n, \boldsymbol{t})\ \} = \big(\ \hat{r}_{M\to n}^{(\boldsymbol{\alpha})}\ \big)_{\boldsymbol{\alpha}\in A_\nu}, \tag{5.49}$$

$$\mathcal{P}^{(\nu)}\{\ q_{n\to M}(x_n, \boldsymbol{t})\ \} = \big(\ \hat{q}_{n\to M}^{(\boldsymbol{\alpha})}\ \big)_{\boldsymbol{\alpha}\in A_\nu}. \tag{5.50}$$

Primenom preslikavanja $\mathcal{P}$ na *MGFMP* jednačine dobijamo *PSMP* algoritam.

5.3 *MP* algoritam nad binomnim poluprstenom

Definicija 7 *Binomni poluprsten reda $\boldsymbol{\nu}$ je petorka $(\mathbb{R}^{|\mathcal{A}_\nu|}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$, gde su $\oplus$ i $\otimes$ definisani sa*

$$u \oplus v = \left(u^{(\boldsymbol{\alpha})} + v^{(\boldsymbol{\alpha})}\right)_{\boldsymbol{\alpha}\in A_\nu}, \tag{5.51}$$

$$u \otimes v = \Big(\ \sum_{\boldsymbol{\beta}+\boldsymbol{\gamma}=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta},\boldsymbol{\gamma}}\, u^{(\boldsymbol{\beta})} \cdot v^{(\boldsymbol{\gamma})}\ \Big)_{\boldsymbol{\alpha}\in A_\nu}, \tag{5.52}$$

za svako $u, v \in \mathbb{R}^{|\mathcal{A}_\nu|}$. Neutralni elementi za $\oplus$ i $\otimes$ su redom

$$\mathbf{0} = (\ \underbrace{0, 0, \dots, 0}_{|\mathcal{A}_\nu|\ times}\ ), \tag{5.53}$$

$$\mathbf{1} = (1, \underbrace{0, \dots, 0}_{|\mathcal{A}_\nu|-1\ times}\ ). \tag{5.54}$$

Kao i u slučaju poluprstena polinoma, binomni poluprsten reda 1 svodi se na entropijski poluprsten definisan u poglavlju 3.4. Sledeća lema daje formule za izračunavanje zbira i proizvoda proizvoljnog broja elemenata binomnog poluprstena.

Lema 5.3.1 *Neka su* $w_n \in \mathbb{R}^{|\mathcal{A}_\nu|}$; $n = 1, \ldots, N$. *Tada važe sledeće jednakosti*

$$\Big(\bigoplus_{n=1}^{N} w_n\Big)^{(\boldsymbol{\alpha})} = \sum_{n=1}^{N} w_n^{(\boldsymbol{\alpha})}, \tag{5.55}$$

$$\Big(\bigotimes_{n=1}^{N} w_n\Big)^{(\boldsymbol{\alpha})} = \sum_{\boldsymbol{\beta}_1+\cdots+\boldsymbol{\beta}_N=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N} \prod_{n=1}^{N} w_n^{(\boldsymbol{\beta}_n)}. \tag{5.56}$$

Dokaz. Lemu dokazujemo indukcijom. Jednakost za sabiranje (5.55) sledi direktno iz definicije sabiranja u binomnom poluprstenu. Jednakost za množenje se za slučaj $N = 2$ svodi na definiciju množenja u binomnom poluprstenu

$$\big(w_1 \otimes w_2\big)^{(\boldsymbol{\alpha})} = \sum_{\boldsymbol{\beta}_1+\boldsymbol{\beta}_2=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2} w_1^{(\boldsymbol{\beta}_1)} w_2^{(\boldsymbol{\beta}_2)}.$$

Neka sada jednakost (5.56) važi za neko N. Tada

$$\begin{aligned}
\Big(\bigotimes_{n=1}^{N+1} w_n\Big)^{(\boldsymbol{\alpha})} &= \Big(\bigotimes_{n=1}^{N} w_n \otimes w_{N+1}\Big)^{(\boldsymbol{\alpha})} = \sum_{\boldsymbol{\gamma}+\boldsymbol{\beta}_{N+1}=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\gamma}, \boldsymbol{\beta}_{N+1}} \cdot \Big(\bigotimes_{n=1}^{N} w_n\Big)^{(\boldsymbol{\gamma})} \cdot w_{N+1}^{(\boldsymbol{\beta}_{N+1})} = \\
&= \sum_{\boldsymbol{\gamma}+\boldsymbol{\beta}_{N+1}=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\gamma}, \boldsymbol{\beta}_{N+1}} \sum_{\boldsymbol{\beta}_1+\cdots+\boldsymbol{\beta}_N=\boldsymbol{\gamma}} \binom{\boldsymbol{\gamma}}{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_N} \prod_{n=1}^{N} w_n^{(\boldsymbol{\beta}_n)} \cdot w_{N+1}^{(\boldsymbol{\beta}_{N+1})} = \\
&= \sum_{\boldsymbol{\gamma}+\boldsymbol{\beta}_{N+1}=\boldsymbol{\alpha}} \sum_{\boldsymbol{\beta}_1+\cdots+\boldsymbol{\beta}_N=\boldsymbol{\gamma}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{N+1}} \prod_{n=1}^{N+1} w_n^{(\boldsymbol{\beta}_n)} = \\
&= \sum_{\boldsymbol{\beta}_1+\cdots+\boldsymbol{\beta}_{N+1}=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{N+1}} \prod_{n=1}^{N+1} w_n^{(\boldsymbol{\beta}_n)},
\end{aligned}$$

što dokazuje lemu. □

5.3.1 *BSMP* algoritam

Slično kao i u prethodno obradjenom slučaju poluprstena polinoma, neka je $w(x_{1:T})$ funkcija više promenljivih, čiji je kodomen binomni poluprsten reda $\boldsymbol{\nu}$, i neka faktorizacija

$$w(x_{1:T}) = \bigotimes_{M \in \mathcal{M}} w_M(x_M) \tag{5.57}$$

važi za indeksni skup $\mathcal{M}$, pri čemu svaki faktor $w_M(x_M)$ zavisi od $x_M \subset x_{1:T}$ i podskupovi x_M pokrivaju $x_{1:T}$. Dalje, neka faktori imaju oblik

$$w(x_{1:T}) = \Big(w_M(x_M)^{(\boldsymbol{\alpha})}\Big)_{\boldsymbol{\alpha} \in A_{\boldsymbol{\nu}}}, \tag{5.58}$$

gde je

$$w_M(x_M)^{(\boldsymbol{\alpha})} = \phi_M(x_M) \boldsymbol{g}_M(x_M)^{\boldsymbol{\alpha}}. \tag{5.59}$$

Koristeći se izrazom (5.56) i multinomnom teoremom, dobija se

$$w(x_{1:T})^{(\boldsymbol{\alpha})} = \prod_{M \in \mathcal{M}} \phi_M(x_M) \cdot \Big(\sum_{M \in \mathcal{M}} \boldsymbol{g}_M(x_M)\Big)^{\boldsymbol{\alpha}}, \tag{5.60}$$

a korišćenjem (5.11), dobija se kros-moment reda $\boldsymbol{\alpha}$

$$\mu_{p,\,g}^{(\boldsymbol{\alpha})} = \sum_{x_{1:T}} w(x_{1:T})^{\boldsymbol{\alpha}}. \tag{5.61}$$

Prema tome, svi kros-momenti do reda $\boldsymbol{\nu}$ mogu se izračunati kao suma u binomnom poluprstenu (5.57)

$$\bigoplus_{x_{1:T}} w(x_{1:T}) = \Big(\sum_{x_{1:T}} w(x_{1:T}) \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}} = \Big(\mu_{p,\,g}^{(\boldsymbol{\alpha})} \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}}. \tag{5.62}$$

Sumiranje se može obaviti uz pomoć *MP* algoritma nad binomnim poluprstenom (binomial semiring message passing, *BSMP*) reda $\boldsymbol{\nu}$, koji sledi.

Inicijalizacija: Postaviti poruke iz čvorova promenljivih i faktor-čvorova u listovima na

$$q_{n\to M}(x_n) = \big(1, \underbrace{0, \ldots, 0}_{|\mathcal{A}_{\boldsymbol{\nu}}|-1 \text{ times}} \big), \tag{5.63}$$

$$r_{M\to n}(x_n) = \Big(\phi_M(x_M) \boldsymbol{g}_M(x_M)^{\boldsymbol{\alpha}} \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}}. \tag{5.64}$$

Indukcija: Kada čvor primi poruke od svih potomaka, šalje poruku roditelju, saglasno sledećim formulama

$$q_{n\to M}(x_n) = \bigotimes_{m'\in\mathcal{N}(n)\setminus m} r_{M'\to n}(x_n), \tag{5.65}$$

$$r_{M\to n}(x_n) = \bigoplus_{x_M\setminus x_n} \Big(\phi_M(x_M) \boldsymbol{g}_M(x_M)^{\boldsymbol{\alpha}} \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}} \otimes \bigotimes_{n'\in\mathcal{N}(M)\setminus n} q_{n'\to M}(x_{n'}). \tag{5.66}$$

Terminacija: Proces se završava u listu koji je izabran za koren stabla. Uredjena n-torka kros-momenata se izračunava kao

$$\Big(\mu_{p,\,g}^{(\boldsymbol{\alpha})} \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}} = \bigoplus_{x_n} r_{M\to n}(x_n). \tag{5.67}$$

5.3.2 *BSMP* kao $\mathcal{B}$-slika od *MGFMP*

Alternativna mogućnost za predstavljanje *MGF* poruka

$$r_{M\to n}(x_n, \boldsymbol{t}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \frac{r_{M\to n}(x_n)^{(\boldsymbol{\alpha})}}{\boldsymbol{\alpha}!} \cdot \boldsymbol{t}^{\boldsymbol{\alpha}} \tag{5.68}$$

je preko $|A_{\boldsymbol{\nu}}|$-torke

$$\Big(r_{M\to n}^{(\boldsymbol{\alpha})}(x_n) \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}}, \tag{5.69}$$

iz binomnog poluprstena. Neka je $\mathcal{B}^{(\boldsymbol{\nu})} : \mathbb{R}(\boldsymbol{t}) \to \mathbb{R}^{|A_{\boldsymbol{\nu}}|}$, tako da,

$$\mathcal{B}^{(\boldsymbol{\nu})} \Big\{ \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \frac{z^{(\boldsymbol{\alpha})}}{\boldsymbol{\alpha}!} \cdot \boldsymbol{t}^{(\boldsymbol{\alpha})} \Big\} = \big(z^{(\boldsymbol{\alpha})} \big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}}. \tag{5.70}$$

Preslikavanje $\mathcal{B}$ slika jedinični element iz poluprstena stepenih redova u jedinični element binomnog poluprstena

$$\mathcal{B}^{(\nu)}\{\ 1\ \} = (\ 1, \underbrace{0, \ldots, 0}_{|\mathcal{A}_\nu|-1\text{times}}\ ), \tag{5.71}$$

i za svaka dva reda

$$z_1(\boldsymbol{t}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \frac{z_1^{(\boldsymbol{\alpha})}}{\boldsymbol{\alpha}!}\cdot \boldsymbol{t}^{(\boldsymbol{\alpha})}, \quad z_2(\boldsymbol{t}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \frac{z_2^{(\boldsymbol{\alpha})}}{\boldsymbol{\alpha}!}\cdot \boldsymbol{t}^{(\boldsymbol{\alpha})}$$

slika operacije sabiranja i množenja u odgovarajuće operacije u binomnom poluprstrenu.

$$\mathcal{B}^{(\nu)}\{\ z_1(\boldsymbol{t}) + z_2(\boldsymbol{t})\ \} = (\ z_1^{(\boldsymbol{\alpha})}\ )_{\boldsymbol{\alpha}\in A_\nu} \oplus (\ z_2^{(\boldsymbol{\alpha})}\ )_{\boldsymbol{\alpha}\in A_\nu}, \tag{5.72}$$

$$\mathcal{B}^{(\nu)}\{\ z_1(\boldsymbol{t}) \cdot z_2(\boldsymbol{t})\ \} = (\ z_1^{(\boldsymbol{\alpha})}\ )_{\boldsymbol{\alpha}\in A_\nu} \otimes (\ z_2^{(\boldsymbol{\alpha})}\ )_{\boldsymbol{\alpha}\in A_\nu}. \tag{5.73}$$

Prva jednakost sledi iz definicije, dok druga sledi iz

$$\mathcal{B}^{(\nu)}\Big\{\ z_1(\boldsymbol{t})\cdot z_2(\boldsymbol{t})\ \Big\} = \mathcal{B}^{(\nu)}\Big\{\ \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}}\sum_{\boldsymbol{\beta}+\boldsymbol{\gamma}=\boldsymbol{\alpha}} \frac{z_1^{(\boldsymbol{\beta})}}{\boldsymbol{\beta}!}\cdot\frac{z_2^{(\boldsymbol{\gamma})}}{\boldsymbol{\gamma}!}\cdot \boldsymbol{t}^{\boldsymbol{\alpha}}\ \Big\} =$$

$$\mathcal{B}^{(\nu)}\{\ \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \frac{1}{\boldsymbol{\alpha}!}\cdot \sum_{\boldsymbol{\beta}+\boldsymbol{\gamma}=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta},\boldsymbol{\gamma}} z_1^{(\boldsymbol{\beta})}\cdot z_2^{(\boldsymbol{\gamma})}\cdot \boldsymbol{t}^{\boldsymbol{\alpha}}\ \} =$$

$$\Big(\ \sum_{\boldsymbol{\beta}+\boldsymbol{\gamma}=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta},\boldsymbol{\gamma}} z_1^{(\boldsymbol{\beta})}\cdot z_2^{(\boldsymbol{\gamma})}\ \Big)_{\boldsymbol{\alpha}\in A_\nu} = (\ z_1^{(\boldsymbol{\alpha})}\ )_{\boldsymbol{\alpha}\in A_\nu} \otimes (\ z_2^{(\boldsymbol{\alpha})}\ )_{\boldsymbol{\alpha}\in A_\nu}.$$

Uzimajući u obzir razvoj u red za $\phi_M(x_M)\boldsymbol{g}_M(x_M)$, $r_{M\to n}(x_n,\boldsymbol{t})$ i $q_{n\to M}(x_n,\boldsymbol{t})$, možemo izračunati

$$\mathcal{B}^{(\nu)}\{\ \phi_M(x_M)\cdot \boldsymbol{e}^{\boldsymbol{g}_M(x_M)\cdot\boldsymbol{t}}\ \} = (\ \phi_M\cdot \boldsymbol{g}_M^{(\boldsymbol{\alpha})}\ )_{\boldsymbol{\alpha}\in A_\nu}, \tag{5.74}$$

$$\mathcal{B}^{(\nu)}\{\ r_{M\to n}(x_n,\boldsymbol{t})\ \} = (\ r_{M\to n}^{(\boldsymbol{\alpha})}\ )_{\boldsymbol{\alpha}\in A_\nu}, \tag{5.75}$$

$$\mathcal{B}^{(\nu)}\{\ q_{n\to M}(x_n,\boldsymbol{t})\ \} = (\ q_{n\to M}^{(\boldsymbol{\alpha})}\ )_{\boldsymbol{\alpha}\in A_\nu}. \tag{5.76}$$

Slično kao i kod *PSMP*-a, primenom preslikavanja $\mathcal{B}$ na *MGFMP* jednačine dobijamo *BSMP* algoritam.

5.4 Vremenska i memorijska kompleksnost *PSMP* i *BSMP* algoritama

5.4.1 Vremenska i memorijska kompleksnost *MP* algoritma u odnosu na realne operacije

U odeljku 2.2.2 razmatrali smo vremensku kompleksnost *MP* algoritma i označili je sa $\mathrm{T}^{\mathrm{mpa}}$. Vremenska kompleksnost označava asimptotski broj operacija u poluprstenu i data je kao zbir $\mathrm{T}^{\mathrm{mpa}} = \mathrm{T}_{\oplus}^{\mathrm{mpa}} + \mathrm{T}_{\otimes}^{\mathrm{mpa}}$, gde su $\mathrm{T}_{\oplus}^{\mathrm{mpa}}$ i $\mathrm{T}_{\otimes}^{\mathrm{mpa}}$ vremenske kompleksnosti definisane kao asimptotski

broj sabiranja i množenja u poluprstenu potrebnih za izvršavanje algoritma, kada $|\mathcal{N}|$, $|\mathcal{M}|$ i $|\mathbb{X}|$ teže beskonačnosti. Pokazano je da je

$$\mathrm{T}_{\oplus}^{\mathrm{mpa}} = \mathcal{O}\Big(\sum_{M \in \mathcal{M}} |\mathbb{X}|^{d(M)} \Big), \tag{5.77}$$

$$\mathrm{T}_{\otimes}^{\mathrm{mpa}} = \mathcal{O}\Big(\sum_{M \in \mathcal{M}} d(M) \cdot |\mathbb{X}|^{d(M)} \Big). \tag{5.78}$$

U ovom odeljku razmatramo vremensku kompleksnost kao asimptotski broj realnih operacija (sabiranja, oduzimanja, množenja i deljenja), potrebnih za izvršenje algoritma, sa dodatnim zahtevom, da $|A_v|$ teži beskonačnosti. Vremenska kompleksnost je data sa

$$\mathrm{T}^{\mathrm{mpa}} = \mathrm{T}_{+}^{\mathrm{mpa}} + \mathrm{T}_{\times}^{\mathrm{mpa}}, \tag{5.79}$$

gde su $\mathrm{T}_{+}^{\mathrm{sr}}$ i $\mathrm{T}_{\times}^{\mathrm{sr}}$ asimptotski brojevi realnih sabiranja i množenja, koji su dati sa

$$\mathrm{T}_{+}^{\mathrm{mpa}} = \mathrm{T}_{\oplus}^{\mathrm{mpa}} \cdot \mathrm{T}_{+}^{\oplus} + \mathrm{T}_{\otimes}^{\mathrm{mpa}} \cdot \mathrm{T}_{+}^{\otimes}, \tag{5.80}$$

$$\mathrm{T}_{\times}^{\mathrm{mpa}} = \mathrm{T}_{\oplus}^{\mathrm{mpa}} \cdot \mathrm{T}_{\times}^{\oplus} + \mathrm{T}_{\otimes}^{\mathrm{mpa}} \cdot \mathrm{T}_{\times}^{\otimes}, \tag{5.81}$$

gde su $\mathrm{T}_{+}^{\oplus}$ i $\mathrm{T}_{+}^{\otimes}$ brojevi realnih sabiranja, a $\mathrm{T}_{\times}^{\oplus}$ i $\mathrm{T}_{\times}^{\otimes}$ brojevi realnih množenja, potrebnih za izvršenje odgovarajuće operacije u poluprstenu.

Slično, memorijska kompleksnost data je maksimalnim brojem realnih brojeva potrebnih za izvršenje algoritma, kada $|\mathcal{V}_l|$ i $|A_v|$ teže beskonačnosti, i može da se izračuna kao

$$\mathrm{M}_{\mathbb{R}}^{\mathrm{mpa}} = \mathrm{M}_{\mathbb{K}}^{\mathrm{mpa}} + \mathrm{M}_{\mathbb{R}}^{\mathrm{add}}. \tag{5.82}$$

Ovde je $\mathrm{M}_{\mathbb{K}}^{\mathrm{mpa}}$ memorijska kompleksnost *MP* algoritma nad poluprstenom, koja je na osnovu izlaganja u odeljku 2.2.2 data sa

$$\mathrm{M}_{\mathbb{K}}^{\mathrm{mpa}} = \mathcal{O}\big(|\mathcal{V}_l| \cdot |\mathbb{X}| \big), \tag{5.83}$$

gde je $|\mathcal{V}_l|$ broj listova, a $\mathrm{M}_{\mathbb{R}}^{\mathrm{psr}}$ predstavlja broj realnih brojeva potrebnih za čuvanje jednog elementa u poluprstenu.

5.4.2 Vremenska i memorijska kompleksnost *PSMP* algoritma

Vremenska kompleksnost: Za sabiranje u poluprstenu polinoma

$$u \oplus v = \Big(u^{(\boldsymbol{\alpha})} + v^{(\boldsymbol{\alpha})} \Big)_{\boldsymbol{\alpha} \in A_v} \tag{5.84}$$

potrebno je $|A_v|$ realnih sabiranja, i nisu potrebna množenja, pa je

$$\mathrm{T}_{+}^{\oplus} = \mathcal{O}\big(|A_v|\big), \quad \mathrm{T}_{\times}^{\oplus} = 0, \tag{5.85}$$

dok je za množenje u poluprstenu polinoma

$$u \otimes v = \Big(\sum_{\boldsymbol{\beta}+\boldsymbol{\gamma}=\boldsymbol{\alpha}} u^{(\boldsymbol{\beta})} \cdot v^{(\boldsymbol{\gamma})} \Big)_{\boldsymbol{\alpha} \in A_v} \tag{5.86}$$

potrebno

$$\mathrm{T}_{+}^{\otimes} = \mathcal{O}(|A_{\boldsymbol{\nu}}|^2), \quad \mathrm{T}_{\times}^{\otimes} = \mathcal{O}(|A_{\boldsymbol{\nu}}|^2), \tag{5.87}$$

pošto

$$\mathcal{O}\Big(\sum_{\alpha_1,\ldots,\alpha_d \in A_{\boldsymbol{\nu}}} \alpha_1 \cdots \alpha_d \Big) = \mathcal{O}\big(\nu_1^2 \cdots \nu_d^2 \big) = \mathcal{O}(|A_{\boldsymbol{\nu}}|^2).$$

Na osnovu razmatranja iz odeljka 5.4.1, dobija se vremenska kompleksnost za *PSMP*

$$\mathrm{T}_{+}^{\mathrm{pmp}} = \mathcal{O}\Big(\sum_{M \in \mathcal{M}} |\mathbb{X}|^{d(M)} \cdot |A_{\boldsymbol{\nu}}| \Big), \tag{5.88}$$

$$\mathrm{T}_{\times}^{\mathrm{pmp}} = \mathcal{O}\Big(\sum_{M \in \mathcal{M}} d(M) \cdot |\mathbb{X}|^{d(M)} \cdot |A_{\boldsymbol{\nu}}| \Big). \tag{5.89}$$

Kada se korsti za izračunavanje kros-momenata, *PSMP* ima dodatne zahteve za izračunavanje faktora

$$\Big(\frac{\phi_M(x_M) \boldsymbol{g}_M(x_M)^{\boldsymbol{\alpha}}}{\boldsymbol{\alpha}!} \Big)_{\boldsymbol{\alpha} \in A_{\boldsymbol{\nu}}}. \tag{5.90}$$

Primetimo da se za svako $\boldsymbol{\alpha} \leq \boldsymbol{\nu}$, članovi $\phi_M(x_M)\boldsymbol{g}_M(x_M)^{\boldsymbol{\alpha}}$ mogu dobiti kao rezultat izračunavanja $\phi_M(x_M)\boldsymbol{g}_M(x_M)^{\boldsymbol{\nu}}$. Ovo izračunavanje zahteva $\mathcal{O}\big(|\mathbb{X}|^{d(M)} \cdot (\nu_1 + \cdots + \nu_d)\big)$ množenja, što ne utiče na asimptotsku kompleksnost za množenje. Pod pretpostavkom da su potrebne vrednosti za faktorijele dostupne, potrebno je $\mathcal{O}\big(|\mathbb{X}|^{d(M)} \cdot |A_{\boldsymbol{\nu}}|\big)$ deljenja u svakom faktor-čvoru, pa je vremenska kompleksnost za deljenje

$$\mathrm{T}_{/}^{\mathrm{pmp}} = \mathcal{O}\Big(\sum_{M \in \mathcal{M}} |\mathbb{X}|^{d(M)} \cdot |A_{\boldsymbol{\nu}}| \Big). \tag{5.91}$$

Sumiranjem izraza (5.88), (5.89) i (5.91), dobijamo totalnu vremensku kompleksnost *PSMP* algoritma za izračunavanje svih kros-momenata reda manjeg ili jednakog $\boldsymbol{\nu}$

$$\mathrm{T}^{\mathrm{pmp}} = \mathcal{O}\Big(\sum_{M \in \mathcal{M}} d(M) \cdot |\mathbb{X}|^{d(M)} \cdot |A_{\boldsymbol{\nu}}| \Big). \tag{5.92}$$

Memorijska kompleksnost: Memorijska kompleksnost *PSMP* algoritma za izračunavanje kros-momenata data je maksimalnim brojem realnih brojeva, potrebnih za izvršenje algoritma, kada $|\mathcal{V}_l|$ i $|A_{\boldsymbol{\nu}}|$ teže beskonačnosti, i može da se izračuna kao

$$\mathrm{M}_{\mathbb{R}}^{\mathrm{pmp}} = \mathrm{M}_{\mathbb{R}}^{\mathrm{mpa}} + \mathrm{M}_{\mathbb{R}}^{\mathrm{add}}. \tag{5.93}$$

Ovde je $\mathrm{M}_{\mathbb{R}}^{\mathrm{mpa}} = \mathrm{M}_{\mathrm{psr}}^{\mathrm{mpa}} \cdot \mathrm{M}_{\mathbb{R}}^{\mathrm{psr}}$ memorijska kompleksnost *MP* algoritma nad poluprstenom polinoma, gde je $\mathrm{M}_{\mathrm{psr}}^{\mathrm{mpa}} = \mathcal{O}\big(|\mathcal{V}_l| \cdot |\mathbb{X}| \big)$, saglasno izrazu (5.83), i $\mathrm{M}_{\mathbb{R}}^{\mathrm{psr}} = |A_{\boldsymbol{\nu}}|$ predstavlja broj realnih brojeva, potrebnih za čuvanje jednog elementa u poluprstenu, kada $|A_{\boldsymbol{\nu}}|$ teži beskonaňosti. Saglasno tome

$$\mathrm{M}_{\mathbb{R}}^{\mathrm{mpa}} = \mathcal{O}\big(|\mathcal{V}_l| \cdot |\mathbb{X}| \cdot |A_{\boldsymbol{\nu}}| \big). \tag{5.94}$$

Član $\mathrm{M}_{\mathbb{R}}^{\mathrm{add}}$ predstavlja dodatni memorijski prostor. Kada se koristi za izračunavanje kros-momenata, *PSMP*-u je potrebna dodatna memorija od $\mathcal{O}(\,|A_{\boldsymbol{\nu}}|\,)$ za čuvanje vrednosti za faktorijele. Faktori se izračunavaju jedanom po faktor-čvoru, i pošto se izračunata vrednost iskoristi, može biti izbrisana. Dakle, maksimalno memorijsko zauzeće je $\mathcal{O}\big(\max_{M \in \mathcal{M}} \{\, |\mathbb{X}|^{d(M)} \cdot |A_{\boldsymbol{\nu}}| \,\} \big)$, pa je

$$\mathrm{M}_{\mathbb{R}}^{\mathrm{add}} = \mathcal{O}\big(\max_{M \in \mathcal{M}} \{\, |\mathbb{X}|^{d(M)} \cdot |A_{\boldsymbol{\nu}}| \,\} \big), \tag{5.95}$$

a asimptotska memorijska kompleksnost algoritma je

$$\mathrm{M}_{\mathbb{R}}^{\mathrm{pmp}} = \mathcal{O}\big(|\mathcal{V}_l| \cdot |\mathbb{X}| \cdot |A_{\boldsymbol{\nu}}| + \max_{M \in \mathcal{M}} \big\{ |\mathbb{X}|^{d(M)} \cdot |A_{\boldsymbol{\nu}}| \big\} \big). \tag{5.96}$$

5.4.3 Vremenska i memorijska kompleksnost *BSMP* algoritma

Vremenska kompleksnost: Slično kao i kod poluprstena polinoma, za sabiranje u binomnom poluprstenu

$$u \oplus v = \Big(u^{(\boldsymbol{\alpha})} + v^{(\boldsymbol{\alpha})} \Big)_{\boldsymbol{\alpha} \in A_{\boldsymbol{\nu}}} \tag{5.97}$$

potrebno je $|A_{\boldsymbol{\nu}}|$ realnih sabiranja

$$\mathrm{T}_{+}^{\oplus} = \mathcal{O}\big(|A_{\boldsymbol{\nu}}|\big); \quad \mathrm{T}_{\times}^{\oplus} = 0. \tag{5.98}$$

Pod pretpostavkom da su svi binomni koeficijenti sačuvani i dostupni za vreme izvršenja algoritma, za množenje u binomnom poluprstenu

$$u \otimes v = \Big(\sum_{\boldsymbol{\beta}+\boldsymbol{\gamma}=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta},\boldsymbol{\gamma}} u^{(\beta)} \cdot v^{(\gamma)} \Big)_{\boldsymbol{\alpha} \in A_{\boldsymbol{\nu}}}$$

potrebno je $\mathcal{O}(\alpha_1 \cdots \alpha_d)$ realnih sabiranja i množenja za element sa indeksom $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d)$, pa izračunavanje cele $|A_{\boldsymbol{\nu}}|$-torke ima kompleksnost

$$\mathrm{T}_{+}^{\otimes} = \mathcal{O}(|A_{\boldsymbol{\nu}}|^2); \quad \mathrm{T}_{\times}^{\otimes} = \mathcal{O}(|A_{\boldsymbol{\nu}}|^2). \tag{5.99}$$

Na osnovu razmatranja iz odeljka 5.4.1, vremenska kompleksnost *PSMP* algoritma je

$$\mathrm{T}_{+}^{\mathrm{bmp}} = \mathcal{O}\Big(\sum_{M \in \mathcal{M}} |\mathbb{X}|^{d(M)} \cdot |A_{\boldsymbol{\nu}}| \Big), \tag{5.100}$$

$$\mathrm{T}_{\times}^{\mathrm{bmp}} = \mathcal{O}\Big(\sum_{M \in \mathcal{M}} d(M) \cdot |\mathbb{X}|^{d(M)} \cdot |A_{\boldsymbol{\nu}}| \Big). \tag{5.101}$$

Na osnovu sličnih razmatranja, kao u odeljku 5.4.2 izračunavanje faktora

$$\big(\phi_M(x_M) \boldsymbol{g}_M(x_M)^{\boldsymbol{\alpha}} \big)_{\boldsymbol{\alpha} \in A_{\boldsymbol{\nu}}} \tag{5.102}$$

ne utiče na vremensku kompleksnost, pa je

$$\mathrm{T}^{\mathrm{bmp}} = \mathcal{O}\Big(\sum_{M \in \mathcal{M}} d(M) \cdot |\mathbb{X}|^{d(M)} \cdot |A_{\boldsymbol{\nu}}| \Big). \tag{5.103}$$

Memorijska kompleksnost: Slično kao i za *PSMP*, memorijska kompleksnost *BSMP* algoritma može se izračunati kao

$$\mathrm{M}_{\mathbb{R}}^{\mathrm{pmp}} = \mathrm{M}_{\mathbb{R}}^{\mathrm{mpa}} + \mathrm{M}_{\mathbb{R}}^{\mathrm{add}}, \tag{5.104}$$

gde je

$$\mathrm{M}_{\mathbb{R}}^{\mathrm{mpa}} = \mathcal{O}\big(|\mathcal{V}_l| \cdot |\mathbb{X}| \cdot |A_{\boldsymbol{\nu}}| \big), \tag{5.105}$$

pri čemu je dodatna memorija označena sa $\mathrm{M}_{\mathbb{R}}^{\mathrm{add}}$. U slučaju *BSMP* algoritma, dodatni memorijski prostor odnosi se na čuvanje binomnih koeficijenata, $\mathcal{O}(\ |A_{\boldsymbol{\nu}}|^2\ )$, a za čuvanje faktora potrebno je $\mathcal{O}\big(\ \max_{M\in\mathcal{M}}\big\{\ |\mathbb{X}|^{d(M)}\cdot|A_{\boldsymbol{\nu}}|\ \big\}\ \big)$, kao sto je objašnjeno u odeljku 5.4.2. Shodno tome

$$\mathrm{M}_{\mathbb{R}}^{\mathrm{add}} = \mathcal{O}\big(\ |A_{\boldsymbol{\nu}}|^2 + \max_{M\in\mathcal{M}}\big\{\ |\mathbb{X}|^{d(M)}\cdot|A_{\boldsymbol{\nu}}|\ \big\}\ \big), \tag{5.106}$$

pa je asimptotska memorijska kompleksnost algoritma

$$\mathrm{M}_{\mathbb{R}}^{\mathrm{bmp}} = \mathcal{O}\big(\ |\mathcal{V}_l|\cdot|\mathbb{X}|\cdot|A_{\boldsymbol{\nu}}| + |A_{\boldsymbol{\nu}}|^2 + \max_{M\in\mathcal{M}}\big\{\ |\mathbb{X}|^{d(M)}\cdot|A_{\boldsymbol{\nu}}|\ \big\}\ \big). \tag{5.107}$$

5.4.4 Poredjenje kompleksnosti *PSMP* i *BSMP* algoritama

PSMP i *BSMP* algoritmi imaju istu vremensku kompleksnost, pri čemu za *BSMP* nije potrebna operacija deljenja kao kod *PSMP*-a. Sa druge strane, *BSMP* ima veću memorijsku kompleksnost $\mathcal{O}(\ |A_{\boldsymbol{\nu}}|^2\ )$, zbog čuvanja binomnih koeficijenata, dok je za *PSMP*, koji čuva faktorijele potrebno $\mathcal{O}(\ |A_{\boldsymbol{\nu}}|\ )$ memorijskog prostora.

5.5 Prethodni rad

Kao što smo već pomenuli, algoritam za efikasno izračunavanje kros-momenata za red manji ili jednak $\boldsymbol{\nu}$

$$\Big(\sum_{x_{1:T}}\prod_{M\in\mathcal{M}}\phi_M(x_M)\cdot\Big(\sum_{M\in\mathcal{M}}\boldsymbol{g}_M(x_M)\Big)^{\boldsymbol{\alpha}}\Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}}, \tag{5.108}$$

za skalarne funkcije, prethodno su dali *Cowell* i saradnici [15], i njihov algoritam pretstavlja skalarnu verziju *PSMP* algoritma. Takodje, *EMP* algoritam iz odeljka 2 je *PSMP*, odnosno *BSMP* algoritam reda $\boldsymbol{\nu} = 1$. U ovom poglavlju dajemo pregled ostalih algoritama koji predstavljaju specijalan slučaj *PSMP* i *BSMP* algoritama.

5.5.1 Kros-momenti reda $\boldsymbol{\nu} = (1,1)$

Efikasno izračunavanje kros-momenata (5.108) za slučaj

$$\mathcal{A}_{\boldsymbol{\nu}} = \big\{(0,0),(0,1),(1,0),(1,1)\big\},$$

razmatrali su *Kulesza* i *Taskar* [50]. Algoritam razmatran u [50] funkcioniše kao *MP* algoritam nad entropijskim poluprstenom drugog reda [56], koji se može dobiti kao poluprsten polinoma, odnosno binomni poluprsten reda $\boldsymbol{\nu} = (1,1)$, tako da je algoritam koji su dali *Kulesza* i *Taskar* u stvari *PSMP*, odnosno *BSMP* algoritam reda $\boldsymbol{\nu} = (1,1)$.

5.5.2 Kros-momenti na lancima

Problem efikasnog izračunavanja kros-momenata u slučaju faktorizacije sa strukturom lanca, razmatran je u [2] i [30]. Algoritmi iz ova dva rada predstavljaju *FB* algoritam (poglavlje 4.1) nad binomnim polurstenom. Dakle, razmatraju se višedimenziona slučajna promenljiva $X_{1:T}$

sa raspodelom $p_{X_{1:T}}$ i funkcija ove slučajne promenljive $\boldsymbol{g} : \mathbb{X}^T \to \mathbb{R}^d$, koje se mogu predstaviti kao proizvod, odnosno zbir

$$p(x_{1:T}) = \prod_{n=1}^{T} \phi_n(x_{n-1}, x_n),\ \boldsymbol{g}(x_{1:T}) = \sum_{n=1}^{T} \boldsymbol{g}_n(x_{n-1}, x_n).$$

U slučaju lanca je $r_0(x_0) = q_{0\to 1}(x_0) = 1$ i $r_n(x_n) = r_{n\to n}(x_n) = q_{n\to n+1}(x_n)$, pa je algoritam sledeći.

Inicijalizacija: Za svako $x_0 \in \mathbb{X}$, $r_0(x_0) = 1$, odnosno

$$r_0^{(\boldsymbol{\alpha})}(x_0) = \begin{cases} 1, & \boldsymbol{\alpha} = \mathbf{0}; \\ 0, & \mathbf{0} \leq \boldsymbol{\alpha} \leq \boldsymbol{\nu}, \boldsymbol{\alpha} \neq 0. \end{cases} \tag{5.109}$$

Indukcija: Na osnovu formula za procesiranje poruka (5.65)-(5.66) i jednakosti $r_n(x_n) = r_{n\to n}(x_n) = q_{n\to n+1}(x_n)$, dobija se

$$r_{n+1}(x_{n+1}) = \bigoplus_{x_n} \left(\phi_{n+1}(x_n, x_{n+1}) \boldsymbol{g}_{n+1}(x_n, x_{n+1})^{\boldsymbol{\alpha}} \right)_{\boldsymbol{\alpha} \in A_{\boldsymbol{\nu}}} \otimes r_n(x_n),$$

odnosno, posle množenja u binomnom poluprstenu

$$r_{n+1}^{(\boldsymbol{\alpha})}(x_{n+1}) = \sum_{x_n} \sum_{\boldsymbol{\beta}+\boldsymbol{\gamma}=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}, \boldsymbol{\gamma}} \phi_{n+1}(x_n, x_{n+1}) \cdot r_n^{(\boldsymbol{\beta})}(x_n) \cdot \boldsymbol{g}_{n+1}(x_n, x_{n+1}), \tag{5.110}$$

za $1 \leq n \leq T$.

Terminacija: Kros-momenti reda manjeg ili jednakog $\boldsymbol{\nu}$ izračunavaju se u korenu, odnosno u poslednjem čvoru u lancu T, kao

$$\left(\mu_{p,\,\boldsymbol{g}}^{(\boldsymbol{\alpha})} \right)_{\boldsymbol{\alpha} \in A_{\boldsymbol{\nu}}} = \bigoplus_{x_T} r_T(x_T). \tag{5.111}$$

Jednačine (5.109)-(5.111) se poklapaju sa [30], a slične jednačine se, takodje, javljaju i u [2].

Glava 6

Izračunavanje kros-momenata nad *PCFG*

Kros-momenti slučajne vektorske promenljive modelovane probabilističkom kontekstnonezavisnom gramatikom (*probabilistic context-free grammar PCFG*) predstavljaju važne veličine prilikom estimacije parametara promenljive [33]. Kao što smo pomenuli, definišu se kao očekivana vrednost proizvoda celobrojnih stepena koordinata slučajne vektorske promenljive, koja kod *PCFG*-a može da predstavlja: dužinu stringa ili izvodjenja, broj korišćenja pravila u izvodjenju ili neizvesnost pridruženu korišćenim pravilima. Očekivanje se može računati za slučaj kada su elementarni dogadjaji sva izvodjenja gramatike ili za slučaj kada su elementarni dogadjaji izvodjenja gramatike koja generišu unapred zadatu reč iz jezika gramatike. U ovoj glavi, termin *kros-moment* ćemo koristi za prvi slučaj, dok ćemo u drugom slučaju govoriti o *uslovnim kros-momentima*.

Izračunavanje kros-momenata može postati zahtevno ukoliko je skup elementarnih dogadjaja veliki. U prošlosti, ovaj problem je razmatran uglavnom za kros-momente skalarnih promenljivih (jednostavno nazvanih *momenti*), zaključno sa redom dva. Izračunavanje momenata prvog reda, kao što su očekivana dužina izvodjenja ili očekivana dužina reči, razmatrani su u [87]. Izračunavanje entropije za *PCFG* razmatrano je u [65]. Postupak za izračunavanje momenata reči i dužine izvodjenja dat je u [33], gde su izvedene eksplicitne formule za momente reda jedan i dva. Momenti prvog reda razmatrani su u [34], gde je izveden algoritam za izračunavanje entropije za *PCFG*. Nešto generalniji algoritam za izračunavanje kros-momenata reda dva razmatran je u [56].

U ovoj glavi izvešćemo rekurzivne formule za izračunavanje kros-momenata i uslovnih kros-momenata vektorskih slučajnih promenljivih proizvoljnog reda [36]. Najpre, u poglavlju 6.1 dajemo formalnu definiciju *PCFG*. Zatim, u poglavlju 6.2 izvodimo formule za kros-momente diferenciranjem rekurzivnih jednačina za *MGF*, koje su dobijene primenom aloritama za izračunavanje particione funkcije za *PCFG* [66]. Na sličan način, u poglavlju 6.3, izvodimo rekurzivne formule za izračunavanje uslovnih kros-momenata diferenciranjem rekurzivnih jednačina za *MGF*, koje su dobijene primenom *inside* algoritma [53], [26].

6.1 *WCFG* i *PCFG*

Neka je Σ neprazan skup. *Slobodni monoid* nad Σ je monoid $(\Sigma^*, \cdot, \epsilon)$, pri cemu je $\Sigma^* = \{a_1 \dots a_n \,|\, n \in \mathbb{N}_0, a_i \in \Sigma\, (1 \leq i \leq n)\}$ skup svih stringova nad Σ, a ϵ je jedinstveni prazan string dužine nula. Operacija $\cdot$ predstavlja kompoziciju (konkatenaciju) stringova i definisana je sa

$\boldsymbol{u}_1 \cdot \boldsymbol{u}_2 = \boldsymbol{u}_1\boldsymbol{u}_2$ za svako $\boldsymbol{u}_1, \boldsymbol{u}_2 \in \Sigma^*$. U narednom tekstu, slobodni monid nad Σ označavaćemo jednostavno sa Σ^*.

Težinska kontekstno-nezavisna gramatika (weigthed context-free grammar WCFG) nad komutativnim poluprstenom $(\mathbb{K}, +, \cdot, 1, 0)$ je petorka $G = (\Sigma, \mathcal{N}, S, \mathcal{R}, w)$, pri čemu:

- $\Sigma = \{u_1, \ldots, u_{|\Sigma|}\}$ je konačan skup *terminala*,
- $\mathcal{N} = \{A_1, \ldots, A_{|\mathcal{N}|}\}$ je konačan skup *neterminala* uzajamno disjunktan sa Σ,
- $S \in \mathcal{N}$ se naziva *startni simbol* (u daljem tekstu pretpostavljamo da je $S = A_1$),
- $\mathcal{R} \subseteq \mathcal{N} \times (\Sigma \cup \mathcal{N})^*$ je konačan skup pravila. Pravilo $(A, \alpha) \in \mathcal{R}$ zapisujemo kao $A \to \alpha$, pri čvemu neterminal A nazivamo *premisa*. Skup svih pravila $A_i \to B_{i,j}$, $B_{i,j} \in (\mathcal{N} \cup \Sigma)^*$ ćemo označavati sa $\mathcal{R}_i$.
- $w : R \to \mathbb{K}$ je funkcija koja se naziva *težinska funkcija*.

Relacija krajnjeg levog izvodjenja $\Rightarrow$ pridružena gramatici G, definisana je skupom trojki $(\alpha, \pi, \beta) \in (\Sigma \cup \mathcal{N})^* \times \mathcal{R} \times (\Sigma \cup \mathcal{N})^*$ za koje postoje sekvence terminala $\boldsymbol{u} \in \Sigma^*$, i neterminala $\delta \in (\Sigma \cup \mathcal{N})^*$, zajedno sa neterminalom $A \in \mathcal{N}$ i sekvencom $\gamma \in (\Sigma \cup \mathcal{N})^*$ tako da $\alpha = \boldsymbol{u}A\delta, \beta = \boldsymbol{u}\gamma\delta$ i $\pi = A \to \gamma$ je pravilo iz $\mathcal{R}$. Trojka kojom je reprezentovana relacija krajnjeg levog izvodjenja (α, π, β) biće označavana sa $\alpha \overset{\pi}{\Rightarrow} \beta$. *Krajnje levo izvodjenje* (u daljem tekstu *izvodjenje*) u gramatici je sekvenca $\pi_1, \ldots, \pi_n \in \mathcal{R}^*$ za koju postoji simbol u gramatici $\alpha, \beta \in \Sigma \cup \mathcal{N}$, tako da je moguće izvesti β iz α primenom sekvence pravila $\pi_1, \ldots, \pi_n$, $\alpha \overset{\pi_1}{\Rightarrow} \cdots \overset{\pi_n}{\Rightarrow} \beta$. Težinska funkcija se proširuje na izvodjenja, tako da $w(\pi_1 \cdots \pi_N) = w(\pi_1) \cdots w(\pi_N)$, za svako $\pi_1 \cdots \pi_N \in R^*$. Neterminal A je *aktivan* ako postoji izvodjenje $\pi_1 \cdots \pi_k$, tako da $A \overset{\pi_1}{\Rightarrow} \cdots \overset{\pi_k}{\Rightarrow} \boldsymbol{u}, \boldsymbol{u} \in \Sigma^*$. Neteminal A je *dostižan* iz neterminala B ako postoji izvodjenje $\pi_1 \cdots \pi_k$, tako da $B \overset{\pi_1}{\Rightarrow} \cdots \overset{\pi_k}{\Rightarrow} \eta A \xi$, gde je $\eta, \xi \in (\Sigma \cup \mathcal{N})^*$ (ako je A dostižno iz S, onda jednostavno kažemo da je dostižno). Neteminal A je *koristan* ako je dostižan i aktivan (u suprotnom je *beskoristan*). Ukoliko postoji izvodjenje $\pi_1, \ldots, \pi_n \in \mathcal{R}^*$, tako da za neki neterminal važi $A \overset{\pi_1}{\Rightarrow} \cdots \overset{\pi_k}{\Rightarrow} A$, kažemo da gramatika ima ciklus. U protivnom je bez ciklusa.

WCFG $G = (\Sigma, \mathcal{N}, A_1, \mathcal{R}, p)$ nad *sum-product* poluprstenom $(\mathbb{R}_+, +, \cdot, 0, 1)$ naziva se *probabilistička kontekstno-nezavisna gramatika* (*PCFG*) ako težinska funkcija p slika skup pravila u realni interval $[0, 1]$. *PCFG* se naziva *očišćena* ako je $p(A \to \gamma) > 0$ za svako $A \to \gamma \in \mathcal{R}$ i svaki neterminal A, i svi neterminali su korisni. Nadalje razmatramo samo očišćene *PCFG*. Takodje, pretpostavljamo da je težinska funkcija p raspodela verovatnoće nad skupom pravila koja možemo primeniti, t.j. $\sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) = 1$ za svako $1 \le i \le |\mathcal{N}|$.

Za *PCFG* $G = (\Sigma, \mathcal{N}, A_1, \mathcal{R}, p)$ definišemo *podgramatiku* $G_i = (\Sigma, \mathcal{N}_i, A_i, \mathcal{R}_i, p_i)$ sa startnim simbolom A_i, gde je $\mathcal{N}_i'$ skup koji se sastoji od A_i i neterminala dostižnih iz A_i, i $\mathcal{R}_i \subseteq \mathcal{R}$ je skup pravila u kojima se samo neterminali iz $\mathcal{N}_i$ pojavljuju kao premise, i p_i je restrikcija od p na $\mathcal{R}_i$, tako da je $p_i(\pi) = p(\pi)$ za svako $\pi \in \mathcal{R}_i$. Može se primetiti da ako je G očišćena, tada je i G_i očišćena.

6.2 Izračunavanje kros-momenata nad *PCFG*

6.2.1 Konzistentnost *PCFG*

Neka je $G = (\Sigma, \mathcal{N}, A_1, \mathcal{R}, p)$ probabilistička kontekstno-nezavisna gramatika, Ω skup izvodjenja u G, i neka je Ω_i skup izvodjenja iz $A_i \in \mathcal{N}$. Gramatika G je *konzistentna* ako je

$$\sum_{\pi \in \Omega_i} p(\pi) = 1,$$

za svako $1 \leq i \leq |\mathcal{N}|$. *Booth* i *Thompson* [7] su dali uslov za konzistentnost za startni simbol $S = A_1$ sledećom teoremom.

Teorema 6.2.1 *Očišćena PCFG G je konzistentna ako je $\rho(M) < 1$, gde je $\rho(M)$ apsolutna vrednost najveće sopstvene vrednosti matrice očekivanja $M = [M_{i,n}], 1 \leq i, n \leq |\mathcal{N}|$, definisane sa*

$$M_{i,n} = \sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) r_n(i,j), \tag{6.1}$$

gde $r_n(i,j)$ označava broj pojave terminala A_n na desnoj strani pravila $\pi = A_i \to B_{i,j}$.

Primetimo da su matrice očekivanja $\mathrm{M}^{(i)}$ svih podgramatika G_i glavne submatrice matrice M, tako da je u saglasnosti sa posledicom 8.1.20 iz [32] $\rho(\mathrm{M}^{(i)}) \leq \rho(\mathrm{M})$, pa su i gramatike G_i takodje konzistentne

$$\sum_{\pi \in \Omega_i} p(\pi) = 1. \tag{6.2}$$

6.2.2 Kros-momenti i funkcija generatrise momenta nad *PCFG*

Neka su dati *PCFG* $G = (\Sigma, \mathcal{N}, A_1, \mathcal{R}, p)$ i njene podgramatike $G_i = (\Sigma, \mathcal{N}_i, A_i, \mathcal{R}_i, p_i)$. Neka su $\boldsymbol{X}_i$ d-dimenzione slučjane promenljive čije su raspodele restrikcije p na $\mathcal{R}_i$. *MGF* slučajne promenljive $\boldsymbol{X}_i$ nazivaćemo *i - tom funkcijom generatrise momenta* ili *i-tom MGF*, i ona je za svako $i = 1, \ldots, |\mathcal{R}|$, definisana sa

$$M_{p,\boldsymbol{X}_i}(\boldsymbol{t}) = \sum_{\pi \in \Omega_i} p(\pi) e^{\boldsymbol{t}^T \boldsymbol{X}_i(\pi)}, \tag{6.3}$$

gde je $\boldsymbol{t} \in \mathbb{R}^d$. *$i$-ti kros-moment reda* $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_d)$, i definiše kao

$$\mu^{(\nu)}_{p,\boldsymbol{X}_i} = \sum_{\pi \in \Omega_i} p(\pi) \cdot X_{i,1}(\pi)^{\nu_1} \cdots X_{i,D}(\pi)^{\nu_d} = \sum_{\pi \in \Omega_i} p(\pi)\, \boldsymbol{X}(\pi)^{\boldsymbol{\nu}}, \tag{6.4}$$

gde je $\boldsymbol{X}_i(\pi) = \left[X_{i,1}(\pi), \ldots, X_{i,D}(\pi)\right]^T$. Kao što je pokazano u [10], *PCFG* kros-momenti su ograničeni, tako da je moguća zamena mesta operatorima diferenciranja i sumiranja, pa je

$$\mu^{(\nu)}_{p,\boldsymbol{X}_i} = \frac{\partial^{|\boldsymbol{\nu}|} M^{(i)}_{\boldsymbol{X}}(\boldsymbol{t})}{\partial^{\nu_1} t_1 \ldots \partial^{\nu_d} t_d}\Big|_{\boldsymbol{t}=\boldsymbol{0}} = \mathcal{D}_{\boldsymbol{\nu}}\left\{M^{(i)}_{\boldsymbol{X}}\right\}. \tag{6.5}$$

Direktno izračunavanje izraza (6.4) enumerisanjem svih izvodjenja je neefikasno, pošto zahteva $\mathcal{O}(|\Omega|)$ operacija, i postaje praktično nemoguće kada je Ω beskonačan skup. S druge

strane, ako izvedemo izraze za efikasno izračunavanje *MGF*-a (6.3), kros-momenti se mogu izračunati njihovim diferenciranjem.

U daljem tekstu posmatramo promenljive $\boldsymbol{X}_i$ koje mogu biti predstavljene kao suma vektorskih funkcija $\boldsymbol{Y} : \mathcal{R} \to \mathbb{R}$

$$\boldsymbol{X}_i(\pi_1 \cdots \pi_N) = \boldsymbol{Y}\big(\pi_1\big) + \cdots + \boldsymbol{Y}\big(\pi_N\big), \tag{6.6}$$

za svako $\pi_1 \cdots \pi_N \in \Omega$ i svako $ui = 1, \ldots, D$. Tada, za $G_i = \big(\Sigma, \mathcal{N}_i, A_i, \mathcal{R}_i, p\big)$ i $\boldsymbol{X}_i$ možemo konstruisati *MGF gramatiku* za $\boldsymbol{X}_i$ nad $\widetilde{G}_i = \big(\Sigma, \mathcal{N}_i, A_i, \mathcal{R}_i, w\big)$, sa težinskom funkcijom koja uzima vrednosti iz poluprstena stepenih redova $w : \mathcal{R} \to \mathbb{R}\big[\mathbb{N}_0^d\big]$, definisanom pomoću

$$w(\pi) = p(\pi) e^{\boldsymbol{t}^T \boldsymbol{Y}(\pi)} \tag{6.7}$$

za svako $\pi \in \mathcal{R}$. Izvodjenje $\boldsymbol{\pi} = \pi_1 \cdots \pi_N$ u G_i sa težinskom funkcijom $p(\boldsymbol{\pi}) = p(\pi_1) \cdots p(\pi_N)$ je takodje izvodjenje u $\widetilde{G}$ sa težinskom funkcijom

$$w(\boldsymbol{\pi}) = w(\pi_1) \cdots w(\pi_N) = p(\pi_1) e^{\boldsymbol{t}^T \boldsymbol{Y}(\pi_1)} \cdots p(\pi_N) e^{\boldsymbol{t}^T \boldsymbol{Y}(\pi_N)} = p(\boldsymbol{\pi}) e^{\boldsymbol{t}^T \boldsymbol{X}_i(\boldsymbol{\pi})}. \tag{6.8}$$

MGF vektorske slučajne promenljive $\boldsymbol{X}_i$ se sada može izraziti kao suma izvodjenja u $\widetilde{G}$

$$M_{p,\boldsymbol{X}_i}(\boldsymbol{t}) = \sum_{\boldsymbol{\pi} \in \Omega} p(\boldsymbol{\pi}) e^{\boldsymbol{t}^T \boldsymbol{X}_i(\boldsymbol{\pi})} = \sum_{\boldsymbol{\pi} \in \Omega} w(\boldsymbol{\pi}). \tag{6.9}$$

Na taj način, problem izračunavanja *MGF*-a svodi se na problem izračunavanja *particione funkcije* [66] nad poluprstenom $\boldsymbol{\nu}$-neprekidnih funkcija u nuli, čime se bavimo u narednom odeljku.

6.2.3 Izračunavanje kros-momenata nad *PCFG*

Neka je $\widetilde{G} = \big(\Sigma, \mathcal{N}, A_1, \mathcal{R}, w\big)$ težinska kontekstno-nezavisna gramatika nad komutativnim poluprstenom $\big(\mathbb{K}, +, \cdot, 1, 0\big)$ obogaćenog topologijom τ. Pod pretpostavkom da su beskonačne kolekcije $\big\{w(\boldsymbol{\pi})\big\}_{\boldsymbol{\pi} \in \Omega_i}$ sumabilne u τ, i da je moguće primeniti distributivni zakon na beskonačne sume, *particiona funkcija* je funkcija $Z : \mathcal{N} \to \mathbb{K}$, koja svakom neterminalu $A_i \in \mathcal{N}$ dodeljuje sumu

$$Z_i = \sum_{\boldsymbol{\pi} \in \Omega_i} w\big(\boldsymbol{\pi}\big). \tag{6.10}$$

Izdvajanjem prvog simbola iz proizvoda u svakom izvodjenju i korišćenjem distributivnog zakona, particiona funkcija se može izraziti uz pomoć sistema [66]

$$Z_i = \sum_{j=1}^{|\mathcal{R}_i|} w(A_i \to B_{i,j}) \cdot \prod_{k=1}^{|\mathcal{N}|} Z_k^{r_k(B_{i,j})}, \tag{6.11}$$

gde je $1 \le i \le |\mathcal{N}|$.

Neka je sada $\widetilde{G}_i = \big(\Sigma, \mathcal{N}_i, A_i, \mathcal{R}_i, w\big)$ *MGF* gramatika za $\boldsymbol{X}_i : \Omega_i \to$ nad $G = \big(\Sigma, \mathcal{N}_i, A_i, \mathcal{R}, p\big)$, gde je

$$w(\pi) = p(\pi) e^{\boldsymbol{t}^T \boldsymbol{Y}(\pi)}, \tag{6.12}$$

i neka je τ topologija inducirana supremum normom. Saglasno diskusiji u odeljku 6.2.2, ukoliko je

$$\boldsymbol{X}(\pi_1 \cdots \pi_N) = \boldsymbol{Y}(\pi_1) + \cdots + \boldsymbol{Y}(\pi_N). \tag{6.13}$$

vrednost particione funkcije u neterminalu A_i odgovara i-toj *MGF*

$$Z_i = M_{\boldsymbol{X}}^{(i)}, \tag{6.14}$$

a i-ti kros-momenat

$$\mu_{p,\boldsymbol{X}_i}^{(\alpha)} = \mathcal{D}_{\boldsymbol{\alpha}}\{M_{\boldsymbol{X}}^{(i)}\} = \mathcal{D}_{\boldsymbol{\alpha}}\{Z_i\} = \sum_{\boldsymbol{\pi} \in \boldsymbol{\Omega}_i} p(\boldsymbol{\pi}) \boldsymbol{X}(\boldsymbol{\pi})^{\boldsymbol{\alpha}}, \tag{6.15}$$

može se izračunati diferenciranjem izraza (6.11) i rešavanjem rezultujuće jednačine. Primetimo da

$$\mu_{p,\boldsymbol{X}_i}^{(0)} = \mathcal{D}_{\boldsymbol{0}}\{Z_i\} = \Big(\sum_{\boldsymbol{\pi} \in \Omega_i} p(\boldsymbol{\pi}) e^{\boldsymbol{t}^T \boldsymbol{X}(\pi)} \Big)|_{\boldsymbol{t}=\boldsymbol{0}} = \sum_{\boldsymbol{\pi} \in \Omega_i} p(\boldsymbol{\pi}) = 1, \tag{6.16}$$

za svako $1 \leq i \leq |\mathcal{N}|$. Kros-momenti višeg reda mogu se izračunati primenom generalisanog Lajbnicovog pravila (1.7) na izraz (6.11), što rezultuje sledećim sistemom

$$\mu_{p,\boldsymbol{X}_i}^{(\alpha)} = \sum_{j=1}^{|\mathcal{R}_i|} \sum_{\boldsymbol{\beta} \leq \boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}} \mathcal{D}_{\boldsymbol{\alpha}-\boldsymbol{\beta}}\{w(A_i \to B_{i,j})\} \cdot \mathcal{D}_{\boldsymbol{\beta}}\Big\{ \prod_{k=1}^{|\mathcal{N}|} Z_k^{r_k(B_{i,j})} \Big\} \tag{6.17}$$

gde je

$$\mathcal{D}_{\boldsymbol{\alpha}-\boldsymbol{\beta}}\{w(A_i \to B_{i,j})\} = p(A_i \to B_{i,j}) \cdot \boldsymbol{Y}(A_i \to B_{i,j})^{\boldsymbol{\alpha}-\boldsymbol{\beta}}, \tag{6.18}$$

pošto je $w(\pi) = p(\pi) e^{\boldsymbol{t}^T \cdot \boldsymbol{Y}}$ za svako $\pi \in \mathcal{R}$. Na osnovu Lajbnicovog pravila (1.8), dobija se

$$\mathcal{D}_{\boldsymbol{\beta}}\Big\{ \prod_{k=1}^{|\mathcal{N}|} Z_k^{r_k(B_{i,j})} \Big\} = \sum_{\boldsymbol{\gamma}_1 + \cdots + \boldsymbol{\gamma}_{|\mathcal{N}|} = \boldsymbol{\beta}} \binom{\boldsymbol{\beta}}{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{|\mathcal{N}|}} \prod_{k=1}^{|\mathcal{N}|} \mathcal{D}_{\boldsymbol{\gamma}_k}\{Z_k^{r_k(B_{i,j})}\} \tag{6.19}$$

i

$$\mathcal{D}_{\boldsymbol{\gamma}_k}\{Z_k^{r_k(B_{i,j})}\} = \mathcal{D}_{\boldsymbol{\gamma}_k}\Big\{ \prod_{l=1}^{r_k(B_{i,j})} Z_k \Big\} = \sum_{\boldsymbol{\delta}_1 + \cdots + \boldsymbol{\delta}_{r_k(B_{i,j})} = \boldsymbol{\gamma}_k} \binom{\boldsymbol{\gamma}_k}{\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{r_k(B_{i,j})}} \prod_{l=1}^{r_k(B_{i,j})} \mu_{p,\boldsymbol{X}_k}^{(\delta_l)}. \tag{6.20}$$

Zamenom (6.20) i (6.19) u (6.17), dobija se

$$\mu_{p,\boldsymbol{X}_i}^{(\alpha)} = \sum_{j=1}^{|\mathcal{R}_i|} \sum_{\boldsymbol{\beta} \leq \boldsymbol{\alpha}} Q_{i,j}(\boldsymbol{\alpha}, \boldsymbol{\beta}), \tag{6.21}$$

gde je

$$Q_{i,j}(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}} p(A_i \to B_{i,j}) \cdot \boldsymbol{Y}(A_i \to B_{i,j})^{\boldsymbol{\alpha}-\boldsymbol{\beta}} \cdot \sum_{\boldsymbol{\gamma}_1 + \cdots + \boldsymbol{\gamma}_{|\mathcal{N}|} = \boldsymbol{\beta}} \binom{\boldsymbol{\beta}}{\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_{|\mathcal{N}|}} \prod_{k=1}^{|\mathcal{N}|} \sum_{\boldsymbol{\delta}_1 + \cdots + \boldsymbol{\delta}_{r_k(B_{i,j})} = \boldsymbol{\gamma}_k} \binom{\boldsymbol{\gamma}_k}{\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_{r_k(B_{i,j})}} \prod_{l=1}^{r_k(B_{i,j})} \mu_{p,\boldsymbol{X}_k}^{(\delta_l)}. \tag{6.22}$$

Sistem (6.21) može se rešiti razbijanjem izraza na dva dela: jedan koji zavisi i drugi koji ne zavisi od $\mu^{(\alpha)}_{p,\boldsymbol{X}_i}$

$$\mu^{(\alpha)}_{p,\boldsymbol{X}_i} = \sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}(\boldsymbol{\alpha},\boldsymbol{\alpha}) + \sum_{j=1}^{|\mathcal{R}_i|} \sum_{\boldsymbol{\beta}<\boldsymbol{\alpha}} Q_{i,j}(\boldsymbol{\alpha},\boldsymbol{\beta}), \tag{6.23}$$

gde je

$$Q_{i,j}(\boldsymbol{\alpha},\boldsymbol{\alpha}) = p(A_i \to B_{i,j}) \cdot W_{i,j}(\boldsymbol{\alpha}) \tag{6.24}$$

i

$$W_{i,j}(\boldsymbol{\alpha}) = \sum_{\gamma_1+\cdots+\gamma_{|\mathcal{N}|}=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}} \prod_{k=1}^{|\mathcal{N}|} \sum_{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_k(B_{i,j})}=\boldsymbol{\gamma}_k} \binom{\boldsymbol{\gamma}_k}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_k(B_{i,j})}} \prod_{l=1}^{r_k(B_{i,j})} \mu^{(\delta_l)}_{p,\boldsymbol{X}_k}. \tag{6.25}$$

Dalje, ako stavimo

$$H_{i,j}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}) = \binom{\boldsymbol{\alpha}}{\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}} \prod_{k=1}^{|\mathcal{N}|} \sum_{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_k(B_{i,j})}=\boldsymbol{\gamma}_k} \binom{\boldsymbol{\gamma}_k}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_k(B_{i,j})}} \prod_{l=1}^{r_k(B_{i,j})} \mu^{(\delta_l)}_{p,\boldsymbol{X}_k} \tag{6.26}$$

izraz za $W_{\boldsymbol{\alpha}}(B_{i,j})$, može se transformisati u

$$W_{i,j}(\boldsymbol{\alpha}) = \sum_{n=1}^{|\mathcal{N}|} H^{(n)}_{i,j}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}) + \sum_{\substack{\gamma_1+\cdots+\gamma_{|\mathcal{N}|}=\boldsymbol{\alpha} \\ \gamma_1,\ldots,\gamma_{|\mathcal{N}|}<\boldsymbol{\alpha}}} H_{i,j}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}), \tag{6.27}$$

gde $H^{(n)}_{i,j}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|})$ označava $H_{i,j}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|})$ sa $\boldsymbol{\gamma}_n = \boldsymbol{\alpha}$ i ostalim $\boldsymbol{\gamma}$-ama jednakim nuli, što je, na osnovu (6.26)

$$H^{(n)}_{i,j}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}) = \sum_{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(B_{i,j})}=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}} \prod_{l=1}^{r_n(B_{i,j})} \mu^{(\delta_l)}_{p,\boldsymbol{X}_n} \cdot \prod_{\substack{k=1 \\ k\neq n}}^{|\mathcal{N}|} \prod_{l=1}^{r_k(B_{i,j})} \mu^{(0)}_{p,\boldsymbol{X}_k}. \tag{6.28}$$

Konačno, pošto iskoristimo $\mu^{(0)}_{p,\boldsymbol{X}_k} = 1$, dobija se

$$H^{(n)}_{i,j}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}) = \sum_{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(B_{i,j})}=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}} \prod_{l=1}^{r_n(B_{i,j})} \mu^{(\delta_l)}_{p,\boldsymbol{X}_n}, \tag{6.29}$$

što se, slično kao kod izraza (6.23), može napisati kao

$$\begin{aligned} H^{(n)}_{i,j}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}) &= \sum_{s=1}^{r_n(B_{i,j})} \mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{X}_n} + \sum_{\substack{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_k(B_{i,j})}=\boldsymbol{\alpha} \\ \boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}<\boldsymbol{\alpha}}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}} \prod_{l=1}^{r_n(B_{i,j})} \mu^{(\delta_l)}_{p,\boldsymbol{X}_m} = \\ &= r_n(B_{i,j}) \cdot \mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{X}_n} + \sum_{\substack{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(B_{i,j})}=\boldsymbol{\alpha} \\ \boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}<\boldsymbol{\alpha}}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}} \prod_{l=1}^{r_n(B_{i,j})} \mu^{(\delta_l)}_{p,\boldsymbol{X}_n}. \end{aligned} \tag{6.30}$$

Zamenom (6.30) u (6.27) dobija se

$$W_{i,j}(\boldsymbol{\alpha}) = \sum_{n=1}^{|\mathcal{N}|} r_n(B_{i,j}) \cdot \mu_{p,\boldsymbol{X}_n}^{(\boldsymbol{\alpha})} + \sum_{n=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(B_{i,j})}=\boldsymbol{\alpha} \\ \boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}<\boldsymbol{\alpha}}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}} \prod_{l=1}^{r_n(B_{i,j})} \mu_{p,\boldsymbol{X}_n}^{(\boldsymbol{\delta}_l)} + \sum_{\substack{\boldsymbol{\gamma}_1+\cdots+\boldsymbol{\gamma}_{|\mathcal{N}|}=\boldsymbol{\alpha} \\ \boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}<\boldsymbol{\alpha}}} H_{i,j}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}). \quad (6.31)$$

Dalje, zamenom (6.31) i (6.24) u (6.23), kros-momenat se može izraziti kao

$$\mu_{p,\boldsymbol{X}_i}^{(\boldsymbol{\alpha})} = \sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} r_n(B_{i,j}) \cdot \mu_{p,\boldsymbol{X}_n}^{(\boldsymbol{\alpha})} + \sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(B_{i,j})}=\boldsymbol{\alpha} \\ \boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}<\boldsymbol{\alpha}}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}} \prod_{l=1}^{r_n(B_{i,j})} \mu_{p,\boldsymbol{X}_n}^{(\boldsymbol{\delta}_l)} + \sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{\substack{\boldsymbol{\gamma}_1+\cdots+\boldsymbol{\gamma}_{|\mathcal{N}|}=\boldsymbol{\alpha} \\ \boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}<\boldsymbol{\alpha}}} H_{i,j}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}) + \sum_{j=1}^{|\mathcal{R}_i|} \sum_{\boldsymbol{\beta}<\boldsymbol{\alpha}} Q_{i,j}(\boldsymbol{\alpha},\boldsymbol{\beta}), \quad (6.32)$$

gde su $H_{i,j}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|})$ i $Q_{i,j}(\boldsymbol{\alpha},\boldsymbol{\beta})$ dati sa (6.26) i (6.22). Konačno, ako uvedemo

$$c_i^{(\boldsymbol{\alpha})} = \sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(B_{i,j})}=\boldsymbol{\alpha} \\ \boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}<\boldsymbol{\alpha}}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}} \prod_{l=1}^{r_n(B_{i,j})} \mu_{p,\boldsymbol{X}_n}^{(\boldsymbol{\delta}_l)} + \sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{\substack{\boldsymbol{\gamma}_1+\cdots+\boldsymbol{\gamma}_{|\mathcal{N}|}=\boldsymbol{\alpha} \\ \boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}<\boldsymbol{\alpha}}} H_{i,j}(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|\mathcal{N}|}) + \sum_{j=1}^{|\mathcal{R}_i|} \sum_{\boldsymbol{\beta}<\boldsymbol{\alpha}} Q_{i,j}(\boldsymbol{\alpha},\boldsymbol{\beta}), \quad (6.33)$$

jednačina (6.32) se može sažeto zapisati kao

$$\mu_{p,\boldsymbol{X}_i}^{(\boldsymbol{\alpha})} = \sum_{n=1}^{|\mathcal{N}|} \mathrm{M}_{i,n} \cdot \mu_{p,\boldsymbol{X}_n}^{(\boldsymbol{\alpha})} + c_i^{(\boldsymbol{\alpha})}, \quad (6.34)$$

ili u matričnoj formi

$$\mathrm{m}^{(\boldsymbol{\alpha})} = \mathrm{M} \cdot \mathrm{m}^{(\boldsymbol{\alpha})} + \boldsymbol{c}^{(\boldsymbol{\alpha})}, \quad (6.35)$$

gde je $\mathrm{m}^{(\boldsymbol{\alpha})} = \left[\mu_{p,\boldsymbol{X}_1}^{(\alpha)},\ldots,\mu_{p,\boldsymbol{X}_{|\mathcal{N}|}}^{(\alpha)}\right]^T$ *vektor kros-momenata*, $\boldsymbol{c}^{(\boldsymbol{\alpha})} = \left[c_1^{(\boldsymbol{\alpha})},\ldots,c_{|\mathcal{N}|}^{(\boldsymbol{\alpha})}\right]^T$ i M je matrica momenata, definisana u teoremi 6.2.1. Pošto je uslov $\rho(\mathrm{M}) < 1$ iz teoreme 6.2.1 zadovoljen, matrica I – M je invertibilna, i matrična jednačina ima jedinstveno rešenje, dato sa

$$\mathrm{m}^{(\boldsymbol{\alpha})} = \left(\mathrm{I}-\mathrm{M}\right)^{-1} \boldsymbol{c}^{(\boldsymbol{\alpha})}. \quad (6.36)$$

Ukoliko je izračunat inverz $\left(\mathrm{I}-\mathrm{M}\right)^{-1}$, koji ne zavisi od $\boldsymbol{\alpha}$, kros-momenat je u potpunosti odredjen članom $\boldsymbol{c}^{(\boldsymbol{\alpha})}$, koji zavisi od svih kros-momenata reda manjeg od $\boldsymbol{\alpha}$ i može se izračunati pomoću (6.33). U narednim odeljcima izvešćemo izraz za $\boldsymbol{c}^{(\boldsymbol{\alpha})}$ za skalarne promenljive reda manjeg ili jednakog dva, i izvodimo izraze za momente prvog i drugog reda date u [7] i [33], kao specijalne slučajeve jednačine (6.36).

6.2.4 Momenti prvog reda

U slučaju momenata prvog reda $\boldsymbol{\alpha} = 1$, izraz (6.15) svodi se na očekivanje promenljive $\boldsymbol{X}_i$

$$\mu^{(1)}_{p,\boldsymbol{X}_i} = \sum_{\boldsymbol{\pi}\in\Omega_i} p(\boldsymbol{\pi})\boldsymbol{X}_i(\boldsymbol{\pi}). \tag{6.37}$$

Vektor momenata $\mathrm{m}^{(\boldsymbol{\alpha})} = \left[\mu^{(1)}_{p,\boldsymbol{X}_1}, \ldots, \mu^{(1)}_{p,\boldsymbol{X}_{|\mathcal{N}|}}\right]$ izračunava se kao u jednačini (6.36)

$$\mathrm{m}^{(1)} = \left(\mathrm{I} - \mathrm{M}\right)^{-1}\boldsymbol{c}^{(1)}, \tag{6.38}$$

gde je $c^{(1)} = \left[c^{(1)}_1, \ldots, c^{(1)}_{|\mathcal{N}|}\right]^T$. Prva i druga suma u izrazu (6.33) za $c^{(\boldsymbol{\alpha})}_i$ svode se na $c^{(1)}_i = \sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}(1,0)$, ili, posle primene izraza (6.22) za $Q_{i,j}(\boldsymbol{\alpha},\boldsymbol{\beta})$, na

$$c^{(1)}_i = \sum_{j=1}^{|\mathcal{R}_i|} p\left(A_i \to B_{i,j}\right)\cdot \boldsymbol{\Upsilon}\left(A_i \to B_{i,j}\right). \tag{6.39}$$

Neka je $\pi_1\cdots\pi_N$ izvodjenje koje počinje sa startnim simbolom A_1 i završava sa se rečju $\boldsymbol{u} \in \Sigma^*$. Ako stavimo $\boldsymbol{\Upsilon}\left(A_i \to B_{i,j}\right) = 1$, na osnovu (6.13), dobijamo $\boldsymbol{X}(\pi_1\cdots\pi_N) = \sum_{n=1}^{N} \boldsymbol{\Upsilon}\left(\pi_n\right) = N$, t.j., $\boldsymbol{X}$ je dužina izvodjenja. Na osnovu izraza (6.37), momenat $\mu^{(1)}_{p,\boldsymbol{X}_1}$ predstavlja očekivanu dužinu izvodjenja, što se poklapa sa [7] i [33].

Slično, ako stavimo $\boldsymbol{\Upsilon}\left(A_i \to B_{i,j}\right) = \sum_{n=1}^{|\Sigma|} t_n(i,j)$, gde $t_n(i,j)$ označava broj terminala u sekvenci $B_{i,j}$, promenljiva $\boldsymbol{X}(\pi_1\cdots\pi_N)$ svodi se na dužinu reči izvedene iz $\pi_1\cdots\pi_N$. U ovom slučaju, momenat $\mu^{(1)}_{p,\boldsymbol{X}_1}$ svodi se na očekivanu dužinu reči, a formula (6.39) svodi se na jednakosti iz [7].

6.2.5 Momenti drugog reda

Formula za momente drugog reda je nešto komplikovanija. Za $\boldsymbol{\alpha} = 2$, $c^{(\boldsymbol{\alpha})}_i$ svodi se na

$$c^{(2)}_i = \sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(B_{i,j})}=\boldsymbol{\alpha} \\ \boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}<2}} \binom{2}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}} \prod_{l=1}^{r_n(B_{i,j})} \mu^{(\boldsymbol{\delta}_l)}_{p,\boldsymbol{X}_n} +$$

$$\sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{\substack{\gamma_1+\cdots+\gamma_{|R|}=2 \\ \gamma_1,\ldots,\gamma_{|R|}<2}} H_{i,j}\left(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_{|R|}\right) + \sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}\left(2,0\right) + \sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}\left(2,1\right). \tag{6.40}$$

Prva suma u prethodnom izrazu može se transformisati na

$$\sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{\delta}_1+\cdots+\boldsymbol{\delta}_{r_n(B_{i,j})}=2 \\ \boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}<2}} \binom{2}{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_{r_n(B_{i,j})}} \prod_{l=1}^{r_n(B_{i,j})} \mu^{(\boldsymbol{\delta}_l)}_{p,\boldsymbol{X}_n} =$$

$$\sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} r_n(B_{i,j})\left(r_n(B_{i,j}) - 1\right)\cdot\left(\mu^{(1)}_{p,\boldsymbol{X}_n}\right)^2. \tag{6.41}$$

Da bismo izračunali drugu sumu, uvodimo $H_{i,j}^{(a,b)}(\gamma_1,\ldots,\gamma_{|\mathcal{N}|})$, što je jednako $H_{i,j}(\gamma_1,\ldots,\gamma_{|\mathcal{N}|})$ za $\gamma_a = \gamma_b = 1$, i svim ostalim γ-ma jednakim nuli. Dobijamo

$$H_{i,j}^{(a,b)}(\gamma_1,\ldots,\gamma_{|\mathcal{N}|}) = 2 \cdot \sum_{\delta_1+\cdots+\delta_{r_a(B_{i,j})}=\gamma_a} \binom{\gamma_a}{\delta_1,\ldots,\delta_{r_a(B_{i,j})}} \prod_{l=1}^{r_a(B_{i,j})} \mu_{p,\boldsymbol{X}_k}^{(\delta_l)}$$
$$\sum_{\delta_1+\cdots+\delta_{r_b(B_{i,j})}=\gamma_b} \binom{\gamma_b}{\delta_1,\ldots,\delta_{r_b(B_{i,j})}} \prod_{l=1}^{r_b(B_{i,j})} \mu_{p,\boldsymbol{X}_a}^{(\delta_l)} \cdot \prod_{\substack{k=1\\k\neq a,b}}^{|\mathcal{N}|} \sum_{\delta_1+\cdots+\delta_{r_k(B_{i,j})}=\gamma_k} \binom{\gamma_k}{\delta_1,\ldots,\delta_{r_k(B_{i,j})}} \prod_{l=1}^{r_k(B_{i,j})} \mu_{p,\boldsymbol{X}_b}^{(\delta_l)}, \quad (6.42)$$

i

$$H_{i,j}^{(a,b)}(\gamma_1,\ldots,\gamma_{|\mathcal{N}|}) = 2 \cdot \sum_{c=1}^{r_a(B_{i,j})} \mu_{p,\boldsymbol{X}_k}^{(1)} \cdot \sum_{d=1}^{r_b(B_{i,j})} \mu_{p,\boldsymbol{X}_k}^{(1)} = 2 \cdot r_a(B_{i,j}) \cdot r_b(B_{i,j}) \cdot \mu_{p,\boldsymbol{X}_a}^{(1)} \mu_{p,\boldsymbol{X}_b}^{(1)}. \quad (6.43)$$

Druga suma iz izraza (6.40) postaje

$$\sum_{j=1}^{|R_i|} p(A_i \to B_{i,j}) \sum_{\substack{\gamma_1+\cdots+\gamma_{|\mathcal{N}|}=2\\ \gamma_1,\ldots,\gamma_{|\mathcal{N}|}<2}} H_{i,j}(\gamma_1,\ldots,\gamma_{|\mathcal{N}|}) = \sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{a=1}^{|\mathcal{N}|} \sum_{b=a+1}^{|\mathcal{N}|} H_{i,j}^{(a,b)}(\gamma_1,\ldots,\gamma_{|\mathcal{N}|}) =$$
$$= 2 \cdot \sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{a=1}^{|\mathcal{N}|} \sum_{b=a+1}^{|\mathcal{N}|} r_a(B_{i,j}) \cdot r_b(B_{i,j}) \cdot \mu_{p,\boldsymbol{X}_a}^{(1)} \mu_{p,\boldsymbol{X}_b}^{(1)} =$$
$$= \sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{a=1}^{|\mathcal{N}|} \sum_{b=1}^{|\mathcal{N}|} r_a(B_{i,j}) \cdot r_b(B_{i,j}) \cdot \mu_{p,\boldsymbol{X}_a}^{(1)} \mu_{p,\boldsymbol{X}_b}^{(1)} - \sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} r_n(B_{i,j})^2 \left(\mu_{p,\boldsymbol{X}_n}^{(1)}\right)^2. \quad (6.44)$$

Sada, (6.40) se svodi na

$$c_i^{(2)} = CR_i + \sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}(2,0) + \sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}(2,1), \quad (6.45)$$

gde je

$$CR_i = \sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{a=1}^{|\mathcal{N}|} \sum_{b=1}^{|\mathcal{N}|} r_a(B_{i,j}) \cdot r_b(B_{i,j}) \cdot \mu_{p,\boldsymbol{X}_a}^{(1)} \mu_{p,\boldsymbol{X}_b}^{(1)} - \sum_{j=1}^{|\mathcal{R}_i|} p(A_i \to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} r_n(B_{i,j}) \left(\mu_{p,\boldsymbol{X}_n}^{(1)}\right)^2, \quad (6.46)$$

i

$$Q_{i,j}(2,0) = p(A_i \to B_{i,j}) \cdot Y(A_i \to B_{i,j})^2, \quad (6.47)$$

$$Q_{i,j}(2,1) = 2 \cdot p(A_i \to B_{i,j}) \cdot Y(A_i \to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} \sum_{a=1}^{r_n(B_{i,j})} \mu_{p,\boldsymbol{X}_n}^{(1)} =$$
$$= 2 \cdot p(A_i \to B_{i,j}) \cdot Y(A_i \to B_{i,j}) \sum_{n=1}^{|\mathcal{N}|} r_n(B_{i,j}) \mu_{p,\boldsymbol{X}_n}^{(1)}. \quad (6.48)$$

Ako stavimo $Y\big(A_i \to B_{i,j}\big) = 1$ za svako $A_i \to B_{i,j} \in \mathcal{R}$, $\boldsymbol{X}_1$ se svodi na dužinu izvodjenja. Formula za izračunavanje momenata drugog reda dužine izvodjenja, data u [33], može se izvesti iz jednačine (6.45), pošto je

$$\sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}(2,0) = \sum_{j=1}^{|\mathcal{R}_i|} p\big(A_i \to B_{i,j}\big) \cdot Y\big(A_i \to B_{i,j}\big)^2 = \sum_{j=1}^{|\mathcal{R}_i|} p\big(A_i \to B_{i,j}\big) = 1, \tag{6.49}$$

$$\sum_{j=1}^{|\mathcal{R}_i|} Q_{i,j}(2,1) = 2 \cdot \sum_{n=1}^{|\mathcal{N}|} \Big(\sum_{j=1}^{|\mathcal{R}_i|} p\big(A_i \to B_{i,j}\big) \cdot r_n(B_{i,j}) \Big) \mu^{(1)}_{p,\boldsymbol{X}_n} = 2 \cdot \sum_{n=1}^{|\mathcal{N}|} e_{i,n} \mu^{(1)}_{p,\boldsymbol{X}_n} = 2 \cdot \mu^{(1)}_{p,\boldsymbol{X}_n} - 2, \tag{6.50}$$

gde poslednja jednačina sledi iz (6.36), i

$$c_i^{(2)} = CR_i + 2 \cdot \mu^{(1)}_{p,\boldsymbol{X}_n} - 1. \tag{6.51}$$

Konačno, zamenom (6.51) u (6.36) dobija se

$$\mathrm{m}^{(2)}\{\boldsymbol{X}\} = \big(\mathrm{I} - \mathrm{M}\big)^{-1} \cdot \Big(\mathrm{CR}_i + 2 \cdot \mathrm{m}^{(1)} - \mathbf{1}\Big), \tag{6.52}$$

gde je $\mathrm{CR}_i = \big[CR_1, \ldots, CR_{|\mathcal{N}|}\big]$ i $\mathbf{1} = \big[1, \ldots, 1\big]$, što se poklapa sa rezultatima iz [33].

6.3 Izračunavanje uslovnih kros-momenata nad *PCFG*

6.3.1 Uslovni kros-momenti i funkcija generatrise uslovnih momenata nad *PCFG*

Neka su dati *PCFG* $G = \big(\Sigma, \mathcal{N}, A_1, \mathcal{R}, p\big)$ i njene podgramatike $G_i = \big(\Sigma, \mathcal{N}_i, A_i, \mathcal{R}_i, p_i\big)$. Neka je $\boldsymbol{\Omega}_i(\boldsymbol{u})$ skup izvodjenja koji startuju u $A_i \in \mathcal{N}$ i završavaju se u $\boldsymbol{u} \in \Sigma^*$. Neka su $\boldsymbol{X}_i$ d-dimenzione slučajne promenljive čije su raspodele restrikcije p na $\mathcal{R}_i$. *I-ta funkcija generatrise uslovnih momenata* ili *i-ta uslovna MGF* od $\boldsymbol{X}_i$, je za svako $i = 1, \ldots, |\mathcal{R}|$, definisana sa

$$M_{p,\boldsymbol{X}_i|\boldsymbol{u}}(\boldsymbol{t}) = \sum_{\boldsymbol{\pi} \in \Omega_i(\boldsymbol{u})} p(\boldsymbol{\pi}) e^{\boldsymbol{t}^T \boldsymbol{X}_i(\boldsymbol{\pi})}, \tag{6.53}$$

gde je $\boldsymbol{t} \in \mathbb{R}^d$. *I-ti uslovni kros-moment reda* $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_d)$ se definiše kao

$$\mu^{(\nu)}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = \sum_{\boldsymbol{\pi} \in \Omega_i(\boldsymbol{u})} p(\boldsymbol{\pi}) \cdot X_{i,1}(\boldsymbol{\pi})^{\nu_1} \cdots X_{i,D}(\boldsymbol{\pi})^{\nu_d} = \sum_{\boldsymbol{\pi} \in \Omega_i(\boldsymbol{u})} p(\boldsymbol{\pi})\, \boldsymbol{X}(\boldsymbol{\pi})^{\boldsymbol{\nu}}, \tag{6.54}$$

gde je $\boldsymbol{X}_i(\boldsymbol{\pi}) = \big[X_{i,1}(\boldsymbol{\pi}), \ldots, X_{i,D}(\boldsymbol{\pi})\big]^T$. Kao i u poglavlju 6.2.3, ukoliko izvedemo izraze za efikasno izračunavanje i-te uslovne *MGF* (6.53), i-ti uslovni kros-momenti mogu se izračunati njihovim diferenciranjem

$$\mu^{(\nu)}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = \frac{\partial^{|\boldsymbol{\nu}|} M^{(i)}_{\boldsymbol{X}}(\boldsymbol{t})}{\partial^{\nu_1} t_1 \ldots \partial^{\nu_d} t_d}\Big|_{\boldsymbol{t}=\boldsymbol{0}} = \mathcal{D}_{\boldsymbol{\nu}}\Big\{M^{(i)}_X\Big\}. \tag{6.55}$$

U daljem tekstu posmatramo promenljive $\boldsymbol{X}_i$ koje mogu biti predstavljene kao suma vektorskih funkcija $\boldsymbol{Y} : \mathcal{R} \to \mathbb{R}$

$$\boldsymbol{X}_i(\pi_1 \cdots \pi_N) = \boldsymbol{Y}\big(\pi_1\big) + \cdots + \boldsymbol{Y}\big(\pi_N\big), \tag{6.56}$$

za svako $\pi_1 \cdots \pi_N \in \Omega$. Tada za $G_i = \left(\Sigma, \mathcal{N}_i, A_i, \mathcal{R}_i, p\right)$ i $\boldsymbol{X}_i$ možemo konstruisati gramatiku i-te uslovne *MGF* za $\boldsymbol{X}_i$ nad $\widetilde{G}_i = \left(\Sigma, \mathcal{N}_i, A_i, \mathcal{R}_i, w\right)$, sa težinskom funkcijom koja uzima vrednosti iz poluprstena stepenih redova $w : \mathcal{R} \to \mathbb{R}\left[\mathbb{N}_0^d\right]$, definisanom pomoću

$$w(\pi) = p(\pi) e^{\boldsymbol{t}^T \boldsymbol{Y}(\pi)}, \tag{6.57}$$

za svako $\pi \in \mathcal{R}$. Izvodjenje $\boldsymbol{\pi} = \pi_1 \cdots \pi_N$ u G_i sa težinskom funkcijom $p(\boldsymbol{\pi}) = p(\pi_1) \cdots p(\pi_N)$ je takodje izvodjenje u $\widetilde{G}$, sa težinskom funkcijom

$$w(\boldsymbol{\pi}) = w(\pi_1) \cdots w(\pi_N) = p(\pi_1) e^{\boldsymbol{t}^T \boldsymbol{Y}(\pi_1)} \cdots p(\pi_N) e^{\boldsymbol{t}^T \boldsymbol{Y}(\pi_N)} = p(\boldsymbol{\pi}) e^{\boldsymbol{t}^T \boldsymbol{X}_i(\boldsymbol{\pi})}. \tag{6.58}$$

I-ta uslovna *MGF* vektorske slučajne promenljive $\boldsymbol{X}_i$ se sada može izraziti kao suma izvodjenja u $\widetilde{G}$

$$M_{p,\boldsymbol{X}_i|\boldsymbol{u}}(\boldsymbol{t}) = \sum_{\boldsymbol{\pi} \in \Omega(\boldsymbol{u})} p(\boldsymbol{\pi}) e^{\boldsymbol{t}^T \boldsymbol{X}_i(\boldsymbol{\pi})} = \sum_{\boldsymbol{\pi} \in \Omega(\boldsymbol{u})} w(\boldsymbol{\pi}). \tag{6.59}$$

Tako se izračunavanje uslovne *MGF* može izvršiti pomoću *inside algoritma* [26] nad poluprstenom $\boldsymbol{\nu}$-neprekidnih funkcija u nuli, što pokazujemo u narednom odeljku.

6.3.2 Izračunavanje uslovnih kros-momenata *PCFG*

Neka je $\widetilde{G} = \left(\Sigma, \mathcal{N}, A_1, \mathcal{R}, w\right)$ *WCFG* nad komutativnim poluprstenom $\left(\mathbb{K}, +, \cdot, 1, 0\right)$, i neka je $\Omega_i(\boldsymbol{u})$ skup svih izvodjenja koja počinju neterminalom A_i, a završavaju se stringom $\boldsymbol{u} \in \Sigma^*$. *Inside težina* u $\widetilde{G}$ je funkcija $\sigma_i : \mathcal{N} \times \Sigma^* \to C_{\boldsymbol{\nu}}$ i definiše se kao suma svih izvodjenja iz $\Omega_i(\boldsymbol{u})$

$$\sigma_i\left(\boldsymbol{u}\right) = \sum_{\boldsymbol{\pi} \in \Omega_i(\boldsymbol{u})} w\left(\boldsymbol{\pi}\right), \tag{6.60}$$

za $1 \leq i \leq |\mathcal{R}|$ i $\boldsymbol{u} \in \Sigma^*$. Neka je $A_i \to B_{i,j} \in \mathcal{R}$ i

$$B_{i,j} = \boldsymbol{v}_1 A_{i_1} \boldsymbol{v}_2 A_{i_2} \cdots \boldsymbol{v}_k A_{i_k} v_{k+1}, \tag{6.61}$$

gde je $\boldsymbol{v}_i \in \Sigma^*$ i $A_{i_n} \in \mathcal{N}$. *Inside* težina za gramatiku bez ciklusa može se izračunati pomoću *inside* algoritma [26], [81], koji je dat rekurzivnim jednačinama

$$\sigma_i(\boldsymbol{u}) = \sum_{j=1}^{|\mathcal{R}_i|} \sum_{\substack{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_k \in \Sigma^* \\ \boldsymbol{u} = \boldsymbol{v}_1 \boldsymbol{u}_1 \boldsymbol{v}_2 \cdots \boldsymbol{v}_k \boldsymbol{u}_k \boldsymbol{v}_{k+1}}} w(A_i \to B_{i,j}) \cdot \prod_{j=1}^{k} \sigma_{i_j}(\boldsymbol{u}_j), \tag{6.62}$$

sa baznim slučajem koji se sastoji iz pravila $A_i \to \boldsymbol{u}$, u kojima se jedino stringovi $\boldsymbol{u} \in \Sigma^*$ pojavljuju na desnoj strani izvodjenja

$$\sigma_i\left(\boldsymbol{u}\right) = w\left(A_i \to \boldsymbol{u}\right). \tag{6.63}$$

Neka je $\widetilde{G} = \left(\Sigma, \mathcal{N}, A_1, \mathcal{R}, w\right)$ gramatika prve uslovne *MGF* za $G = \left(\Sigma, \mathcal{N}, A_1, \mathcal{R}, p\right)$ definisana u odeljku 6.3.1. Tada je uslovna *MGF* $M_{p,\boldsymbol{X}_i|\boldsymbol{u}}(\boldsymbol{t})$ *inside* težina u poluprstenu stepenih redova

$$\sigma_i\left(\boldsymbol{u}\right) = \sum_{\boldsymbol{\alpha} \in \mathbb{N}_0^{\mathrm{d}}} \frac{\mu_{p,\boldsymbol{X}_i|\boldsymbol{u}}^{(\boldsymbol{\alpha})}}{\boldsymbol{\alpha}!} \cdot \boldsymbol{t}^{(\boldsymbol{\alpha})}, \tag{6.64}$$

pa se može izračunati uz pomoć rekurzivne jednačine date izrazima (6.62) i (6.63). S druge strane

$$M_{p,\boldsymbol{X}_i|\boldsymbol{u}}(\boldsymbol{t}) = \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \frac{\mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{X}_i|\boldsymbol{u}}}{\boldsymbol{\alpha}!}\cdot \boldsymbol{t}^{(\boldsymbol{\alpha})}. \tag{6.65}$$

pa se uslovni kros-momenti mogu dobiti primenom preslikavanja $\mathcal{B}^{(\boldsymbol{\nu})} : \mathbb{R}(\boldsymbol{t}) \to \mathbb{R}^{|A_{\boldsymbol{\nu}}|}$ na $\sigma_i(\boldsymbol{u})$, pri čemu je preslikavanje definisano sa

$$\mathcal{B}^{(\boldsymbol{\nu})}\Big\{ \sum_{\boldsymbol{\alpha}\in\mathbb{N}_0^{\mathrm{d}}} \frac{z^{(\boldsymbol{\alpha})}}{\boldsymbol{\alpha}!}\cdot \boldsymbol{t}^{(\boldsymbol{\alpha})} \Big\} = \big(z^{(\boldsymbol{\alpha})} \big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}}, \tag{6.66}$$

pa je

$$\mathcal{B}^{(\boldsymbol{\nu})}\big\{\sigma_i(\boldsymbol{u})\big\} = \big(\mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{X}_i|\boldsymbol{u}} \big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}}. \tag{6.67}$$

Preslikavanje $\mathcal{B}^{(\boldsymbol{\nu})}$ slika poluprsten stepenih redova u binomni poluprsten reda $\boldsymbol{\nu}$, koji je definisan kao petorka $(\mathbb{R}^{|\mathcal{A}_{\boldsymbol{\nu}}|}, \oplus, \otimes, \mathbf{0}, \mathbf{1})$, gde su $\oplus$ i $\otimes$ definisani sa

$$u \oplus v = \Big(u^{(\boldsymbol{\alpha})} + v^{(\boldsymbol{\alpha})} \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}}, \tag{6.68}$$

$$u \otimes v = \Big(\sum_{\boldsymbol{\beta}\le\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}} u^{(\boldsymbol{\beta})}\cdot v^{(\boldsymbol{\alpha}-\boldsymbol{\beta})} \Big)_{\boldsymbol{\alpha}\in A_{\boldsymbol{\nu}}}, \tag{6.69}$$

za svako $u, v \in \mathbb{R}^{|\mathcal{A}_{\boldsymbol{\nu}}|}$. Na taj način kros-momenti reda manjeg ili jednakog $\boldsymbol{\nu}$ mogu se izračunati pomoću

$$\mathcal{B}^{(\boldsymbol{\nu})}\big\{\sigma_i(\boldsymbol{u})\big\} = \bigoplus_{j=1}^{|\mathcal{R}_i|} \bigoplus_{\substack{\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_k\in\Sigma^* \\ \boldsymbol{u}=\boldsymbol{v}_1\boldsymbol{u}_1\boldsymbol{v}_2\cdots\boldsymbol{v}_k\boldsymbol{u}_k\boldsymbol{v}_{k+1}}} w(A_i\to B_{i,j}) \otimes \bigotimes_{j=1}^{k} \mathcal{B}^{(\boldsymbol{\nu})}\big\{\sigma_{i_j}(\boldsymbol{u}_j)\big\} \tag{6.70}$$

sa baznim slučajem

$$\sigma_i(\boldsymbol{u}) = w\big(A_i\to\boldsymbol{u}\big). \tag{6.71}$$

Lema 5.3.1 daje formule za izračunavanje zbira i proizvoda proizvoljnog broja elemenata poluprstena, $w_n \in \mathbb{R}^{|\mathcal{A}_{\boldsymbol{\nu}}|};\ n = 1,\ldots,N$.

$$\Big(\bigoplus_{n=1}^{N} w_n\Big)^{(\boldsymbol{\alpha})} = \sum_{n=1}^{N} w_n^{(\boldsymbol{\alpha})}, \tag{6.72}$$

$$\Big(\bigotimes_{n=1}^{N} w_n\Big)^{(\boldsymbol{\alpha})} = \sum_{\boldsymbol{\beta}_1+\cdots+\boldsymbol{\beta}_N=\boldsymbol{\alpha}} \binom{\boldsymbol{\alpha}}{\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_N} \prod_{n=1}^{N} w_n^{(\boldsymbol{\beta}_n)}, \tag{6.73}$$

pa se dobija

$$\mu^{(\boldsymbol{\alpha})}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = \sum_{j=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_k\in\Sigma \\ \boldsymbol{u}=\boldsymbol{v}_1\boldsymbol{u}_1\boldsymbol{v}_2\cdots\boldsymbol{u}_k\boldsymbol{v}_k\boldsymbol{u}_{k+1}}} \sum_{\beta\le\alpha} \binom{\alpha}{\beta} p\big(A_i\to B_{i,j}\big)\cdot \boldsymbol{\Upsilon}\big(A_i\to B_{i,j}\big)^{\boldsymbol{\alpha}-\boldsymbol{\beta}}\cdot$$

$$\sum_{\boldsymbol{\gamma}_1+\cdots\boldsymbol{\gamma}_k=\boldsymbol{\beta}} \prod_{j=1}^{k} \binom{\boldsymbol{\beta}}{\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_k} \mu^{(\boldsymbol{\gamma}_j)}_{p,\boldsymbol{X}_{i_j}|\boldsymbol{u}_j}, \tag{6.74}$$

sa bazom rekurzije

$$\mu^{(\boldsymbol{\gamma})}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = p\big(A_i\to\boldsymbol{u}\big)\cdot\boldsymbol{\Upsilon}\big(A_i\to\boldsymbol{u}\big)^{\boldsymbol{\gamma}}. \tag{6.75}$$

Prethodno opisani algoritam za izračunavanje kros-momenata predstavlja generalizaciju algoritma koji su razvili *Li* i *Eisner* [56], za kros-momente reda $\boldsymbol{\alpha} = (1,1)$. Li i Eisner su uveli entropijski poluprsten drugog reda, i iskoristili ga u kombinaciji sa *inside* algoritmom. Entropijski poluprsten drugog reda predstavlja instancu binomnog poluprstena reda $(1,1)$, a pomenuti algoritam se može dobiti primenom preslikavanja $\mathcal{B}^{(\nu)}$ na rekurzivnu jednačinu (6.70)-(6.71).

6.3.3 Uslovni momenti prvog reda

Algoritam za izračunavanje momenata prvog reda dat je u [34], gde je razmatrana uslovna entropija za *PCFG*. U slučaju $\boldsymbol{\alpha} = 1$ uslovni-kros momenti (6.54) se svode na očekivanje od $\boldsymbol{X}_i$

$$\mu^{(1)}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = \sum_{\boldsymbol{\pi}\in\Omega_i(\boldsymbol{u})} p(\boldsymbol{\pi})X(\boldsymbol{\pi}). \tag{6.76}$$

U ovom slučaju rekurzivna jednačina (6.74)-(6.75) svodi se na

$$\mu^{(0)}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = \sum_{j=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_k\in\Sigma \\ \boldsymbol{u}=\boldsymbol{v}_1\boldsymbol{u}_1\boldsymbol{v}_2\cdots\boldsymbol{u}_k\boldsymbol{v}_k\boldsymbol{u}_{k+1}}} p\big(A_i \to B_{i,j}\big)\cdot\prod_{j=1}^{k}\mu^{(0)}_{p,\boldsymbol{X}_{i_j}|\boldsymbol{u}_j}, \tag{6.77}$$

$$\mu^{(1)}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = \sum_{j=1}^{|\mathcal{N}|} \sum_{\substack{\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_k\in\Sigma \\ \boldsymbol{u}=\boldsymbol{v}_1\boldsymbol{u}_1\boldsymbol{v}_2\cdots\boldsymbol{u}_k\boldsymbol{v}_k\boldsymbol{u}_{k+1}}} p\big(A_i \to B_{i,j}\big)\cdot\boldsymbol{Y}\big(A_i \to B_{i,j}\big)\cdot\prod_{j=1}^{k}\mu^{(0)}_{p,\boldsymbol{X}_{i_j}|\boldsymbol{u}_j} + p\big(A_i \to B_{i,j}\big)\cdot\sum_{n=1}^{k}\mu^{(1)}_{p,\boldsymbol{X}_{i_n}|\boldsymbol{u}_n}\prod_{\substack{j=1\\ j\neq n}}^{k}\mu^{(0)}_{p,\boldsymbol{X}_{i_j}|\boldsymbol{u}_j}, \tag{6.78}$$

sa baznim slučajem

$$\mu^{(0)}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = p\big(A_i \to \boldsymbol{u}\big), \tag{6.79}$$

$$\mu^{(1)}_{p,\boldsymbol{X}_i|\boldsymbol{u}} = p\big(A_i \to \boldsymbol{u}\big)\cdot Y\big(A_i \to \boldsymbol{u}\big). \tag{6.80}$$

Hwa [34] razmatra uslovnu entropiju za *PCFG*, koja je data u normalnoj formi Čomskog, za koju je $B_{i,j} = v_1 A_{i_1} v_2 A_{i_2} v_3$ i v_1, v_2, v_3 su prazni stringovi. Uslovna entropija se dobija kao moment $\mu^{(1)}_{p,\boldsymbol{X}_1|\boldsymbol{u}}$, pri čemu je $\boldsymbol{X}(\boldsymbol{\pi}) = -\ln p(\boldsymbol{\pi})$, za svako $\boldsymbol{\pi} \in \Omega_1$. Na taj način algoritam izložen u [34], može se izvesti zamenom uslova za Čomski normalnu formu u (6.74)-(6.75), sa $\boldsymbol{Y}(\pi_i) = -\ln p(\pi_i)$.

Literatura

[1] S. Aji and R. McEliece, *The generalized distributive law*, Information Theory, IEEE Transactions on, 46 (2000), pp. 325 –343.

[2] A. Azuma and Y. Matsumoto, *A generalization of forward-backward algorithm*, in Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part I, ECML PKDD '09, Berlin, Heidelberg, 2009, Springer-Verlag, pp. 99–114.

[3] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, *Optimal decoding of linear codes for minimizing symbol error rate (corresp.)*, Information Theory, IEEE Transactions on, 20 (1974), pp. 284 – 287.

[4] L. Baum, *An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes*, Inequalities, 3 (1972), pp. 1–8.

[5] L. E. Baum and T. Petrie, *Statistical inference for probabilistic functions of finite state Markov chains*, The Annals of Mathematical Statistics, 37 (1966), pp. 1554–1563.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[7] T. L. Booth and R. A. Thompson, *Applying probability measures to abstract languages*, IEEE Trans. Comput., 22 (1973), pp. 442–450.

[8] O. Cappé, E. Moulines, and T. Rydén, *Inference in hidden Markov models*, Springer series in statistics, Springer, Aug. 2005.

[9] R. Chang and J. Hancock, *On receiver structures for channels having memory*, Information Theory, IEEE Transactions on, 12 (1966), pp. 463 – 468.

[10] Z. Chi, *Statistical properties of probabilistic context-free grammars*, Comput. Linguist., 25 (1999), pp. 131–160.

[11] A. Churbanov and S. Winters-Hilt, *Implementing EM and Viterbi algorithms for hidden Markov model in linear memory*, BMC Bioinformatics, 9 (2008).

[12] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, McGraw-Hill Science / Engineering / Math, 2nd ed., December 2003.

[13] C. Cortes, M. Mohri, A. Rastogi, and M. Riley, *On the computation of the relative entropy of probabilistic automata*, Int. J. Found. Comput. Sci., 19 (2008), pp. 219–242.

[14] T. M. Cover and J. A. Thomas, *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*, Wiley-Interscience, 2006.

[15] R. G. Cowell, P. A. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter, *Probabilistic Networks and Expert Systems (Information Science and Statistics)*, Springer, New York, May 2003.

[16] J. Dauwels, A. Eckford, S. Korl, and H.-A. Loeliger, *Expectation maximization as message passing - part i: Principles and gaussian messages*, CoRR, abs/0910.2832 (2009).

[17] J. Dauwels, S. Korl, and H.-A. Loeliger, *Expectation maximization as message passing*, (2005).

[18] ——, *Steepest descent as message passing*, in Information Theory Workshop, 2005 IEEE, Sept. 2005, p. 5 pp.

[19] A. P. Dempster, N. M. Laird, and D. B. Rubin, *Maximum likelihood from incomplete data via the EM algorithm*, JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B, 39 (1977), pp. 1–38.

[20] F. Desbouvries, *Entropy Computation in Partially Observed Markov Chains*, in Bayesian Inference and Maximum Entropy Methods In Science and Engineering, A. Mohammad-Djafari, ed., vol. 872 of American Institute of Physics Conference Series, Nov. 2006, pp. 355–357.

[21] A. W. Eckford, *Channel estimation in block fading channels using the factor graph EM algorithm*, in In Proc. 22nd Biennial Symposium on Communications, 2004.

[22] J. Eisner, *Expectation semirings: Flexible EM for finite-state transducers*, in Proceedings of the ESSLLI Workshop on Finite-State Methods in Natural Language Processing (FSMNLP), G. van Noord, ed., 2001. Extended abstract (5 pages).

[23] J. Eisner, *Parameter estimation for probabilistic finite-state transducers*, in Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, Morristown, NJ, USA, 2002, Association for Computational Linguistics, pp. 1–8.

[24] Y. Ephraim and N. Merhav, *Hidden Markov processes*, Information Theory, IEEE Transactions on, 48 (2002), pp. 1518 –1569.

[25] S. Feng, R. Manmatha, and A. McCallum, *Exploring the use of conditional random field models and HMMs for historical handwritten document recognition*, in DIAL, 2006, pp. 30–37.

[26] J. Goodman, *Semiring parsing*, Comput. Linguist., 25 (1999), pp. 573–605.

[27] J. A. Grice, R. Hughey, and D. Speck, *Reduced space sequence alignment*, Computer Applications in the Biosciences, 13 (1997), pp. 45–53.

[28] R. Gupta, *Conditional random fields*, 2006. Technique Report, IIT Bombay.

[29] T. E. Harris, *The theory of branching processes*, (1963).

[30] A. Heim, V. Sidorenko, and U. Sorger, *Computation of distributions and their moments in the trellis*, Advances in Mathematics of Communications (AMC), 2 (2008), pp. 373–391.

[31] D. Hernando, V. Crespi, and G. Cybenko, *Efficient computation of the hidden Markov model entropy for a given observation sequence*, Information Theory, IEEE Transactions on, 51 (2005), pp. 2681 – 2685.

[32] R. A. Horn and C. R. Johnson, *Matrix analysis*, Cambridge University Press, New York, NY, USA, 1985.

[33] S. E. Hutchins, *Moments of string and derivation lengths of stochastic context-free grammars*, Information Sciences, 4 (1972), pp. 179 – 191.

[34] R. Hwa, *Sample selection for statistical grammar induction*, in Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13, Morristown, NJ, USA, 2000, Association for Computational Linguistics, pp. 45–52.

[35] V. M. Ilić, D. I. Mančev, B. T. Todorović, and M. S. Stanković, *Gradient computation in linear-chain conditional random fields using the entropy message passing algorithm*, Pattern Recognition Letters, 33 (2012), pp. 1776 – 1784.

[36] V. M. Ilić, M. D. Ćirić, and M. S. Stanković, *Cross-moments computation for stochastic context-free grammars*, CoRR, abs/1108.0353 (2011).

[37] V. M. Ilić, M. S. Stanković, and B. T. Todorović, *Computation of cross-moments using message passing over factor-graphs*, vol. 6, American Institute of MAthematical Sciences.

[38] ——, *Entropy message passing*, IEEE Transactions on Information Theory, 57 (2011), pp. 219–242.

[39] ——, *Entropy semiiring forward-backward algorithm for HMM entropy computation*, Transactions on Advanced Research, 8 (2012), pp. 8–15.

[40] V. Isham, *An introduction to spatial point processes and Markov random fields*, International Statistical Review / Revue Internationale de Statistique, 49 (1981), pp. 21–43.

[41] Z. Ivković, *Teorija verovatnoća sa matematičkom statistikom*, Prirodno-matematicki fakultet, Beograd, 1986.

[42] F. Jensen, *An Introduction to Bayesian Networks*, Springer Verlag, New York, 1996.

[43] F. Jensen, F. V. Jensen, and S. L. Dittmer, *From influence diagrams to junction trees*, in Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, L. R. de Mantaras and D. Poole, eds., San Francisco, CA, USA, 1994, Morgan Kaufmann Publishers Inc., pp. 367–373.

[44] F. V. Jensen and F. Jensen, *Optimal junction trees*, in UAI, R. L. de Mántaras and D. Poole, eds., Morgan Kaufmann, 1994, pp. 360–366.

[45] W. Khreich, E. Granger, A. Miri, and R. Sabourin, *On the memory complexity of the forward-backward algorithm*, Pattern Recognition Letters, 31 (2010), pp. 91–99.

[46] R. Kindermann, *Markov Random Fields and Their Applications (Contemporary Mathematics; Volume 1)*, Amer. Mathematical Society.

[47] S. Koenig and R. G. Simmons, *Unsupervised learning of probabilistic models for robot navigation*, in in Proceedings of the IEEE International Conference on Robotics and Automation, 1996, pp. 2301–2308.

[48] A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler, *Hidden Markov models in computational biology: applications to protein modeling*, Journal of Molecular Biology, 235 (1994), pp. 1501–1531.

[49] F. Kschischang, B. Frey, and H.-A. Loeliger, *Factor graphs and the sum-product algorithm*, Information Theory, IEEE Transactions on, 47 (2001), pp. 498 –519.

[50] A. Kulesza and B. Taskar, *Structured determinantal point processes*, in Advances in neural information processing systems 23, 2011.

[51] J. Lafferty, A. Mccallum, and F. Pereira, *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*, in Proc. 18th International Conf. on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2001, pp. 282–289.

[52] T. D. Lane, *Machine learning techniques for the computer security domain of anomaly detection*, PhD thesis, 2000. Major Professor-Brodley, Carla E.

[53] K. Lari and S. J. Young, *The estimation of stochastic context-free grammars using the inside-outside algorithm*, Computer Speech & Language, 4 (1990), pp. 35 – 56.

[54] S. Lauritzen and F. Jensen, *Local computation with valuations from a commutative semigroup*, Annals of Mathematics and Artificial Intelligence, 21 (1997), pp. 51–69. 10.1023/A:1018953016172.

[55] S. L. Lauritzen and D. J. Spiegelhalter, *Local computations with probabilities on graphical structures and their application to expert systems*, Journal of the Royal Statistical Society. Series B (Methodological), 50 (1988).

[56] Z. Li and J. Eisner, *First- and second-order expectation semirings with applications to minimum-risk training on translation forests*, in Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1, EMNLP '09, Morristown, NJ, USA, 2009, Association for Computational Linguistics, pp. 40–51.

[57] R. Lund, *Elementary probability theory with stochastic processes and an introduction to mathematical finance (4th ed.)*, The American Statistician, 58 (2004), pp. 173–174.

[58] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press, 2003.

[59] A. Madsen and F. Jensen, *Solving linear-quadratic conditional gaussian influence diagrams*, International Journal of Approximate Reasoning, 38 (2005), pp. 263–282.

[60] G. S. Mann and A. McCallum, *Efficient computation of entropy gradient for semi-supervised conditional random fields*, in Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers on XX, NAACL '07, Morristown, NJ, USA, 2007, Association for Computational Linguistics, pp. 109–112.

[61] R. J. McEliece, D. J. C. MacKay, and J.-F. Cheng, *Turbo decoding as an instance of Pearl's belief propagation algorithm*, Selected Areas in Communications, IEEE Journal on, 16 (1998), pp. 140–152.

[62] I. M. Meyer and R. Durbin, *Gene structure conservation aids similarity based gene prediction*, Nucleic acids research, 32 (2004), pp. 776–783.

[63] I. Miklós and I. M. Meyer, *A linear memory algorithm for Baum-Welch training*, BMC bioinformatics, 6 (2005).

[64] K. P. Murphy, Y. Weiss, and M. I. Jordan, *Loopy belief propagation for approximate inference: An empirical study*, in Proceedings of Uncertainty in AI, 1999, pp. 467–475.

[65] M.-J. Nederhof and G. Satta, *Kullback-leibler distance between probabilistic context-free grammars and probabilistic finite automata*, in Proceedings of the 20th international conference on Computational Linguistics, COLING '04, Stroudsburg, PA, USA, 2004, Association for Computational Linguistics.

[66] ——, *Computing partition functions of PCFGs*, Research on Language and Computation, 6 (2008), pp. 139–162. 10.1007/s11168-008-9052-8.

[67] D. Nilsson, *The computation of moments of decomposable functions in probabilistic expert systems*, in Proceedings of the 3rd International Symposium on Adaptive Systems, 2001, pp. 116–121.

[68] J. Pearl, *Reverend Bayes on inference engines: A distributed hierarchical approach*, in Proceedings of the American Association of Artificial Intelligence National Conference on AI, Pittsburgh, PA, 1982, pp. 133–136.

[69] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1 ed., Sept. 1988.

[70] W. Pieczynski, *Pairwise Markov chains*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, 25 (2003), pp. 634 – 639.

[71] W. Pieczynski and F. Desbouvries, *On Triplet Markov chains*, in Proceedings of the International Symposium on Applied Stochastic Models and Data Analysis (ASMDA 2005), May 2005, pp. 17–20.

[72] A. B. Poritz, *Hidden Markov models: a guided tour*, in Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on, 1988, pp. 7–13 vol.1.

[73] M. Protter, *Basic elements of real analysis*, Undergraduate texts in mathematics, Springer, New York, 1998.

[74] L. R. Rabiner, *A tutorial on hidden Markov models and selected applications in speech recognition*, in Proceedings of the IEEE, 1989, pp. 257–286.

[75] X. S. Raymond, *Elementary Introduction to the Theory of Pseudodifferential Operators (Studies in Advanced Mathematics)*, CRC Press, Boca Raton, 1991.

[76] F. Sha and F. Pereira, *Shallow parsing with conditional random fields*, 2003, pp. 213–220.

[77] C. E. Shannon, *A mathematical theory of communication*, Bell system technical journal, 27 (1948).

[78] S. Sivaprakasam and K. Sam Shanmugan, *A forward-only recursion based HMM for modeling burst errors in digital channels*, vol. 2, nov. 1995, pp. 1054 –1058 vol.2.

[79] C. A. Sutton, *Efficient training methods for conditional random fields*, PhD thesis, 2008. Adviser-Mccallum, Andrew K.

[80] C. Tarnas and R. Hughey, *Reduced space hidden Markov model training*, BIOINFORMATICS, 14 (1998), pp. 401–406.

[81] F. Tendeau, *Computing abstract decorations of parse forests using dynamic programming and algebraic power series*, Theoretical Computer Science, 199 (1998), pp. 145 – 166.

[82] A. B. Thaheema and A. Laradjia, *Classroom note: A generalization of leibniz rule for higher derivatives*, International Journal of Mathematical Education in Science and Technology, 34 (2001), pp. 905–907.

[83] S. Verdu and H. V. Poor, *Backward, forward and backward-forward dynamic programming models under commutativity conditions*, in Decision and Control, 1984. The 23rd IEEE Conference on, vol. 23, dec. 1984, pp. 1081 –1086.

[84] S. V. N. Vishwanathan, N. N. Schraudolph, M. W. Schmidt, and K. P. Murphy, *Accelerated training of conditional random fields with stochastic gradient methods*, in In ICML, 2006, pp. 969–976.

[85] C. Warrender, S. Forrest, and B. Pearlmutter, *Detecting intrusions using system calls: alternative data models*, 1999, pp. 133 –145.

[86] Y. Weiss, *Correctness of local probability propagation in graphical models with loops*, Neural Computation, 12 (2000), pp. 1–41.

[87] C. S. Wetherell, *Probabilistic languages: A review and some open questions*, ACM Comput. Surv., 12 (1980), pp. 361–379.

[88] N. Wiberg, *Codes and Decoding on General Graphs*, 1996.

[89] J. Yedidia, W. Freeman, and Y. Weiss, *Constructing free-energy approximations and generalized belief propagation algorithms*, Information Theory, IEEE Transactions on, 51 (2005), pp. 2282 – 2312.

Прилог 4/1

ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ
НИШ

КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, **РБР**:	
Идентификациони број, **ИБР**:	
Тип документације, **ТД**:	монографска
Тип записа, **ТЗ**:	текстуални / графички
Врста рада, **ВР**:	докторска дисертација
Аутор, **АУ**:	Велимир М. Илић
Ментор, **МН**:	Мирослав Д. Ћирић
Наслов рада, **НР**:	Израчунавање крос-момената над пробабилистичким контекстно-независним граматикама и пробабилистичким графичким моделима
Језик публикације, **ЈП**:	српски
Језик извода, **ЈИ**:	српски
Земља публиковања, **ЗП**:	Србија
Уже географско подручје, **УГП**:	Србија
Година, **ГО**:	2012.
Издавач, **ИЗ**:	ауторски репринт
Место и адреса, **МА**:	Ниш, Вишеградска 33.
Физички опис рада, **ФО**: (поглавља/страна/ цитата/табела/слика/графика/прилога)	6 глава, 92 стране, 89 цитата, 7 слика
Научна област, **НО**:	Рачунарске науке
Научна дисциплина, **НД**:	Алгоритми, пробабилистички модели
Предметна одредница/Кључне речи, **ПО**:	Алгоритми, крос-моменти, графички модели, контекстно-независне граматике
УДК	517.983.2 (043.3) 517.984/.986 (043.3)
Чува се, **ЧУ**:	Библиотека
Важна напомена, **ВН**:	Истраживања су финансирана од стране Министарства за просвету и науку republike Srbije, у оквиру развојног пројекта III044006.

Извод, **ИЗ**:	Крос-моменти векторске случајне променљиве представљају базичне статистичке величине које описују расподелу променљиве. Дефинишу се као очекивана вредност производа целобројних степена координата векторске случајне променљиве. Израчунавање крос-момената може постати захтевно уколико је број могућих реализација случајне променљиве велики, а практично неизводљив у случају када је бесконачан. Медјутим, за специфичну структуру расподеле случајне променљиве и структуру случајне променљиве овај проблем може бити решен на ефикасан начин, што представља тему ове дисертације. У дисертацији разматрамо три типа пробабилистичких модела: - Марковљеви ланци (скривени Марковљеви ланци и условна случајна поља), - Пробабилистички графички модели, - Пробабилистичке контекстно-независне граматике. Главни допринос ове дисертације налази се у новим алгоритмима за израчунавање крос-момената поменутих пробабилистичких модела који могу наћи примену у низу области као што су: вештачка интелигенција, статистика и обрада сигнала.
Датум прихватања теме, **ДП**:	13.11.2011.
Датум одбране, **ДО**:	
Чланови комисије, **КО**: Председник:	
Члан:	
Члан, ментор:	

ПРИРОДНО - МАТЕМАТИЧКИ ФАКУЛТЕТ
НИШ

KEY WORDS DOCUMENTATION

Accession number, **ANO**:	
Identification number, **INO**:	
Document type, **DT**:	monograph
Type of record, **TR**:	textual / graphic
Contents code, **CC**:	doctoral dissertation
Author, **AU**:	Velimir M. Ilić
Mentor, **MN**:	Miroslav D. Ćirić
Title, **TI**:	Cross-moments computation for probabilistic context-free grammars and probabilistic graphical models
Language of text, **LT**:	Serbian
Language of abstract, **LA**:	English
Country of publication, **CP**:	Serbia
Locality of publication, **LP**:	Serbia
Publication year, **PY**:	2012
Publisher, **PB**:	author's reprint
Publication place, **PP**:	Niš, Višegradska 33.
Physical description, **PD**: (chapters/pages/ref./tables/pictures/graphs/appendixes)	6 chapters, 92 pages, 89 citations, 7 figures
Scientific field, **SF**:	Algorithms, probabilistic models
Scientific discipline, **SD**:	Algorithms, cross-moments, graphical models, context-free grammars
Subject/Key words, **S/KW**:	Computer science
UC	517.983.2(043.3) 517.984/.986(043.3)
Holding data, **HD**:	Library
Note, **N**:	Research supported by Ministry of Education and Science of the Republic of Serbia, Grants No. III44006.

Abstract, **AB**:		The cross-moments of vector random variables are basic statistical quantities. They are defined as expected value of the product of integer powers of the entries of random variable. The computation of cross-moments may become demanding if random variable takes large number of values and practically impossible if the number is infinite. Nevertheless, if random variable and its distribution have specific structure, the problem can be efficiently solved, which is the topic of this thesis. Three types of probabilistic models are considered: - Markov chains (hidden Markov models and conditional random fields), - Probabilistic graphical models, - Probabilistic context-free grammars. In this thesis, we develop new algorithms for cross-moments computation for mentioned probabilistic models, which can be usefull in artificial intelligence, statistics and signal processing.
Accepted by the Scientific Board on, **ASB**:		13.11.2011.
Defended on, **DE**:		
Defended Board, **DB**:	President:	
	Member:	
	Member, Mentor:	